Enhancing controller's tuning reliability with multi-objective optimisation: from *Model in the loop* to *Hardware in the loop*

Jesús Velasco Carrau [a,*], Gilberto Reynoso-Meza [b,*], Sergio García-Nieto [a], Xavier Blasco [a]

[a] *Instituto Universitario de Automática e Informática Industrial, Universitat Politècnica de València, Camino de Vera s/n , Valencia 46022, Spain*

[b] *Industrial and Systems Engineering Graduate Program (PPGEPS), Pontificia Universidade Católica do Paraná (PUCPR), Rua Imaculada Conceição, 1155, Zip code 80215-901, Curitiba, PR, Brazil*

Abstract

In general, the starting point for the complex task of designing a robust and efficient control system is the use of nominal models that allow to establish a first set of parameters for the selected control scheme. Once the initial stage of design is achieved, control engineers face the difficult task of Fine-Tuning for a more realistic environment, where the environment conditions are as similar as possible to the real system. For this reason, in the last decades the use of Hardware-in-The-Loop (HiL) systems has been introduced. This simulation technique guarantees realistic simulation environments to test the designs but without danger of damaging the equipment. Also, in this iterative process of Fine-Tuning, it is usual to use different (generally conflicting/opposed) criteria that take into account the sensitivities that al- ways appear in every project, such as economic, security, robustness, performance, for example. In this framework, the use of multi-objective techniques are especially useful since they allow to study the different design alternatives based on the multiple existing criteria. Unfortunately, the combination of multi-objective techniques and verification

[*] Corresponding authors

Email addresses: jevecar@upvnet.upv.es (Jesús Velasco Carrau), g.reynosomeza@pucpr.br (Gilberto Reynoso-Meza), sergarro@isa.upv.es (Sergio García-Nieto), xblasco@isa.upv.es (Xavier Blasco)

schemes based on Hardware-In-The-Loop presents a high incompatibility. Since obtaining the optimal set of solutions requires a high computational cost that is greatly increased when using Hardware- In-the-Loop. For this reason, it is often necessary to use less realistic but more computationally efficient verification schemes such as Model in the Loop (MiL), Software in the Loop (SiL) and Processor in the Loop (PiL). In this paper, a combined methodology is presented, where multi-objective optimisation and multi-criteria decision making steps are sequentially performed to achieve a final control solution. The authors claim that while going towards the optimisation sequence over MiL → SiL → PiL → HiL platforms, the complexity of the problem is unveiled to the designer, allowing to state meaningful design objectives. In addition, safety in the step between simulation and reality is significantly increased.

Keywords: multi-objective optimisation, controller tuning, hardware in the loop, flight control system.
20XX MSC: 00-01, 99-00

1. Introduction

A controller tuning task typically starts with a certain nominal model of the process under consideration. With such a nominal model, and with a previously selected controller structure, the tuning process will seek a suitable controller, fulfilling several requirements and performance specifications (hereafter design objectives) imposed by the designer. Such design objectives range from time to frequency domain exigencies, requirements and/or constraints.

In spite of the usefulness of using a nominal model for controller tuning purposes, for some applications further performance evaluation is required. Therefore, with the aim of enhancing controller's performance evaluation, different platforms could be used; for example, using a *hardware in the loop* (HiL) platform has become an standard practice in order to evaluate embedded controllers, with the goal of getting a more reliable measure of their performance [Lu et al., 2007, White et al., 2011]. Such platforms are common in automotive [Choi and Lee, 2012] and aeronautic/aerospace sectors [Jeon and Jung, 2012], where it is required to enhance the quality, safety and verification testing of their subsystems [Samad and Stewart, 2013].

On the other hand, it is not unusual to state a controller tuning task as an optimisation problem. The designer's task is to define one or more performance objectives to fulfil; afterwards, adjusting the tunable controller's parameters using an optimisation algorithm in order to meet such design objectives. Nevertheless, designs found with a *pure-performance* optimisation approach are often prone to be highly sensitive to the parameters used in the nominal model [Panagopoulos and Åström, 2000, Åström and Hägglund, 2001, Garpinger et al., 2014]; therefore, they might be useless in a practical sense. According to this, assessing robustness and relia- bility constraints (or objectives) has become the standard in such optimisation instances. The former lead to robust design optimisation (RDO), where the aim is to optimise the performance of the controller in the nominal model and simultaneously minimize its sensitivity; the latter leads to reliability-based design optimisation (RBDO), commonly based on stochastic analysis and its aim is to provide a measure of risk of failure [Frangopol and Maute, 2003]. Different approaches for RBDO have been used, as montecarlo sampling, simulation techniques or first/second order reliability methods [Valdebenito and Schuëller, 2010]

Therefore, the designer is, in general, dealing with a multi-objective problem (MOP), where performance measures are in conflict with the reliability or robustness indexes. Multi-objective optimisation (MOO) has shown to be a valuable tool for controller tuning [Reynoso-Meza et al., 2014b] when multiple and conflictive design objectives appear. It handles the simultaneous optimisation of several conflicting objectives, in order to provide what is known as the Pareto set [Miettinen, 1998], where all solutions are Pareto optimal *i.e.* they have different trade-off between conflicting objectives.

The aim of this paper is to provide a systematic approach, using (successively) different platforms in order to evaluate the controller's performance with multi-objective optimisation techniques. Reliability methods have been merged before with multi-objective optimisation [Coelho, 2015] or HiL platforms within the MOO process [Stewart et al., 2004, Woźniak, 2011] or within the MCDM stage [Gladwin et al., 2010]; nevertheless new methodologies to integrate such approach when the computational burden in the HiL is considerable, might be useful for control engineers. This is because, although tuning controllers directly in a HiL set-up by means of MOO would be a perfect match, it is usually too time-demanding in practice. This time cost leads to other difficulties that make optimising *from scratch* in the HiL platform prohibitive.

Other less realistic (and less complex) platforms such as *Model in the Loop* (MiL), *Software in the Loop* (SiL) and *Processor in the Loop* (PiL) can be previously used in the multi-objective optimisation procedure. Thereby, in this paper a methodology is presented, where multi-objective optimisation and multi-criteria decision making steps are sequentially performed over those platforms, going from the least to the most complex, in order to achieve a final control solution. First, more meaningful objectives can be posed as the designer gets more knowledge about the interaction between the system and the control structure. Also preferences on the objectives are more ”maturely” included. Second, objectives and decision variables bounds can be better delimited.

The remainder of this paper is as follows: in Section 2 brief backgrounds on controller’s performance and MOO are given; in Section 3 the methodological proposal of this work is presented and it is evaluated in an aircraft platform in Section 4. The purpose will be to accomplish a certain flight mission via the supervision of several way-points autonomously, which is reported in Section 5. Finally, some concluding remarks and further directions of this work are commented.

2. Background

In this section a brief background on controller's performance evaluation in engineering design and MOO techniques will be given, in order to state a common framework for the methodological proposal in this work.

2.1. Controllers' evaluation in engineering design

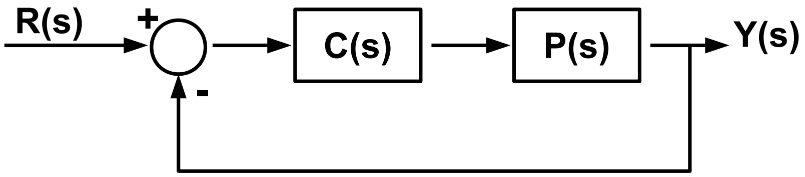


Figure 1: Basic control loop.

According to [Åström and Hägglund, 2001], any controller tuning procedure should consider design objectives related with:

- Load disturbance response
- Measurement noise response
- Setpoint response
- Robustness to model uncertainties

In agreement with the problem at hand, fulfilling one or some of them will be more (or less) preferable by the designer. According to the basic control loop of Fig. 1, some common choices in controller tuning [Reynoso-Meza et al., 2014b] for design objectives are:

- Maximum value of sensitivity function

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

- Integral of the absolute error value

$$J_{IAE}(\boldsymbol{x}) = \int_{t=t_0}^{T_f} |r(t) - y(t)|\, dt \tag{2}$$

- Total variation of control action

$$J_{TV}(\boldsymbol{x}) = \int_{t=t_0}^{T_f} \left| \frac{du}{dt} \right| \tag{3}$$

where $r(t)$, $y(t)$, $u(t)$ are the reference, measured variable and control action in time t. Equations (2) and (3) are commonly used for setpoint response and load disturbance, while for example Equation (1) has been used to guarantee a desired level of robustness. Time performance design objectives are usually preferred in industrial applications over frequency domain, as industrial requirements are usually expressed in such terms [Moberg et al., 2009].

Different platforms are available to evaluate the performance of a controller. Regarding proximity to the real set-up, the authors are using the following division:

- *Model in the loop* (MiL): a classical approach, where a nominal model is used to calculate and evaluate the performance of a controller.

- *Software in the loop* (SiL): the approach where the controller is evaluated *as it will be embedded*; that is, using the coding/script as it will be implemented in the embedded control device.

- *Processor in the loop* (PiL): the approach where the controller is executed in the processor/device where it will be embedded. Note that this is normally a real-time simulation.

- *Hardware in the loop* (HiL): the platform where the interactions (including physical communications) among processor, sensors and actuators are placed inside the real-time simulation loop.

The goal of using one platform over another, is on the one hand, getting a more meaningful and deeper understanding of the controller's performance to be implemented; on the other hand, getting a certain grade of reliability on its performance measure. Such measure can be expressed as risk of failure [Stengel and Marrison, 1992] or with probabilistic indices [Alfi et al., 2015] Hereafter, this set of platforms will be denoted as **X**iL platforms.

In any case, the conflict between robustness and performance arises [Garpinger et al., 2014], and therefore, MOO techniques might be an appealing tool to address the controller tuning problem.

2.2. Multi-objective optimisation design review

As referred in [Miettinen, 1998], a MOP [1], can be stated as follows:

$$\min_{\boldsymbol{x}} \boldsymbol{J(x)} \quad = \quad [J_1(\boldsymbol{x}), \ldots, J_m(\boldsymbol{x})] \tag{4}$$

subject to

$$\boldsymbol{K}(\boldsymbol{x}) \quad \leq \quad 0 \tag{5}$$

$$\boldsymbol{L}(\boldsymbol{x}) \quad = \quad 0 \tag{6}$$

$$\underline{x_i} \leq x_i \quad \leq \quad \overline{x_i}, \;\; \forall i = [1, \ldots, n] \tag{7}$$

where $\boldsymbol{x} = [x_1, x_2, \ldots, x_n]$ is defined as the decision vector; $\boldsymbol{J}(\boldsymbol{x})$ as the objective vector and $\boldsymbol{K}(\boldsymbol{x})$, $\boldsymbol{L}(\boldsymbol{x})$ as the inequality and equality constraint vectors respectively; $\underline{x_i}, \overline{x_i}$ are the lower and upper bounds in the decision (or search) space $\boldsymbol{X}$.

It has been pointed out that there is not a single solution in MOPs, because there is not (in general) a better solution in all the objectives. Therefore, a

[1] Any maximisation problem can be converted to a minimization one. For each of the objectives that have to be maximised, the transformation: $\max J_i(\boldsymbol{x}) = -min(-J_i(\boldsymbol{x}))$ could be applied.

set of solutions, the Pareto set $\mathbf{\Theta}_P$, is defined. Each solution in the Pareto set defines an objective vector in the Pareto front $\boldsymbol{J}_P$. All solutions in the Pareto front conform a set of Pareto optimal and non-dominated solutions (Fig. 2):

Definition (Pareto optimality [Miettinen, 1998]): An objective vector $\boldsymbol{J}(\boldsymbol{x}^1)$ is Pareto optimal if there is not another objective vector $\boldsymbol{J}(\boldsymbol{x}^2)$ such that $J_i(\boldsymbol{x}^2) \leq J_i(\boldsymbol{x}^1)$ for all $i \in [1, 2, \ldots, m]$ and $J_j(\boldsymbol{x}^2) < J_j(\boldsymbol{x}^1)$ for at least one j, $j \in [1, 2, \ldots, m]$.

Definition (Dominance [Coello and Lamont, 2004]): An objective vector $\boldsymbol{J}(\boldsymbol{x}^1)$ is dominated by another objective vector $\boldsymbol{J}(\boldsymbol{x}^2)$ iff $J_i(\boldsymbol{x}^2) \leq J_i(\boldsymbol{x}^1)$ for all $i \in [1, 2, \ldots, m]$ and $J_j(\boldsymbol{x}^2) < J_j(\boldsymbol{x}^1)$ for at least one j, $j \in [1, 2, \ldots, m]$. This is denoted as $\boldsymbol{J(x_2)} \preceq \boldsymbol{J(x_1)}$.

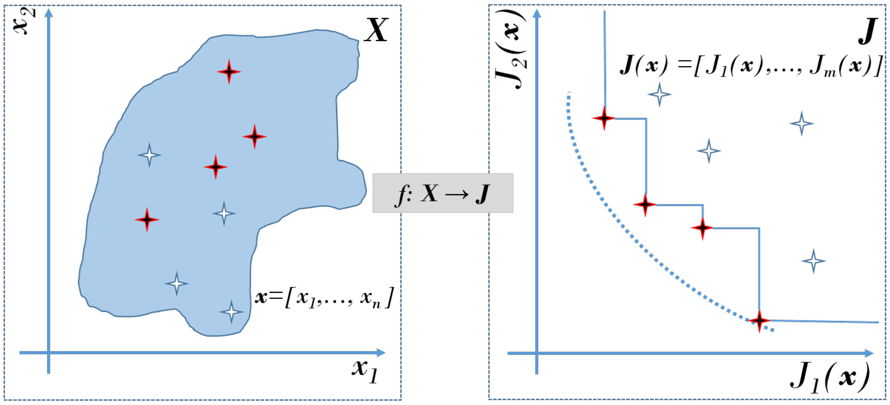


Figure 2: Pareto optimality and dominance concepts. A Pareto front (dotted line in objective space $\boldsymbol{J}$) is approximated with a set solutions (depicted with stars) selected from the fesible decision space $\boldsymbol{X}$. Dark solutions are non-dominated solutions in the set and therefore, they are used to build a Pareto front approximation (solid line). Remainder solutions are dominated solutions.

It is important to notice that the Pareto front is usually unknown, and the designer can only rely on a Pareto front approximation $\boldsymbol{J}_P^*$ and Pareto set approximation $\mathbf{\Theta}_P^*$. In order to successfully embed the multi-objective optimisation concept into a design process, three fundamental steps are required: the MOP statement (measure); the MOO process (search); and the multi-criteria decision making (MCDM) step (multicriteria analysis). This procedure will be named Multi-objective Optimisation Design (MOOD) procedure (Fig. 3).

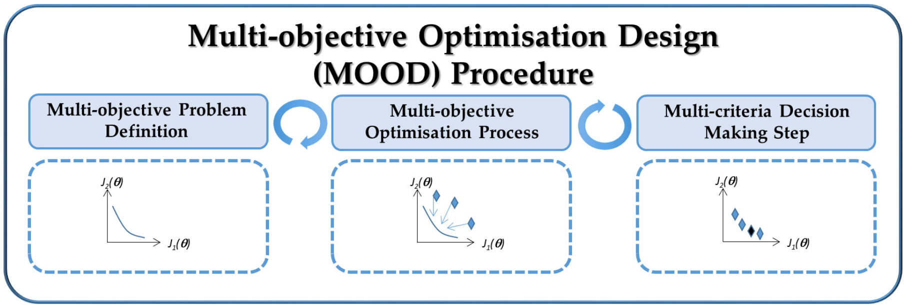

Multi-objective Optimisation Design (MOOD) Procedure
Multi-objective Problem Definition
Multi-objective Optimisation Process
Multi-criteria Decision Making Step
$J_2(\theta)$
$J_1(\theta)$
$J_2(\theta)$
$J_1(\theta)$
$J_2(\theta)$
$J_1(\theta)$

Figure 3: Multi-objective Optimisation Design (MOOD) procedure.

MOOD procedures have shown to be a valuable tool for controller tuning applications (see [Reynoso-Meza et al., 2014a, Meza et al., 2017], and references therein). Such techniques have been used in different controller structures [Meza et al., 2016]; for example PID controllers [Wang et al., 2016], fractional order controllers [Sánchez et al., 2017, Zamani et al., 2017] and state space controllers [Hassani and Lee, 2016]. They enable the designer or decision maker (DM) to have a close embedment into the design process; since it is possible to take into account each design objective individually, they also enable comparing design alternatives, in order to select a controller fulfilling the expected trade-off among conflicting objectives. Such procedures have been used with success when (1) it is difficult to find a reasonable trade-off for a controller tuning fulfilling several requirements; and (2) it is worthwhile analysing design objectives exchange among design alternatives.

In spite of its success in controller tuning applications, few works focus on incorporating the **X**iL platforms (besides the HiL) in their procedures ([Stewart et al., 2004, Woźniak, 2011] in the MOO process or [Gladwin et al., 2010] in the MCDM stage). Next, an integrative framework between **X**iL platforms and the MOOD procedure for controller tuning will be presented. Here it is assumed that:

- Using directly the HiL is not always possible in the MOO process, due to computational burden.
- While going towards the sequence MiL → SiL → PiL → HiL the efforts required to evaluate a controller are gradually increased. This is because (1) the more meaningful the objectives are the more complex they become (2) setup and analysis also increase in complexity and time.
- Not only robustness but also reliability of the controller's performance will be sought.

3. XiL platforms within a MOOD framework for controller tuning applications

In this section, an integrative framework to enhance controller performance evaluation in MOOD procedures is given. Such framework considers using different **X**iL platforms throughout the optimisation and decision making process. Firstly, the MOP statement, the MOO process and the MCDM stage will be stated for controller tuning; afterwards, a full integration with the **X**iL platforms will be proposed.

3.1. The MOOD process

3.1.1. MOP statement

According to Section 2, the general MOP that must be considered in controller tuning applications is:

$$\begin{aligned} \min_{\boldsymbol{x}} \boldsymbol{J(x)} &= [\boldsymbol{J}_{SR}(\boldsymbol{x}), \boldsymbol{J}_{LR}(\boldsymbol{x}), \boldsymbol{J}_{NR}(\boldsymbol{x}), \\ &\quad \boldsymbol{J}_{RDO}(\boldsymbol{x}), \boldsymbol{J}_{RBDO}(\boldsymbol{x}, \phi)] \end{aligned} \tag{8}$$

Where $\boldsymbol{x}$ is now the vector of tunable parameters of the controller structure selected; $\boldsymbol{J}_{SR}(\boldsymbol{x})$, $\boldsymbol{J}_{LR}(\boldsymbol{x})$, $\boldsymbol{J}_{NR}(\boldsymbol{x})$ are the set of design objectives related with setpoint response, load disturbance response and measurement noise response respectively (performance objectives); $\boldsymbol{J}_{RBO}(\boldsymbol{x})$, $\boldsymbol{J}_{RBDO}(\boldsymbol{x})$ are the set of design objectives related with robust design optimisation and reliability-based design optimisation respectively (robustness/reliability objectives). RBDO design objectives will be related with the stochastic evaluation of the performance objectives for different scenarios. That is, for example, given a set of scenarios Φ and a performance objective for a given scenario $J(\boldsymbol{x}, \cdot)$, plausible design objectives are (for example) the worst case performance (Equation (9)) or the variance on the performance (Equation (10))

$$\begin{aligned} J_{worst} &= \max\left(\boldsymbol{\varsigma}\right) \\ &\varsigma = J(\boldsymbol{x}, \phi), \forall \phi \in \Phi \end{aligned} \tag{9}$$

$$\begin{aligned} J_{var} &= \sigma\left(\boldsymbol{\varsigma}\right) \\ &\varsigma = J(\boldsymbol{x}, \phi), \forall \phi \in \Phi \end{aligned} \tag{10}$$

While it is common to state the design objectives or constraints in a RDO sense, stating them to get reliability (RBDO) and actively using them in the optimisation process is less common. However, they are quite useful, since they provide a deeper (more reliable) insight on controller performance, its risk of failures and its expected behaviour [Alfi et al., 2015, Yurchenko and Alevras, 2014, Moberg et al., 2009, Stengel and Marrison, 1992].

As a system might comprise several sub-processes and their interactions (that is, a multi-variable process), several design objectives for each of the sub-processes may appear. The designer might need to measure and optimise them all. In [Reynoso-Meza et al., 2013b] it was intended to provide a general framework for controller tuning, dealing with such many-objective optimisation statement (usually $m > 3$). Nevertheless, in spite of its usefulness, the approximated Pareto front could contain a considerable amount of solutions, which could in turn overwhelm the DM in the MCDM stage, even for a 2x2 multi-variable process. Therefore a different approach is needed.

A feasible approach in controller tuning can be found when incorporating preferences in the optimisation process[2]. This is possible since (1), the designer has an idea of the objectives which he/she needs to meet and/or (2), a reference

[2] Correlation analysis might be useful in order to reduce the dimensionality of the problem, nevertheless they don't embed information about the preferences that might be useful for the designer.

controller exists (manually tuned or using tuning rules)[3]. Lets assume that the designer can incorporate such preferences via the following function $\mathfrak{f}$:

$$\mathfrak{f} : \boldsymbol{x} \subseteq X \rightarrow (\mathfrak{P} \vee \mathfrak{P}^C) \tag{11}$$

That is, $\mathfrak{f}(x)$ is a function which determines if a given solution is aligned with the designer's preferences ($\mathfrak{P}$) or not ($\mathfrak{P}^C$), regarding the desired performance of the control loop (time and/or frequency domain). Such function and preferences [4] can be incorporated numerically into the set of constraints previously stated in eq. (4). Hence, the MOP statement of eq. (8) will be subject to:

$$\begin{aligned} \boldsymbol{K}(\boldsymbol{x}) &\leq 0 \qquad (12)\\ \boldsymbol{L}(\boldsymbol{x}) &= 0 \qquad (13)\\ \mathfrak{f}(\boldsymbol{x}) &\in \mathfrak{P} \qquad (14)\\ \underline{x_i} \leq x_i &\leq \overline{x_i}, \ \ \forall i = [1, \ldots, n] \qquad (15) \end{aligned}$$

Given the above, an algorithm to deal with this many-objectives optimisation instance, as well as with preferences and constraints, is required.

3.1.2. MOO process

Several algorithms exist, and the selection of one over another should be pondered by the characteristics of the problem (for instance multimodality, many-objectives, expensive optimisation) and the expected outcome (convergence, diversity and pertinency).

The approach presented by [Reynoso-Meza et al., 2016b] using the sp-MODEII[5] algorithm [Reynoso-Meza et al., 2014a] will be followed in this paper. Its main

[3] If no reference controller is available, a preliminary analysis on a Pareto front approximation (calculated without preferences) might provide the required information in order to state the preference matrix.

[4] Hereafter, *fraktur* style will be used to denote such input from the designer.

[5] Scripts and tutorials available at `www.mathworks.com/matlabcentral/fileexchange/authors/289050`.

characteristics are useful in order to deal with many-objective optimisation statements, covering the basic properties of convergence, diversity and pertinency of the Pareto front. Such characteristics are:

- It uses Differential Evolution [Storn and Price, 1997] algorithm as optimisation engine, which has shown a good trade off between global search and convergence to the Pareto front [Das and Suganthan, 2010].
- The objective space is partitioned in spherical sectors [Reynoso-Meza et al., 2010], in order to improve the spreading along the Pareto front.
- Preferences $\mathfrak{P}$ are coded *a priori* in the form of a preference matrix $\mathfrak{m}$ by the designer (Table 1). For each design objective $J_i(\boldsymbol{x})$, $i \in [1, \cdots, m]$ six values $(J_i^0, \cdots, J_i^5)$ are stated (in the original units for each design objective) in order to define 5 preference ranges: highly desirable (HD), desirable (D), tolerable (T) undesirable (U) and highly undesirable (HU).
- Preferability function $\mathfrak{f}(\cdot)$ is computed using the above commented matrix $\mathfrak{m}$. With such preference matrix, the algorithm computes a global physical programming index (GPP) [Blasco et al., 2012] to evaluate the preferability of one solution over another solution, which is a modified form of the physical programming methods [Messac, 1996]. Such index is used to prune the approximated Pareto front, in order to get a compact and pertinent approximation, with the number of desired solutions imposed by the designer.
- This approach enables to state a difference between design objectives for decision making and for optimisation. In the former case, they represent the design objectives where de DM is willing to perform a decision making and where the objective space is partitioned; the latter, are design objectives that are used in the optimisation stage, used to calculate the GPP index, but are not used *a priori* in the design objective partitioning. For example, in Fig. 4 such difference is depicted for a 3-objective problem where two Pareto-optimal solutions lie. Assume that square solution has

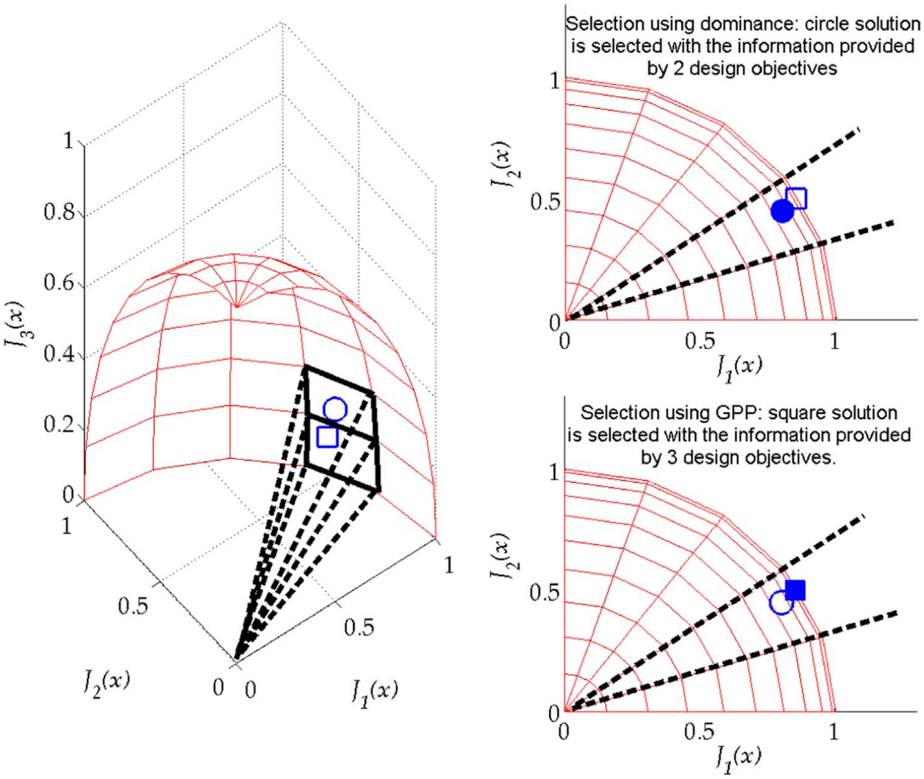
Table 1: Typical preference matrix $\mathfrak{m}$ for GPP index. Five preference ranges have been defined: highly desirable (HD), desirable (D), tolerable (T) undesirable (U) and highly undesirable (HU).

	Preference matrix $\mathfrak{m}$										
	←	HD	→ ←	D	→ ←	T	→ ←	U	→ ←	HU	→
Objective	J_i^0		J_i^1		J_i^2		J_i^3		J_i^4		J_i^5
$J_1(\boldsymbol{x})$ [-]	J_1^0		J_1^1		J_1^2		J_1^3		J_1^4		J_1^5
$\vdots$	$\vdots$		$\vdots$		$\vdots$		$\vdots$		$\vdots$		$\vdots$
$J_m(\boldsymbol{x})$ [-]	J_m^0		J_m^1		J_m^2		J_m^3		J_m^4		J_m^5

a better GPP index than the circle solution. In the 2D projection (design objective for decision making), seems that the circle solution should be selected, since it dominates the square solution and both belong to the same spherical sector; nevertheless when considering the overall objective space in the 3D representation the selection of one over the another will rely on the $\mathfrak{f}(\boldsymbol{x})$ preferability function and thus, the square solution is selected. Such technique is helpful for many-objectives optimisation instances in controller tuning purposes [Reynoso-Meza et al., 2016b]. Basically, we are spread- ing solutions in a 2D space, where the designer might feel more comfortable to analyse the Pareto front, but taking into account the m-dimensional space with the GPP index, and therefore, all design objectives are considered.

3.1.3. MCDM step

The main goal in the MCDM step, is to (1) select one preferable solution $\mathfrak{x} \in \boldsymbol{\Theta}_P^*$ or (2) select a subset $\boldsymbol{\mathfrak{X}} \subseteq \boldsymbol{\Theta}_P^*$ of preferable and feasible solutions for further evaluation. In any case, the DM needs to perform an accurate analysis of the $\boldsymbol{\Theta}_P^*$ approximation in a multidimensional and multicriteria environment. It is widely accepted that visualization tools are valuable and provide to the DM



Selection using dominance: circle solution is selected with the information provided by 2 design objectives

Selection using GPP: square solution is selected with the information provided by 3 design objectives.

Figure 4: Difference between design objectives for optimisation (3-objective) and for decision making (2-objective) using dominance and preferences. Square-solution has a better GPP index for 3-objective than circle-solution.

meaningful methods to analyse the Pareto front and take decisions [Bonissone et al., 2009, Tušar and Filipič, 2014].

For two-dimensional problems (and sometimes for three-dimensional problems) it is usually straightforward to make an accurate graphical analysis of the Pareto front, but the difficulty increases with the dimension of the problem. Common alternatives to tackle an analysis in higher dimensions are Scatter Diagrams, Parallel Coordinates [Inselberg, 1985, 2009] and Level Diagrams (LD) [Blasco et al., 2008, Reynoso-Meza et al., 2013a]. Recently, hybrid tools merging Parallel Coordinates, Dendograms, and Cluster Maps have been proposed [Cela and Bollaín, 2012].

Given that design objectives for decision making and for optimisation will be stated, a reduced space for MCDM will be sought (up to 3 objectives); nevertheless, as auxiliary visualization, a LD tool will be used [6], due to its capabilities to propagate interpretability from $\boldsymbol{J}_P^*$ to $\boldsymbol{\Theta}_P^*$ and due to its robustness, scalability and simplicity properties [Tušar and Filipič, 2014]. They have been used before for design applications [Perera et al., 2013, Pourzeynali et al., 2013] and controller tuning [Hajiloo et al., 2012, Koetje et al., 2013]. Additionally, in order to evaluate 2-5 design alternatives, radial plots[7] are used due to their simplicity to depict few design options with several design objectives.

3.2. Integration to enhance controller's performance evaluation

We will state a minimal example of an integrated MOOD using the MiL and SiL platforms. Again, it is important to remember that, enhancing controller's performance evaluation is always possible with a HiL platform. What is not always possible, however, is to use such platform actively in the MOO process (globally or locally) at least *from scratch.*

The general purpose, before getting into the SiL platform, is to get a suitable set of solutions, as well as to improve the DM's knowledge on the problem's

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

[7] Also known as star, rose, spider diagrams.

trade-off by using the information extracted from the MiL platform. That is, the designer will define a MOP at the MiL platform:

$$
\begin{array}{rcll}
\multicolumn{4}{c}{\min_{\boldsymbol{x}} \boldsymbol{J(x)}\,|_{MiL}} \\
s.t. & & & \\
\boldsymbol{K}(\boldsymbol{x})\,|_{MiL} & \leq & 0 & \\
\boldsymbol{L}(\boldsymbol{x})\,|_{MiL} & = & 0 & \\
\mathfrak{f}_{MiL}(\boldsymbol{x}) & \in & \mathfrak{P}\,|_{MiL} & \\
\underline{x_i} \leq x_i & \leq & \overline{x_i}, & \forall i = [1, \ldots, n]
\end{array}
\tag{16}
$$

And starting from such MOP statement, a MOO process will output the Pareto Set and Front approximations $\boldsymbol{\Theta}_P^*\,|_{MiL}$, $\boldsymbol{J}_P^*\,|_{MiL}$ respectively, with a given $\mathfrak{m}_{MiL}$.

A MCDM analysis on $\boldsymbol{\Theta}_P^*\,|_{MiL}$, $\boldsymbol{J}_P^*\,|_{MiL}$ might include the SiL platform. Design objectives could be evaluated in the SiL environment, or new indexes might be calculated (different form the $\boldsymbol{J(x)}\,|_{MiL}$ statement, but important to consider). After such procedure at the MCDM stage, the designer will select a set $\mathfrak{X} \subseteq \boldsymbol{\Theta}_P^*\,|_{MiL}$ of preferable solutions; lets denote this subset as $\mathfrak{X}_{MiL}$.

Nevertheless, if such additional indexes are important, the designer could also consider an active seeking in the SiL platform. Lets suppose this is the case, then a new MOP can be stated:

$$
\begin{array}{rcll}
\multicolumn{4}{c}{\min_{\boldsymbol{x}} \boldsymbol{J(x)}\,|_{SiL}} \\
s.t. & & & \\
\boldsymbol{K}(\boldsymbol{x})\,|_{SiL} & \leq & 0 & \\
\boldsymbol{L}(\boldsymbol{x})\,|_{SiL} & = & 0 & \\
\mathfrak{f}_{SiL}(\boldsymbol{x}, \mathfrak{X}_{MiL}) & \in & \mathfrak{P}\,|_{SiL} & \\
\underline{x_i} \leq x_i & \leq & \overline{x_i}, & \forall i = [1, \ldots, n]
\end{array}
\tag{17}
$$

Again, after a MOO process based on the MOP from eq. (17), the designer will obtain the Pareto Set and Front approximations $\boldsymbol{\Theta}_P^* |_{SiL}$, $\boldsymbol{J}_P^* |_{SiL}$ respectively, with a given $\mathfrak{m}_{SiL}$. It is important to notice that the $\mathfrak{f}_{SiL}(\boldsymbol{x}, \mathfrak{X}_{MiL})$ function includes, besides preferences of the DM, the information gained in the previous MCDM step (here represented by the subset $\mathfrak{X}_{MiL}$). This is aligned with the philosophy of *innovization* [Deb et al., 2014], where information from the MOO stage is retrieved in order to gain a deeper knowledge on trade-off of the current MOP and its design objectives (that is, innovation trough optimisation).

The advantage of following this process relies on two facts. First, from the computational sense, evaluating performance of a controller in SiL platforms could be more expensive than doing it in MiL platforms; this is due to the fact that SiL platforms would include (for example) sampling rate effects (hence slowing down the simulations). Therefore, performing the MOO process *from scratch* in a SiL platform is sometimes impractical. More practical could be however, using previous information from the $\boldsymbol{\Theta}_P^* |_{MiL}$, $\boldsymbol{J}_P^* |_{MiL}$ in order to refine the preference matrix $\mathfrak{m}_{MiL}$ and the objective space bounds $\overline{x_i}$ and $\underline{x_i}$ $\forall i = [1, \ldots, n]$; furthermore, solutions $\mathfrak{X}_{MiL}$ might be used as initial population (initial candidate solution in the optimisation algorithm) to accelerate the convergence in the SiL platform. Second, from the problem knowledge sense, a progressive approximation to reality may help to gradually reduce the engineer's uncertainty about the problem. Note that an approach to reality usually means an increase in complexity. In this way, the more information the designer gets from previous iterations, the more accurate is the request on the controller performance.

Following this idea, the MOPs of Equations (18) and (19) could be defined. Such elements are summarised in Fig. 5.

$$\min_{\boldsymbol{x}} \boldsymbol{J(x)}\,|_{PiL} \tag{18}$$

$s.t.$

$$\begin{array}{rcl}
\boldsymbol{K}(\boldsymbol{x})\,|_{PiL} & \leq & 0 \\
\boldsymbol{L}(\boldsymbol{x})\,|_{PiL} & = & 0 \\
\mathfrak{f}_{PiL}(\boldsymbol{x}, \mathfrak{X}_{MiL}, \mathfrak{X}_{SiL}) & \in & \mathfrak{P}\,|_{PiL} \\
\underline{x_i} \;\leq\; x_i & \leq & \overline{x_i}, \qquad \forall i = [1, \ldots, n]
\end{array}$$

$$\min_{\boldsymbol{x}} \boldsymbol{J(x)}\,|_{HiL} \tag{19}$$

$s.t.$

$$\begin{array}{rcl}
\boldsymbol{K}(\boldsymbol{x})\,|_{HiL} & \leq & 0 \\
\boldsymbol{L}(\boldsymbol{x})\,|_{HiL} & = & 0 \\
\mathfrak{f}_{HiL}\,(\boldsymbol{x}, \mathfrak{X}_{MiL}, \mathfrak{X}_{SiL}, \mathfrak{X}_{PiL}) & \in & \mathfrak{P}\,|_{HiL} \\
\underline{x_i} \;\leq\; x_i & \leq & \overline{x_i}, \qquad \forall i = [1, \ldots, n]
\end{array}$$

It is important to remark that sometimes using all the **X**iL platforms is not possible nor even practical. As an example, the designer might use only two of them: a SiL (comprising the MiL) and the HiL (comprising the PiL). This election will highly depend on the facilities and infrastructures available for such tests, as well as on the complexity of the problem. A case of study for which this methodology could be suitable will be presented next.

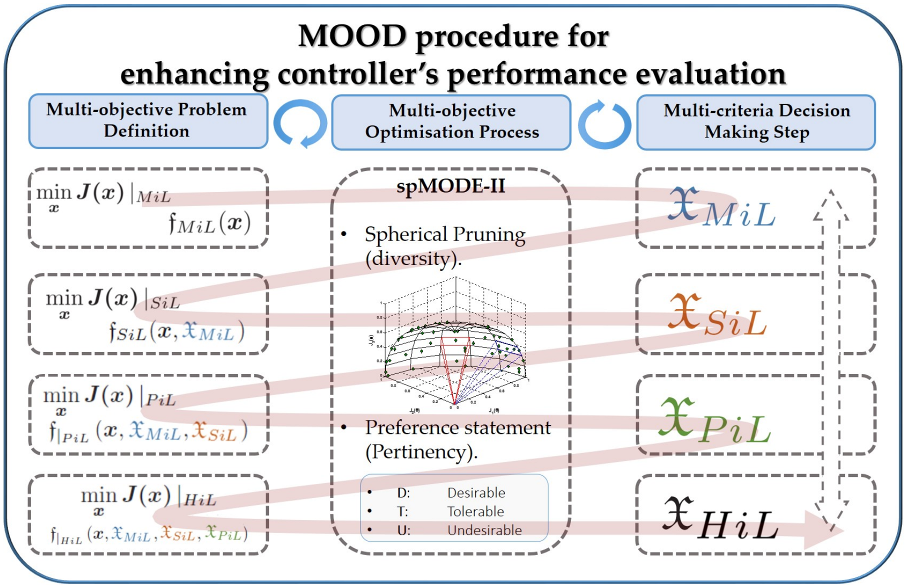

MOOD procedure for
enhancing controller's performance evaluation
Multi-objective Problem Definition
Multi-objective Optimisation Process
Multi-criteria Decision Making Step
$\min_{\boldsymbol{x}} \boldsymbol{J}(\boldsymbol{x})|_{MiL}$
$\mathfrak{f}_{MiL}(\boldsymbol{x})$
$\min_{\boldsymbol{x}} \boldsymbol{J}(\boldsymbol{x})|_{SiL}$
$\mathfrak{f}_{SiL}(\boldsymbol{x}, \mathfrak{X}_{MiL})$
$\min_{\boldsymbol{x}} \boldsymbol{J}(\boldsymbol{x})|_{PiL}$
$\mathfrak{f}_{|PiL}(\boldsymbol{x}, \mathfrak{X}_{MiL}, \mathfrak{X}_{SiL})$
$\min_{\boldsymbol{x}} \boldsymbol{J}(\boldsymbol{x})|_{HiL}$
$\mathfrak{f}_{|HiL}(\boldsymbol{x}, \mathfrak{X}_{MiL}, \mathfrak{X}_{SiL}, \mathfrak{X}_{PiL})$
spMODE-II
• Spherical Pruning (diversity).
• Preference statement (Pertinency).
• D: Desirable
• T: Tolerable
• U: Undesirable
$\mathfrak{X}_{MiL}$
$\mathfrak{X}_{SiL}$
$\mathfrak{X}_{PiL}$
$\mathfrak{X}_{HiL}$

Figure 5: The Multi-objective Optimisation Design Procedure (MOOD) for controller tuning using **X**iL platforms.

4. Methodology implementation

As commented before, **X**iL platforms are very useful for automotive and aeronautic applications. Therefore, in order to validate the usability of this methodology using MOOD procedures, the attitude and navigation control of an aircraft intended to perform way-points supervision tasks will be tuned. Hence, MOPs increase in complexity not only because of the change of platform but also because of the addition of new objectives.

4.1. System description

The aircraft for test and validation is presented in Fig. 6. The main component of the UAV flight system is a Kadett 2400 aircraft manufactured by Graupner. The aircraft has a very lightweight frame and characteristics that make it suitable for the purposes of this research. These characteristics include a 2.4 m wing span, 0.9 m^2 of wing surface, 48.07 $\mathrm{N/m}^2$ wing loading, and 1.65×10^{-2} m^3 of available volume to house control hardware.

During normal flight, the tail rudder, elevators, and ailerons serve as the control surfaces. Propulsion is provided by a brushless alternating current engine supplied by two lithium-ion polymer (LiPo) batteries through a frequency variator. The variator and the servomotors are controlled by pulse width modulated (PWM) command signals.

The flight control station (FCS), housed in an ARM-based microcontroller, hosts the control algorithms. The control loop is closed by a IG500N unit from SBG Systems, that integrates a wide range of sensors, including the accelerometers, gyroscopes, and magnetometers. A Kalman filter fuses the sensor information to estimate position, orientation, linear and angular speed, and acceleration. This same platform was presented with more detail in [Velasco and Nieto, 2014, Velasco-Carrau et al., 2015] together with the results of the first flight tests.

The purpose of the aircraft (for this paper), is to perform a supervision of several way-points. As general approach and without loss of generality, a

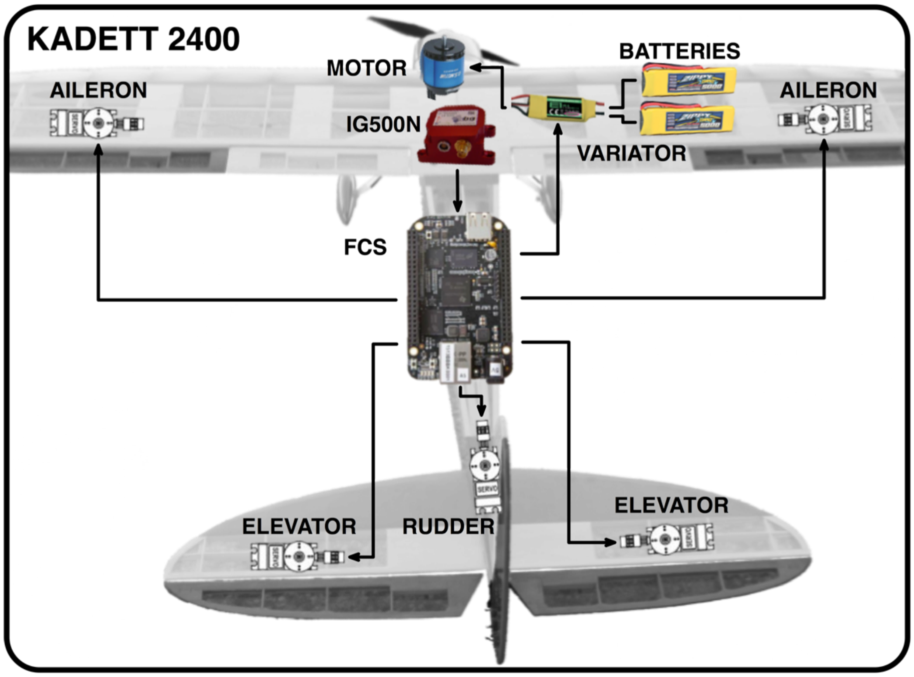


Figure 6: Hardware elements on board experimental platform, the aircraft Kadett 2400

proportional-integral (PI) controller structure is selected to drive each control variable to its set-point.

$$C(s) = k_p \left(1 + \frac{1}{T_i s}\right) \tag{20}$$

where k_p is the proportional gain and T_i the integral time.

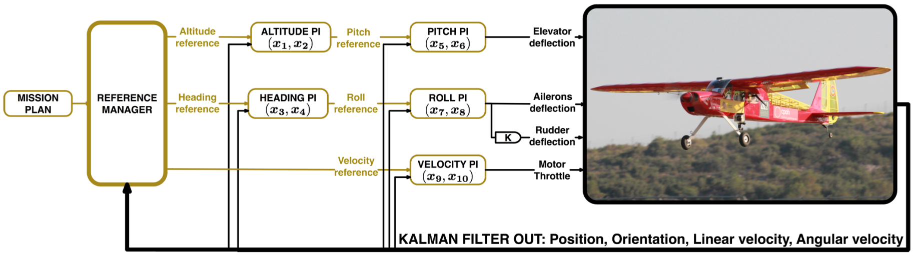


Figure 7: Kadett's control loop structure. Control and guidance over a user-defined mission.

The complete control structure is a set of five PI regulators as the one shown in Fig. 7. Three references (altitude, heading and velocity) are served to three of the five regulators. Those references are calculated by a reference manager based on the mission plan (way-points through which the aircraft must pass) and the aircraft current position. Thrust is directly applied to the motor as result of the velocity's PI, whereas pitch and roll references are respectively obtained from altitude and heading regulators. Finally, pitch and roll PIs generate deflections to be applied on elevators and ailerons. A total of ten variables $[x_1, \ldots, x_{10}]$, five pairs of the form (k_p, T_i), must then be adjusted so that a set of user-defined objectives becomes Pareto optimal.

*4.2. The MOOD-**X**iL definition*

Now, following the methodology, three **X**iL platforms are presented for this example:

MiL: A non-linear model has been identified according to [Velasco and Nieto, 2014, Velasco-Carrau et al., 2015][8]; roughly speaking, multi-objective optimisation techniques have been used to adjust several constants of a first principle model of our aircraft.

SiL: The same non-linear model is used, but here the controller's scripts are added as they will be coded in the FCS; this includes different sampling rates (50 ms for outer loops and 20 ms for inner loops), together with bumpless transfer and anti-windup mechanisms.

HiL: A National Instruments PXI with a real time running model substitutes most of the hardware components; those components are "virtualized" so that the FCS can be added to the loop without any software modification, *i.e.* as it is programmed in a real flight (see Fig. 8).

Given these three platforms, a particularized version of the methodology depicted in Fig. 5 can be posed for the problem at hand. As we can see, only MiL and SiL platforms are used actively in the MOO process whereas all three (MiL, SiL, and HiL) are employed in the MCDM stage. This is because the current HiL platform's infrastructure does not allow for complete integration with the MOO algorithm. The MIL platform will be used with a simple control test in order to identify suitable controllers in the optimisation phase; afterwards, with such results a new optimisation process will be carried out in the SiL platform, where a flight mission will be stated and used in order to evaluate the performance of a given controller. Finally, after this optimisation phase using the simulation model, a final decision making process will be carried out in the HiL platform.

Additionally, a reference controller $\boldsymbol{x}_{ref}$, adjusted via pole placement, is available from previous works. A reference controller is useful for two main reasons: (*i*) it provides a rough first idea of what preferences might be reasonable to ask for; (*ii*) it can be taken as a starting point for the MOO process in the first round.

[8] This model has been implemented in Simulink© Matlab© version R2013a, with ode3(Bogacki-Shampine) solver with a fixed-step size of 1ms.

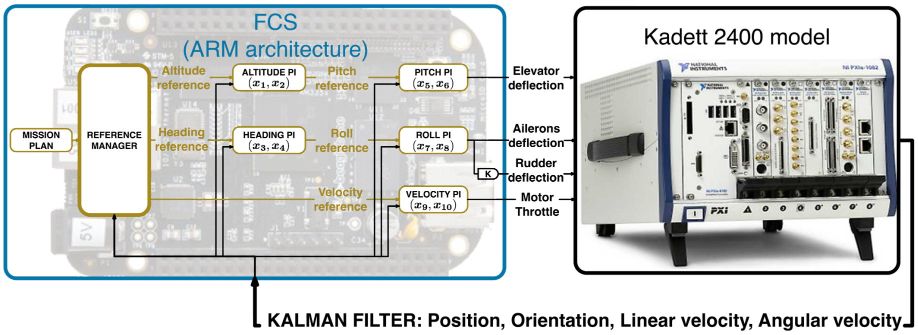


Figure 8: HiL platform for Kadett aircraft.

4.2.1. MOOD at the MiL platform

This MOOD statement will be named hereafter, *TimePiL* (time performance in the loop) and it will be stated as a RDO instance. For this step the MOP definition is:

$$\min_{\boldsymbol{x}} \boldsymbol{J}(\boldsymbol{x})\,|_{MiL} \quad = \quad [J_{M_1}(\boldsymbol{x}), \ldots, J_{M_7}(\boldsymbol{x})] \tag{21}$$

$$s.t.$$

$$\begin{aligned} \mathfrak{f}_{MiL}(\boldsymbol{x}) \quad &\in \quad \mathfrak{P}|_{MiL} \\ \underline{x_i} \leq x_i \quad &\leq \quad \overline{x_i}, i = [1, \ldots, n] \end{aligned}$$

with $\boldsymbol{x} = [k_{p_1}, T_{i_1}, \cdots, k_{p_5}, T_{i_5}]$ and $|k_p| \in [0, 5]$ and $T_i \in]0, 50] \cup \{+\infty\}$; a setpoint response is evaluated for a simultaneous change in heading and altitude. The design objectives are:

- $J_{M_1}(\boldsymbol{x})$: Settling time for yaw at ±2%.
- $J_{M_2}(\boldsymbol{x})$: Settling time for altitude at ±2%.
- $J_{M_3}(\boldsymbol{x})$: Throttle's total variation of control action (Eq. (3)).

- $J_{M_4}(\boldsymbol{x})$: Aileron's total variation of control action (Eq. (3)).
- $J_{M_5}(\boldsymbol{x})$: Elevator's total variation of control action (Eq. (3)).
- $J_{M_6}(\boldsymbol{x})$: Roll's total variation of control action (Eq. (3)).
- $J_{M_7}(\boldsymbol{x})$: Pitch's total variation of control action (Eq. (3)).

Design objectives $J_{M_1}(\boldsymbol{x})$ and $J_{M_2}(\boldsymbol{x})$ are stated for performance ($J_{SR}(\boldsymbol{x})$) while $J_{M_3}(\boldsymbol{x})$ to $J_{M_7}(\boldsymbol{x})$ for robustness ($J_{RBO}(\boldsymbol{x})$), since total variation is a valid measure for robustness [Sanchez and Vilanova, 2013].

The preference matrix $\mathfrak{m}_{MiL}$ is depicted in Table 2. The idea behind preferences $J_{M_3}(\boldsymbol{x})$ to $J_{M_7}(\boldsymbol{x})$ is to provide some meaning to the values obtained from Eq. (3) by posing them relative to the reference controller $\boldsymbol{x}_{ref}$. This idea comes from the fact that the total variation of a control action by itself does not provide the same level of interpretability as, for example, time related indexes $J_{M_1}(\boldsymbol{x})$ and $J_{M_2}(\boldsymbol{x})$, for which preferences are easy to state. This has been previously exposed in [Reynoso-Meza et al., 2016a]. For example, for J_{M_3} in Table 2 it has been defined as a tolerable value up to a 10% of additional control effort of the reference controller $\boldsymbol{x}_{ref}$. The desirable value ranges from a reduction of 10% and 20% of such control action, and the highly desirable value for a reduction up to 30%.

Now $J_{M_1}(\boldsymbol{x})$ to $J_{M_7}(\boldsymbol{x})$ are used in the MOO process. Pareto set $\boldsymbol{\Theta}_P^* |_{MiL}$ and front $\boldsymbol{J}_P^* |_{MiL}$ are obtained for the seven design objectives, however only $J_{M_1}(\boldsymbol{x})$ and $J_{M_2}(\boldsymbol{x})$ are analysed in the MCDM stage; this means that, while all the design objectives are considered in the evolution process, and used to calculate the GPP index in the pruning mechanism of the spMODE-II algorithm, only the first two (the most interpretable) are used to partition the objective space. Optimisations were carried out in a desktop computer, with Intel$^{\mathrm{R}}$ Cor$^{\mathrm{TM}}$ i7-4790 processor, 3.60GHz and RAM memory 32GB; a total of 5000 function evaluations were computed in a time lapse of 6h27m.

The resulting Pareto set and front approximations $\boldsymbol{\Theta}_P^* |_{MiL}$ and $\boldsymbol{J}_P^* |_{MiL}$ are illustrated in Fig. 9. From such Figure, and with the help of the SiL platform, a

Table 2: Preference matrix $\mathfrak{m}_{MiL}$ for the TimePiL ($\boldsymbol{J(x)}\,|_{MiL}$) statement. Five preference ranges have been defined: highly desirable (HD), desirable (D), tolerable (T) undesirable (U) and highly undesirable (HU). For readability purposes, $J_{M_i}(\boldsymbol{x_{ref}})$ has been substituted by $\widehat{J_{M_i}}$.

Preference Matrix						
	← HD	→ ← D	→ ← T	→ ← U	→ ← HU	→
Objective	J_i^0	J_i^1	J_i^2	J_i^3	J_i^4	J_i^5
$J_{M_1}(\boldsymbol{x})$ [s]	10	15	20	25	50	100
$J_{M_2}(\boldsymbol{x})$ [s]	10	20	30	40	80	160
$J_{M_3}(\boldsymbol{x})$ [-]	$0.7 \cdot \widehat{J_{M_3}}$	$0.8 \cdot \widehat{J_{M_3}}$	$0.9 \cdot \widehat{J_{M_3}}$	$1.1 \cdot \widehat{J_{M_3}}$	$1.2 \cdot \widehat{J_{M_3}}$	$1.4 \cdot \widehat{J_{M_3}}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$J_{M_7}(\boldsymbol{x})$ [-]	$0.7 \cdot \widehat{J_{M_7}}$	$0.8 \cdot \widehat{J_{M_7}}$	$0.9 \cdot \widehat{J_{M_7}}$	$1.1 \cdot \widehat{J_{M_7}}$	$1.2 \cdot \widehat{J_{M_7}}$	$1.4 \cdot \widehat{J_{M_7}}$

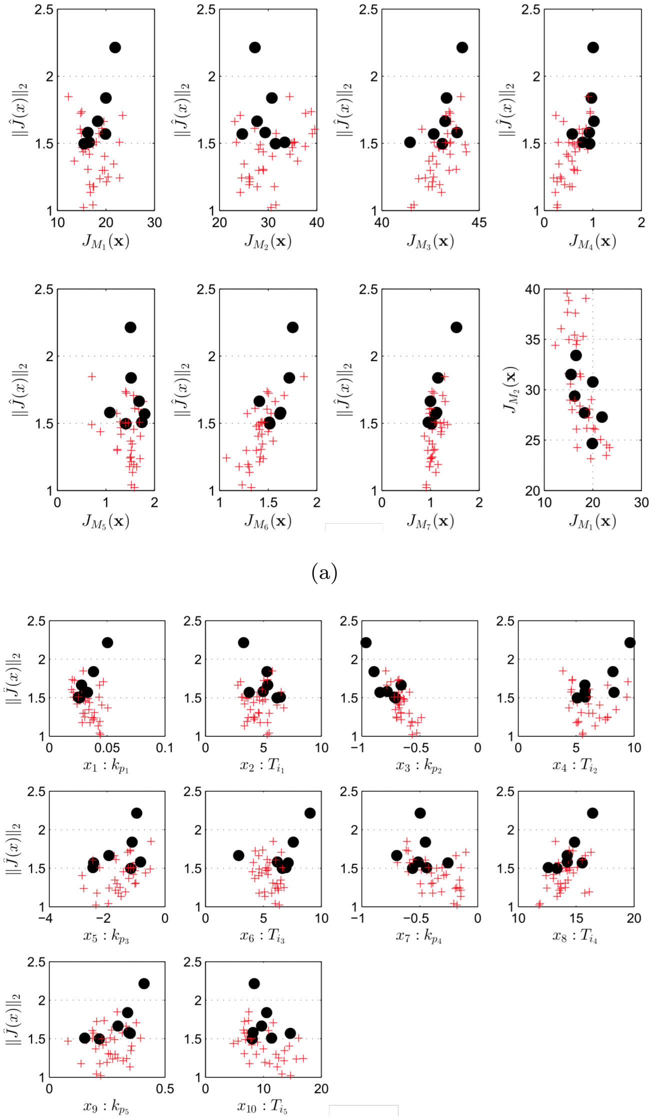
subset $\mathfrak{X}_{MiL}$ (depicted as ◯) is selected for the next step within the MOOD-**X**iL procedure.

4.2.2. MOOD definition at the SiL platform

The performance of the adjusted control structure with a given mission ϕ will be evaluated in this round. For that reason, this MOOD statement will be named hereafter, *MissionPiL* (mission performance in the loop). Such mission comprises the supervision of five different way points in a bounded air space. MissionPiL statement is intended to be a RBDO instance, where reliability on controller's performance to fulfil different flight missions is evaluated. For this purpose, a set Φ of 15 flight missions are defined. Each mission has five randomly distributed and feasible (according to the characteristics of the aircraft) way-points.

Therefore, the MOP definition for this step stays as follows:

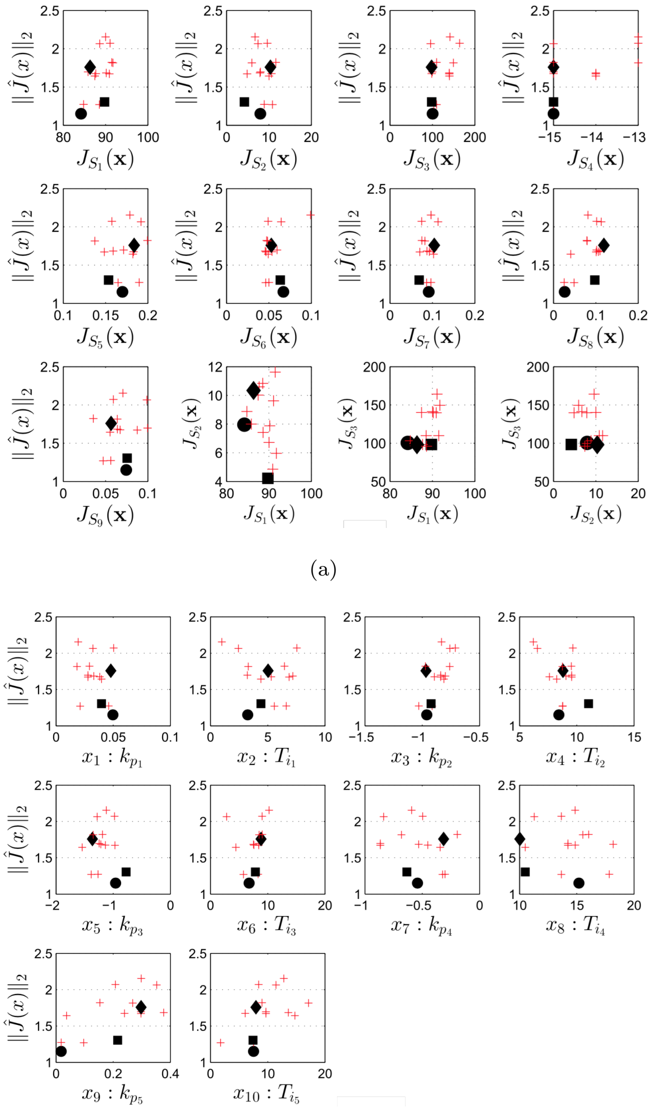
$$\min_{\boldsymbol{x}} \boldsymbol{J(x)}\,|_{SiL} \quad = \quad [J_{S_1}(\boldsymbol{x}), \ldots, J_{S_9}(\boldsymbol{x})] \qquad (22)$$



(a)

(b)

Figure 9: Pareto Front (a) and Pareto Set (b) for TimePiL statement at MiL platform. Solutions marked with ◯ are the subset $\mathfrak{X}_{MiL}$ from $\mathbf{\Theta}_P^*|_{MiL}$ selected for further analysis.



(a)

(b)

Figure 10: Pareto Front (a) and Pareto Set (b) for MissionPiL statement at SiL platform. Solutions marked with ○,□,◊ are the subset $\mathfrak{X}_{SiL}$ from $\mathbf{\Theta}_P^* |_{SiL}$ selected for further analysis.

$$
\begin{array}{rcl}
 & s.t. & \\
\mathfrak{f}_{SiL}(\boldsymbol{x}, \mathfrak{X}_{MiL}) & \in & \mathfrak{P}|_{SiL} \\
\underline{x_i} \leq x_i & \leq & \overline{x_i}, i = [1, \ldots, n]
\end{array}
$$

with $\boldsymbol{x} = [k_{p_1}, T_{i_1}, \cdots, k_{p_5}, T_{i_5}]$ and $k_p \in [0, 5]$ and $T_i \in]0, 50] \cup \{+\infty\}$; The design objectives stated are:

- $J_{S_1}(\boldsymbol{x})$: Median of time required to perform a flight mission ς[s] $\forall \phi \in \Phi$:

$$
\begin{aligned}
J_{S_1}(\boldsymbol{x}) &= median(\boldsymbol{\varsigma}) \qquad (23)\\
\varsigma &= MissionTime(\boldsymbol{x}, \phi), \forall \phi \in \Phi
\end{aligned}
$$

- $J_{S_2}(\boldsymbol{x})$: Median absolute deviation (MAD) of time required [s] to perform a flight mission $\forall \phi \in \Phi$

$$
\begin{aligned}
J_{S_2}(\boldsymbol{x}) &= median(|\boldsymbol{\varsigma} - J_{S_1}(\boldsymbol{x})|) \qquad (24)\\
\varsigma &= MissionTime(\boldsymbol{x}, \phi), \forall \phi \in \Phi
\end{aligned}
$$

- $J_{S_3}(\boldsymbol{x})$: Maximum time required [s] to perform a flight mission $\forall \phi \in \Phi$ (Eq. (9)).
- $J_{S_4}(\boldsymbol{x})$: (Negative) number of successful fight missions.
- $J_{S_5}(\boldsymbol{x})$: Median of roll's total variation of control action per flight time duration (Eq. (25)).
- $J_{S_6}(\boldsymbol{x})$: Median of pitch's total variation of control action per flight time duration (Eq. (25)).
- $J_{S_7}(\boldsymbol{x})$: Median of elevator's total variation of control action per flight time duration (Eq. (25)).

- $J_{S_8}(\boldsymbol{x})$: Median of throttle's total variation of control action per flight time duration (Eq. (25)).
- $J_{S_9}(\boldsymbol{x})$: Median of aileron's total variation of control action per flight time duration (Eq. (25)).

$$\begin{aligned} J_{TV_2}(\boldsymbol{x}) &= median\,(\boldsymbol{\upsilon}) \\ \upsilon &= \left[\int\limits_{t=t_0}^{T_f} \left|\frac{du}{dt}\right|\right] \cdot \left[\frac{1}{\varsigma}\right], \forall \phi \in \Phi \end{aligned} \tag{25}$$

The preference matrix $\mathfrak{m}_{SiL}$ is depicted in Table 3. In this case, preferences values are in accordance with the values observed when testing $\mathfrak{X}_{MiL}$ solutions in the SiL platform. Indeed, when controllers optimised for **M**iL platform are confronted to the more realistic **S**iL platform, they reveal additional information and trade-off among design objectives. When passing through this "experience", the DM reaches better understanding on the capabilities of the overall control structure for the problem at hand, and hence, he/she is able to glimpse the limits, in terms of performance, that the control algorithm can be led to. Therefore, mixing the observed trade-off in the preference matrix $\mathfrak{m}_{SiL}$, aligns with the search of a controller able to satisfy every preference simultaneously. In the same way, initialization bounds are designed given the results of the $\boldsymbol{\Theta}_P^* \,|_{MiL}$. Optimisations were carried out in a desktop computer, with Intel$^{\text{R}}$ Cor$^{\text{TM}}$ i7-4790 processor, 3.60GHz and RAM memory 32GB; a total of 528 function evaluations were computed in a time lapse of 33h03m.

Again the Pareto set and front approximations $\boldsymbol{\Theta}_P^* \,|_{SiL}$ and $\boldsymbol{J}_P^* \,|_{SiL}$ result from the MOO process. They can be visualized in Fig. 10. For example, circle solution has a better performance on J_{S_1} when compared with the square solution; nevertheless, this is at expenses of worsening J_{S_2}. That is, a trade-off between median performance and dispersion when evaluated with the flight missions set. Similar analysis can be performed with the remainder plots in level

Table 3: Preference matrix $\mathfrak{m}_{SiL}$ for the MissionPiL statement. Five preference ranges have been defined: highly desirable (HD), desirable (D), tolerable (T) undesirable (U) and highly undesirable (HU).

Preference Matrix $\mathfrak{m}_{SiL}$						
	← HD	→ ← D	→ ← T	→ ← U	→ ← HU	→
Objective	J_i^0	J_i^1	J_i^2	J_i^3	J_i^4	J_i^5
$J_{S_1}(\boldsymbol{x})$ [s]	60	85	90	95	120	150
$J_{S_2}(\boldsymbol{x})$ [s]	5	7	10	15	30	50
$J_{S_3}(\boldsymbol{x})$ [s]	60	95	120	175	240	300
$J_{S_4}(\boldsymbol{x})$ [-]	-15	-15	-15	-13	-10	-5
$J_{S_5}(\boldsymbol{x})$ [-]	0.10	0.15	0.18	0.20	0.25	0.30
$J_{S_6}(\boldsymbol{x})$ [-]	0.10	0.40	0.50	0.67	0.10	0.15
$J_{S_7}(\boldsymbol{x})$ [-]	0.05	0.08	0.10	0.12	0.15	0.20
$J_{S_8}(\boldsymbol{x})$ [-]	0.02	0.05	0.08	0.12	0.15	0.20
$J_{S_9}(\boldsymbol{x})$ [-]	0.02	0.05	0.08	0.10	0.15	0.20

diagrams. After the MCDM analysis, a subset $\mathfrak{X}_{SiL}$ of three suitable controllers (in their trade-off sense) is selected.

4.2.3. Final MCDM stage at the HiL platform

In order to help with the final decision making, the subset $\mathfrak{X}_{SiL}$ will be analysed in the HiL platform. Although an active MOO is not possible at the moment (and could even be impractical from a computational point of view), manually implementing those pre-selected controllers in the HiL might definitely shed some light on which controller is the best final choice. Five random missions of eight way-points each have been generated for this purpose. The three controllers $\mathfrak{x}_1, \mathfrak{x}_2, \mathfrak{x}_3$ from $\mathfrak{X}_{SiL}$, together with the reference controller, have been implemented in the aircraft FCS. To ensure the most similar initial conditions, each flight path comes preceded by three extra way-points that constitute de initialisation track. Those way-points are only used for homogenisation purposes and hence they are not considered in the performance analysis. Since five sets of data are available for each controller, cost functions $J_{S_i} \forall i = \{1, ..., 9\}$ can be now obtained.

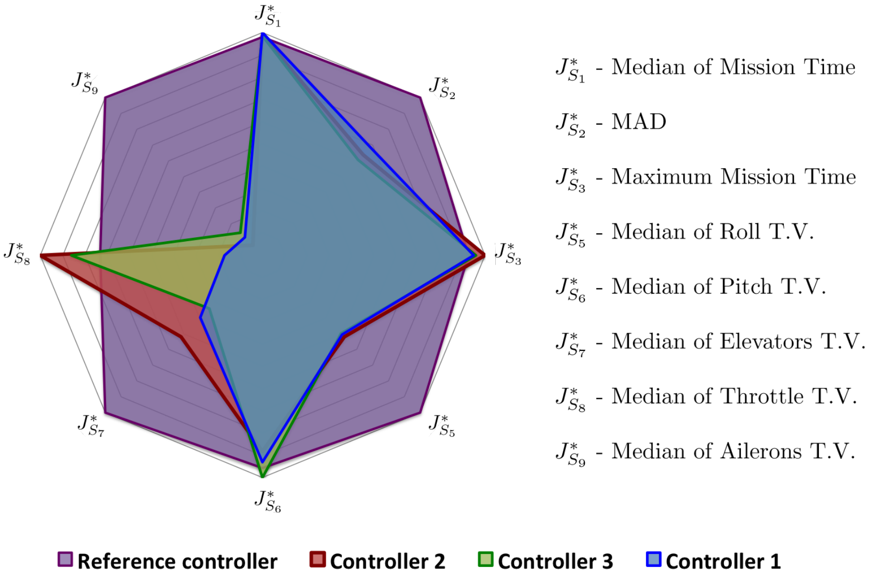


Figure 11: Performance of the selected controllers together with the reference controller in the HiL platform

Figure 11 is a radial representation of the values of J_{S_i} functions with a particularity. This radial representation gives an idea about the trade-off of a particular solution when compared with others. A solution covering all the radial space is a solution which is worsening all design objectives. The smaller the area, the closer to the ideal solution (center of the representation). All indices J_{S_i} in Fig.11 have been scaled over their worst value. This is, if a specific controller shows a value of 1 in one cost index, it means that such controller has the worst performance observed for that particular objective (see Eq. (26)). Additionally, objective 4 has been removed from this analysis because all controllers were able to complete the five flight missions successfully. When observing a radial representation like the one in Fig. 11, the closeness to the nadir solution and utopian solution are directly related to the vertices position of the polygon given by one specific solution[9]. Hence, those controllers whose vertices are closer to the center are to be preferred, since they fulfil each individual objective with a better performance. We can then conclude from Fig. 11 that $\mathfrak{X}_{SiL}$ controllers outperform the reference one. They do not only show smaller costs for most of the objectives, but also have a very similar value in those for which the reference regulator is slightly better.

$$J^*_{S_i}(\boldsymbol{x}) = J_{S_i}(\boldsymbol{x})/J^{max}_{S_i} \tag{26}$$

where

$$\begin{aligned} J^{max}_{S_i} = \quad & \max\{J_{S_i}(\mathfrak{x}_1), J_{S_i}(\mathfrak{x}_2), J_{S_i}(\mathfrak{x}_3), \\ & J_{S_i}(\boldsymbol{x_{ref.}})\} \end{aligned}$$

Now, if the reference controller is removed from the analysis a new perspective is achieved. Thereby, the three controllers $\mathfrak{x}_1$, $\mathfrak{x}_2$ and $\mathfrak{x}_3$, that result from our MOO methodology are illustrated again in Fig. 12. The first fact we can observe is that they are very similar in performance when the first six objectives are taken into account. Differences appear though, when low-level actuators are analysed. As we see, $\mathfrak{x}_1$ has its vertices closer to the center than the other two, being much better when throttle total variation is considered, and with an intermediate performance for elevators and ailerons. Note that smooth behaviours in the engine are directly related to lower energy consumptions. Hence, $\mathfrak{x}_1$ is to be preferred as the final controller. $\mathfrak{x}_2$ behaves very well with ailerons and turns out to be the worst in managing throttle and elevators. On the opposite,

[9] In an approximation set, nadir solution is the objective vector with the worst values in the set; on the opposite, the utopian solution is the objective vector with the better values contained in the set.

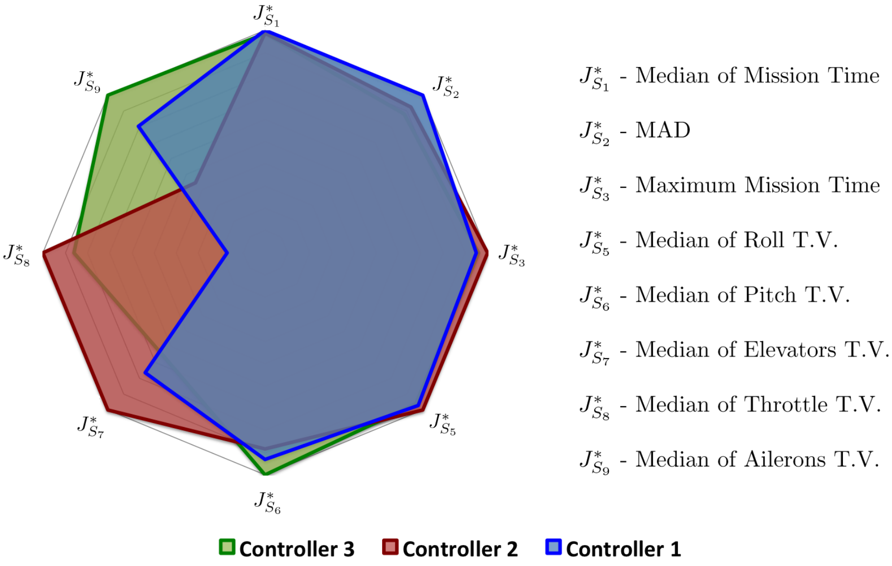


Figure 12: Performance of the selected controllers in the HiL platform

$\mathfrak{x}_3$ is smoother with elevators and throttle, but more aggressive when using the ailerons. Controllers 1 and 2 have been tested in real flight. Since HiL platform includes the aircraft FCS, controllers were tested exactly as they were coded for the simulations. Results from both flights and some conclusions are now exposed in Section 5.

Table 4: Parameter of the selected controllers

	Altitude PI		Heading PI		Pitch PI		Roll PI		Velocity PI	
Controller	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}
1	0.0497	3.2483	-0.9604	8.4343	-0.9591	6.7427	-0.5409	15.1889	0.0179	7.5268
2	0.0398	4.4099	-0.9232	11.0291	-0.7739	7.8579	-0.6339	10.5024	0.2155	7.4387

5. Results and validation on Flight test

Controllers selected for further validation appear in Table 4. A mission of six way-points (plus three from the initialisation track) has been programmed and performed in real flight. A way-point is defined in the 3-D space by its latitude and longitude coordinates and its altitude above sea level. Around each of them, a tolerance cylinder (defined by an altitude error and a plain circle) is placed. Therefore, a way-point is considered to be passed when the aircraft is targeting that way-point and enters its tolerance cylinder. Besides, the flight path has been randomly generated to lie within the volume enclosed inside a 500 m diameter and 150 m height cylinder and does not share any of the way-points previously used in simulations.

Figs. 13 and 14 and Tables 5 and 6 have been included in order to present flight test's results. On one side, Figs. 13 and 14 show the resulting trajectories (cyan line) along the mission path (red line), obtained by controllers 1 and 2 respectively. Reference tracking of altitude and velocity are also represented. The tolerance cylinders are depicted by green circles around the way-points (graphs on the top) and two green lines underneath and above the altitude references (graphs in the middle). To support discussion, Table 5 includes the value of the mean IAE (MIAE) got by each controller for every controlled variable. MIAE gives a sense of the tracking error in average, and is easy to interpret since it has the magnitude of the variable for which it is calculated. The expression to calculate MIAE is

$$J_{MIAE}(\boldsymbol{x}) = \frac{1}{T_m} \int_{t=0}^{T_m} |r_{\boldsymbol{x}}(t) - y_{\boldsymbol{x}}(t)| \, dt \tag{27}$$

where T_m is the mission time, $r_{\boldsymbol{x}}(t)$ is the reference signal and $y_{\boldsymbol{x}}(t)$ is the controlled variable value during a mission in which controller $\boldsymbol{x}$ is active. Finally, Table 6 shows the values of the subset of SiL design objectives that can be calculated for a single real flight. They are:

- $J_{R_1}(\boldsymbol{x})$: time required to perform the flight mission (in representation of $[J_{S_1}(\boldsymbol{x}), J_{S_2}(\boldsymbol{x}), J_{S_3}(\boldsymbol{x})]$).
- $J_{R_2}(\boldsymbol{x})$: roll total variation of control action per flight time duration (in representation of $J_{S_5}(\boldsymbol{x})$).
- $J_{R_3}(\boldsymbol{x})$: pitch total variation of control action per flight time duration (in representation of $J_{S_6}(\boldsymbol{x})$).
- $J_{R_4}(\boldsymbol{x})$: elevators total variation of control action per flight time duration (in representation of $J_{S_7}(\boldsymbol{x})$).
- $J_{R_5}(\boldsymbol{x})$: throttle total variation of control action per flight time duration (in representation of $J_{S_8}(\boldsymbol{x})$).
- $J_{R_6}(\boldsymbol{x})$: ailerons total variation of control action per flight time duration (in representation of $J_{S_9}(\boldsymbol{x})$).

Now, going back to Figs. 13 and 14, we can see that both controllers are able to successfully complete the mission, driving the aircraft through every way-point of the flight path. Every time the UAV enters the tolerance cylinder of its targeted way-point, a new reference is imposed by the reference manager. This process keeps going until the last way-point is reached and the mission is finished. Although the reference manager also imposes different velocities for each path section, accomplishing them is not a requirement of the mission. Several points can be highlighted from Figs. 13 and 14. First, $\mathfrak{x}_1$ makes the aircraft draw smother trajectories with more opened turns; this has an influence in the trajectory length and consequently, in the mission time (for which $\mathfrak{x}_2$ is slightly better). Second, every change in the aircraft orientation strongly disturbs altitude tracking and even more velocity tracking; this is logic, since we are dealing with a coupled non-linear system. Third, while $\mathfrak{x}_1$ is better in

Table 5: Mean IAE indices for the five controlled variables achieved by controller 1 and 2 in a real flight experiment

	Mean IAE of the controlled variables				
Objective	$J(\mathfrak{x}_1)$	$J(\mathfrak{x}_2)$	J^{max}	$J^{*}(\mathfrak{x}_1)$	$J^{*}(\mathfrak{x}_2)$
Mean IAE Roll	0.06	0.08	0.08	**0.73**	1.00
Mean IAE Pitch	0.03	0.04	0.04	**0.91**	1.00
Mean IAE Heading	0.71	0.68	0.71	1.00	**0.96**
Mean IAE Altitude	4.17	4.98	4.98	**0.84**	1.00
Mean IAE Velocity	2.94	1.51	2.94	1.00	**0.51**

sticking to the altitude reference, it is significantly worse than $\mathfrak{x}_2$ when trying to follow velocity references. Finally, the reader should note that velocity tracking is not within the design objectives, and hence, no cost index explicitly accounts for velocity tracking performance; even so, both controllers are able to drive the system towards the velocity references; this is interesting, and might be explained under the assumption that it could be a correlation between velocity tracking and the success in a flight mission.

If Fig. 7 is looked, three control flows are observed. On the top, the reference manager proposes altitude references to be achieved by the altitude PI, which in turn proposes pitch references to be accomplished by the pitch PI acting over elevators deflections. If we now return to Tables 5 and 6, it can be observed that $\mathfrak{x}_1$ is softer than $\mathfrak{x}_2$ both managing elevators and proposing pitch references (66% in the case of J_{R_4} and 26% in the case of J_{R_3}); and even so, $\mathfrak{x}_1$ is capable of outperforming $\mathfrak{x}_2$ by obtaining 9% and 16% smaller MIAEs for pitch and altitude respectively. This last data confirms the better behaviour in altitude tracking already observed in Figs. 13 and 14. The second control flow shown in Fig. 7 is the lateral control, where the reference manager imposes heading

Table 6: Objective function values achieved by controller 1 and 2 in a real flight experiment

	Cost indices for real flight test				
Objective	$J(\mathfrak{x}_1)$	$J(\mathfrak{x}_2)$	J^{max}	$J^*(\mathfrak{x}_2)$	$J^*(\mathfrak{x}_2)$
J_{R_1} - Mission Time	110.13	104.84	110.13	1.00	**0.95**
J_{R_2} - Roll Total Variation	0.17	0.21	0.21	**0.80**	1.00
J_{R_3} - Pitch Total Variation	0.06	0.08	0.08	**0.74**	1.00
J_{R_4} - Elevators Total Variation	0.15	0.33	0.33	**0.44**	1.00
J_{R_5} - Throttle Total Variation	0.04	0.24	0.24	**0.15**	1.00
J_{R_6} - Ailerons Total Variation	0.16	0.11	0.16	1.00	**0.69**

references to the heading PI, which in turn proposes roll references tracked by the roll PI acting over ailerons deflections. In this case $\mathfrak{x}_2$ is 31% smoother with the ailerons ($J^*_{R_6}(\mathfrak{x}_2) = 0.69$) and slightly better in heading tracking (4% smaller heading MIAE), while $\mathfrak{x}_1$ achieves a 26% smaller roll total variation J_{R_3}, and 27% better roll MIAE; it means that the heading PI in $\mathfrak{x}_1$ is softer proposing control actions and, at the same time, $\mathfrak{x}_1$'s roll PI is better in following them, at the cost of a higher usage of the ailerons. This is again aligned with what was observed in Figs. 13 and 14, where controller 2 exhibits closer turns than controller 1. Finally, Fig. 13 showed that $\mathfrak{x}_1$ is slower in converging velocity to its set-point; this fact is also supported by the values on Table 5 where velocity's MIAE of controller 2 is 49% smaller than that of controller 1. However, Table 6 evidences that the throttle total variation of $\mathfrak{x}_1$ is 85% smaller ($J^*_{R_5}(\mathfrak{x}_1) = 0.15$) than the throttle total variation obtained by $\mathfrak{x}_2$.

Fig. 15 has been included to derive final conclusions. Two radial graphs are present in that figure. On the top, (Fig. 15 (a)) the values of the six cost indices $\{J_{R_1}, ..., J_{R_6}\}$ are represented both for $\mathfrak{x}_1$ and $\mathfrak{x}_2$. The values obtained from the HIL platform for controllers 1 and 2 have been again represented in Fig. 15 (b). Two remarks must be mentioned. First, note that only six over the nine indices are represented; this is because only the six SiL design objectives that can be directly compared to the real flight indices J_{R_i}, have been included in the graph. Second, those indices have been re-scaled, taking now into account only the values obtained by $\mathfrak{x}_1$ and $\mathfrak{x}_2$. Thus, Fig. 15 (b) can be seen as a version of Fig. 12 with a smaller amount of objectives and without $\mathfrak{x}_3$. Something remarkable from Fig. 15 is the resemblance among graphs (a) and (b). Although obviously not equal, the shapes that one controller gets for the HiL platform and for the real flight are quite similar. This is a significant fact because clearly shows the importance of having an accurate model of the system and reliable **X**iL platforms for the RBDO statement in the MOOD procedure. And there is where the strength of our methodology resides. With a good dynamic model, a designer passing through every step gains real knowledge on the design problem and is able to understand what should be optimised and how to do it. In addition, a realistic HiL platform (only achievable with a good dynamic model) assures safety in the hop from simulation to real experimentation. All this is finally translated in a controller that is well designed and behaves in reality as expected from simulations. As a final comment, Fig. 15 (a) proves that $\mathfrak{x}_1$ should be preferred over $\mathfrak{x}_2$, as its vertices are closer to the center. This was already concluded is Section 4, and evidences that a final MCDM stage with several simulations in the HiL platform is indeed a good practice.

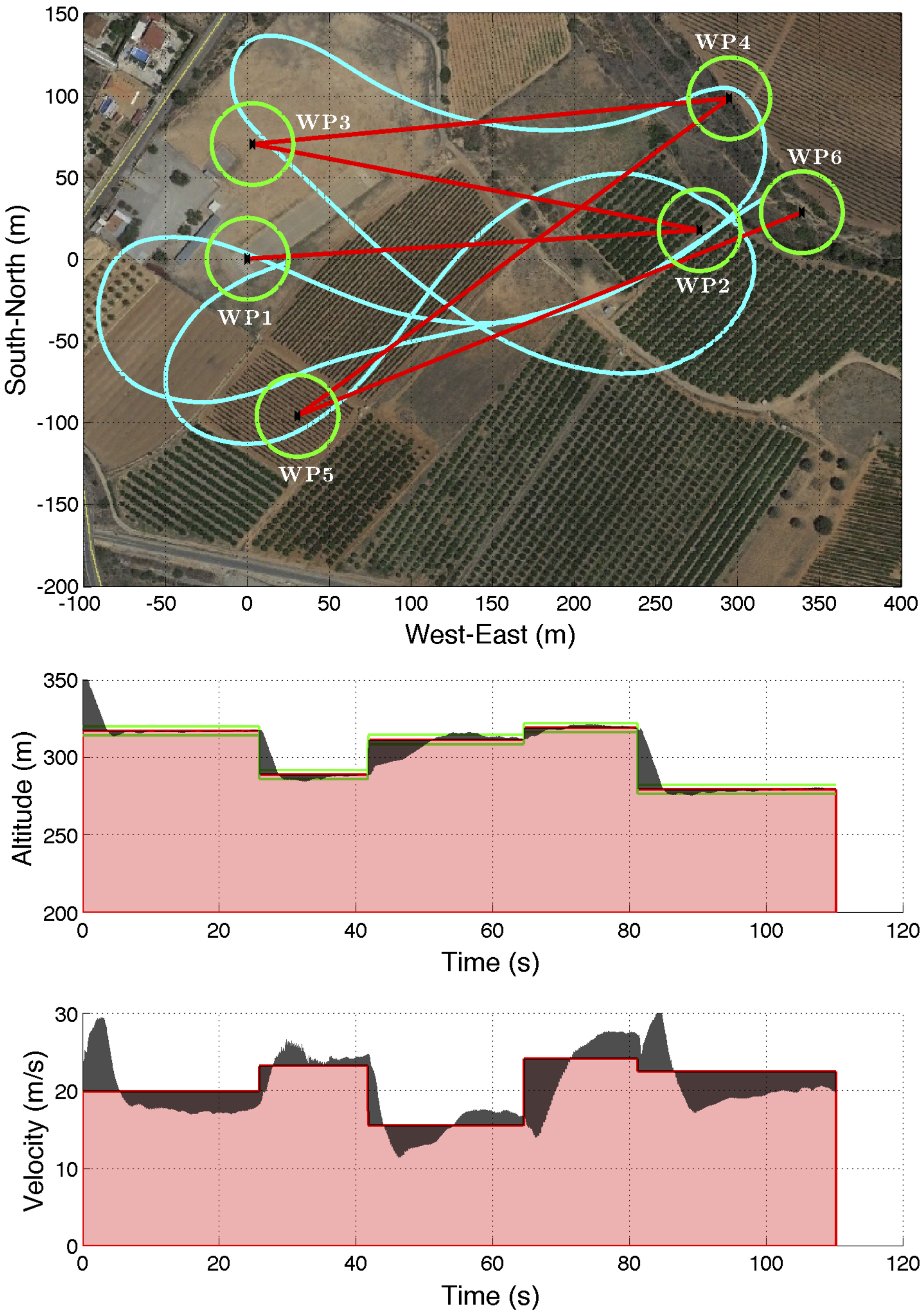


Figure 13: Performance of the Controller 1 in the Kadett in a real flight mission.

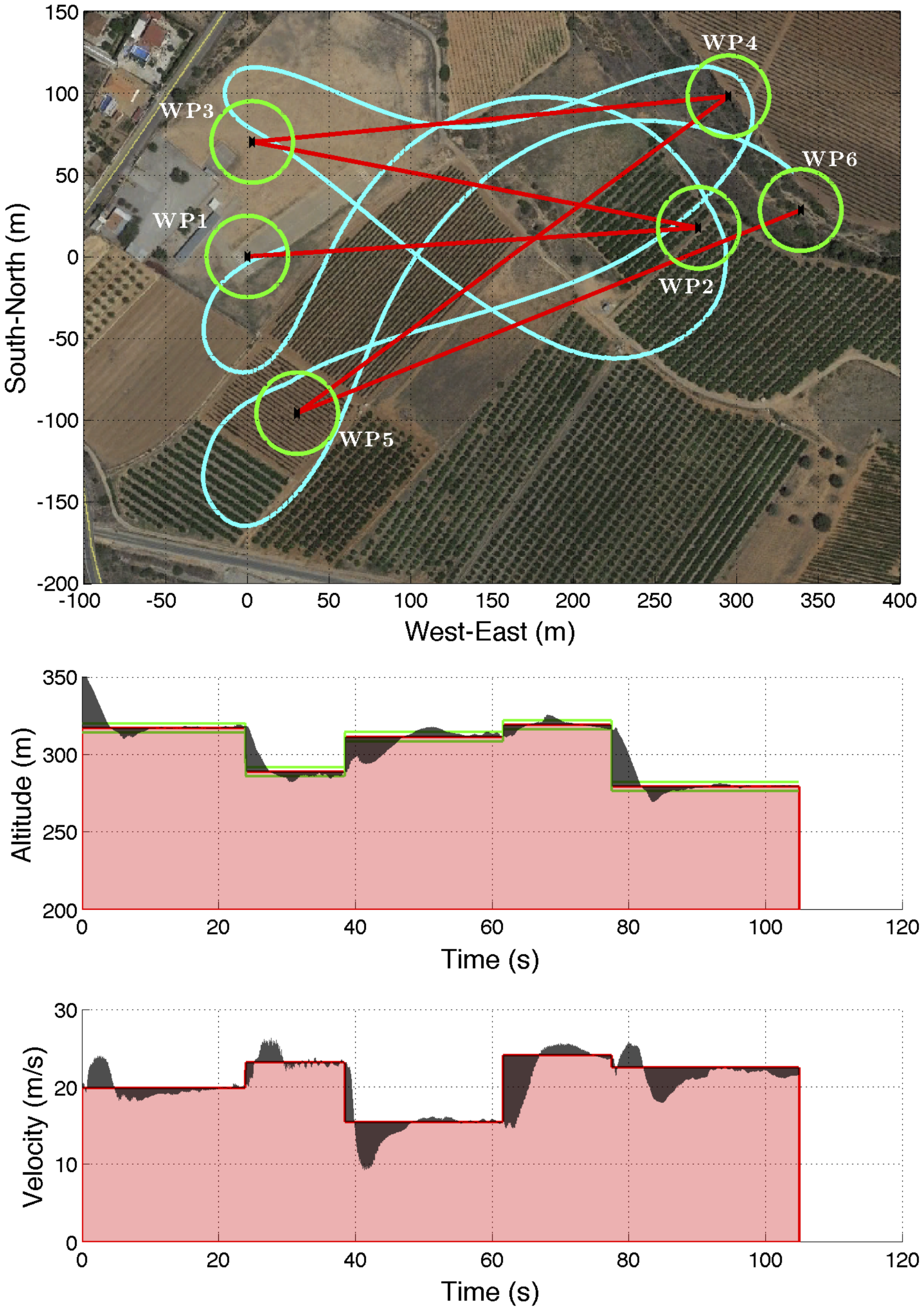
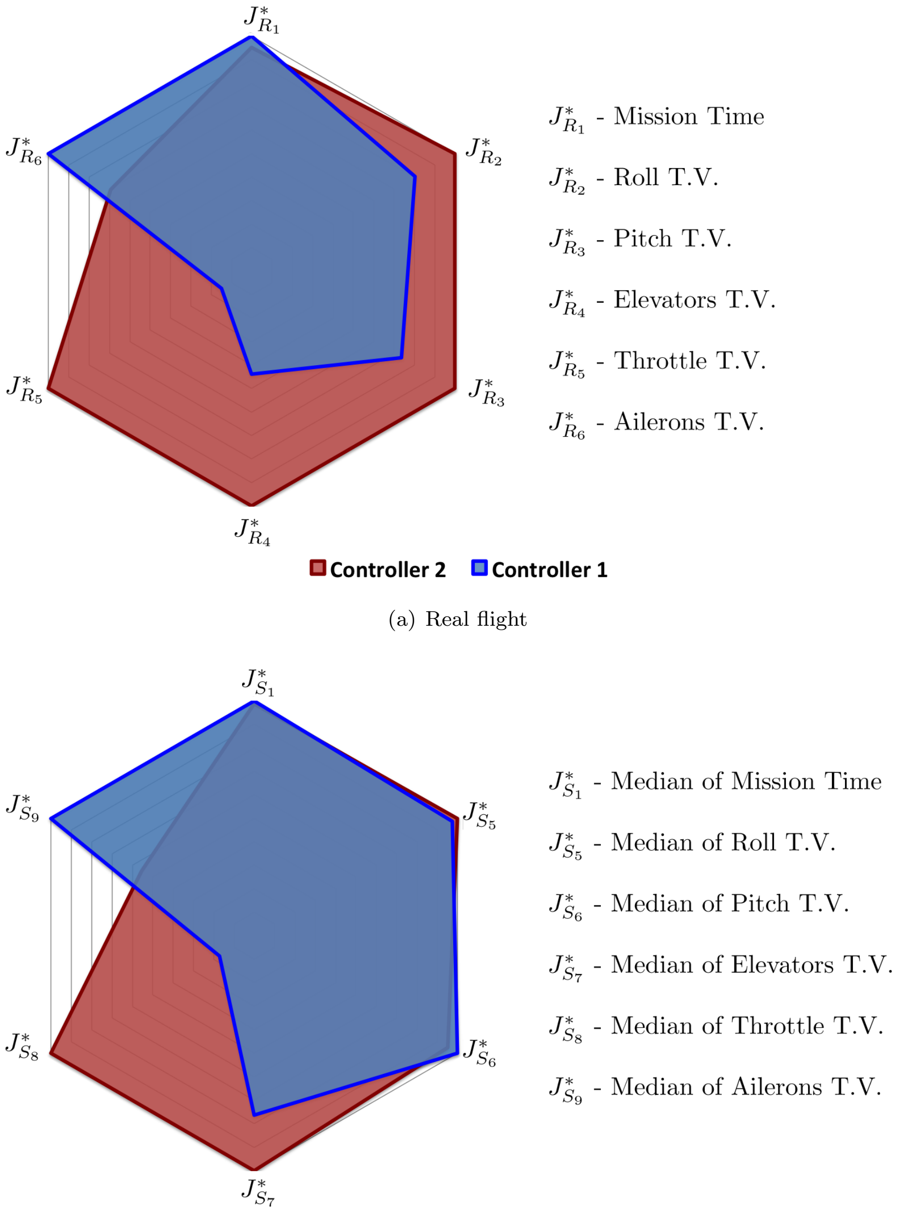


Figure 14: Performance of the Controller 2 in the Kadett in a real flight mission.



(a) Real flight

(b) HIL flights

Figure 15: Scaled Pareto front of the selected controllers during Real flight

6. Conclusions and future work

A systematic approach to enhance controller performance evaluation and design has been presented throughout this paper. Multi-objective optimisation is used in conjunction with different simulation platforms in order to provide the integrative framework on which the methodology is based. Thereby, MiL, SiL, PiL and HiL platforms (or a subset of them) can be employed in successive multi-objective optimisations in order to gain better understanding of the problem as the process moves forward. As we saw, the information obtained in previous stages is used in two directions when a new optimisation is to be posed: (*i*) more meaningful objectives can be stated and (*ii*) preferences and solution constraints can be better delimited. On one side, (*i*) generally leads to more complex MOPs, what obviously increases the computational burden. On the other side, (*ii*) reduces the search space helping, therefore, to decrease that complexity during the optimisation stage. Both (*i*) and (*ii*) go in the direction of getting more adequate solutions (in the sense of what the designer prefers) for the problem at hand.

As a demonstrator, a UAV system with a predefined control structure has been presented. That structure is formed by a total of five PI controllers that perform attitude control and navigation tasks. The ten PI parameters had to be tuned so that the resulting controller was able to drive the aircraft in the supervision of several way-points. Section 4 showed how the methodology can be adapted to this specific problem, in accordance to the available simulation platforms. As the process moved forward, reliability objectives, where several missions were actively accomplished inside the MOO, were posed. At the same time, a better understanding of the problem allowed the designer to refine preferences' matrices, as well as optimisation limits. A final MCDM stage comprising several missions inside the HiL platform has been performed. In that last phase, a controller was chosen from the rest.

To validate the final choice, a real flight mission has been carried out. The chosen controller and a second one within the final Pareto set have been tested exactly as they were coded for HiL simulations. In Section 5 experimental results confirmed that the selected controller was as good accomplishing flight missions as the MCDM stage suggested. This evidenced that a realistic model of the system is of great importance to obtain a controller that behaves as the designer expects. A second conclusion is that, even when the HiL platform cannot be actively used in the MOO process, including it in the final MCDM stage is a good practice.

Several new flight paths can be taken as future works. First, adapting the HiL simulation platform so that it can be actively used in the MOO process could be of great interest (although practicality must be had in mind). Second, the authors would like to study the applicability of the technique on the design of multi-variable controllers. Indeed, tuning weighting matrices for linear quadratic regulators or adjusting design parameters in model based predictive controllers can be something tricky when the system comprises many states and inputs. Third, applying this technique to systems with modelled parametric uncertainties can be a good practice to achieve a certain level of robustness.

Acknowledgement

The authors would like to acknowledge the Spanish Ministry of Economy and Competitiveness for providing funding through the project DPI2015-71443-R and the grant BES-2012-056210. This work has been partially supported by the National Council of Scientific and Technological Development of Brazil (CNPq) through the BJT/304804/2014-2 and PQ-2/304066/2016-8 grants.

References

A. Alfi, A. Shokrzadeh, and M. Asadi. Reliability analysis of h-infinity control for a container ship in way-point tracking. *Applied Ocean Research*, 52:309–316, 2015.

K. J. Åström and T. Hägglund. The future of pid control. *Control engineering practice*, 9(11):1163–1175, 2001.

X. Blasco, J. M. Herrero, J. Sanchis, and M. Martínez. A new graphical visualization of n-dimensional Pareto front for decision-making in multiobjective optimization. *Information Sciences*, 178(20):3908 – 3924, 2008.

X. Blasco, S. García-Nieto, and G. Reynoso-Meza. Autonomous trajectory control of a quadricopter vehicle. simulation and evaluation. *Revista Iberoamericana de Automática e Informática Industrial*, 9(2):194 – 199, 2012.

P. Bonissone, R. Subbu, and J. Lizzi. Multicriteria decision making (MCDM): a framework for research and applications. *IEEE Computational Intelligence Magazine*, 4(3):48 –61, 2009. ISSN 1556-603X. doi: 10.1109/MCI.2009.933093.

R. Cela and M. Bollaín. New cluster mapping tools for the graphical assessment of non-dominated solutions in multi-objective optimization. *Chemometrics and Intelligent Laboratory Systems*, 114(0):72 – 86, 2012. ISSN 0169-7439. doi: 10.1016/j.chemolab.2012.03.005.

C. Choi and W. Lee. Analysis and compensation of time delay effects in hardware-in-the-loop simulation for automotive pmsm drive system. *Industrial Electronics, IEEE Transactions on*, 59(9):3403–3410, Sept 2012.

R. F. Coelho. Probabilistic dominance in multiobjective reliability-based optimization: theory and implementation. *IEEE Transactions on Evolutionary Computation*, 19(2):214–224, 2015.

C. A. C. Coello and G. B. Lamont. *Applications of Multi-Objective evolutionary algorithms*. World scientific publishing, advances in natural computation vol. 1 edition, 2004.

S. Das and P. N. Suganthan. Differential evolution: A survey of the state-of-the-art. *IEEE Transactions on Evolutionary Computation*, PP(99):1 –28, 2010. ISSN 1089-778X. doi: 10.1109/TEVC.2010.2059031.

K. Deb, S. Bandaru, D. Greiner, A. Gaspar-Cunha, and C. C. Tutum. An integrated approach to automated innovization for discovering useful design principles: Case studies from engineering. *Applied Soft Computing*, 15(0):42 – 56, 2014. ISSN 1568-4946.

D. M. Frangopol and K. Maute. Life-cycle reliability-based optimization of civil and aerospace structures. *Computers & structures*, 81(7):397–410, 2003.

O. Garpinger, T. Hägglund, and K. J. Åström. Performance and robustness trade-offs in pid control. *Journal of Process Control*, 24(5):568–577, 2014.

D. Gladwin, P. Stewart, J. Stewart, R. Chen, and E. Winward. Improved decision support for engine-in-the-loop experimental design optimization. *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, 224(2):201–218, 2010.

A. Hajiloo, N. Nariman-zadeh, and A. Moeini. Pareto optimal robust design of fractional-order pid controllers for systems with probabilistic uncertainties. *Mechatronics*, 22(6):788 – 801, 2012. ISSN 0957-4158. doi: https://doi.org/

10.1016/j.mechatronics.2012.04.003. Special Issue on Intelligent Mechatronics (LSMS2010 & ICSEE2010).

K. Hassani and W.-S. Lee. Multi-objective design of state feedback controllers using reinforced quantum-behaved particle swarm optimization. *Applied Soft Computing*, 41:66 – 76, 2016. ISSN 1568-4946. doi: https://doi.org/10.1016/j.asoc.2015.12.024.

A. Inselberg. The plane with parallel coordinates. *The Visual Computer*, 1: 69–91, 1985. ISSN 0178-2789.

A. Inselberg. *Parallel Coordinates: Visual Multidimensional Geometry and Its Applications*. Springer, 2009. doi: DOI:10.1007/978-0-387-68628-8.

S. Jeon and S. Jung. Hardware-in-the-loop simulation for the reaction control system using pwm-based limit cycle analysis. *Control Systems Technology, IEEE Transactions on*, 20(2):538–545, March 2012.

T. Koetje, M. Braae, and M. Tsoeu. Multi-objective performance evaluation of controllers for a thermal process. In *Emerging Trends in Computing, Informatics, Systems Sciences, and Engineering*, pages 131–145. Springer, 2013.

B. Lu, X. Wu, H. Figueroa, and A. Monti. A low-cost real-time hardware-in-the-loop testing approach of power electronics controls. *Industrial Electronics, IEEE Transactions on*, 54(2):919–931, April 2007.

A. Messac. Physical programming-effective optimization for computational design. *AIAA journal*, 34(1):149–158, 1996.

G. R. Meza, X. B. Ferragud, J. S. Saez, and J. M. H. Durá. *Controller tuning with evolutionary multiobjective optimization: A holistic multiobjective optimization design procedure*, volume 85. Springer, 2016.

G. R. Meza, X. B. Ferragud, J. S. Saez, and J. M. H. Durá. Background on multiobjective optimization for controller tuning. In *Controller Tuning with Evolutionary Multiobjective Optimization*, pages 23–58. Springer, 2017.

K. M. Miettinen. *Nonlinear multiobjective optimization*. Kluwer Academic Publishers, 1998.

S. Moberg, J. Öhr, and S. Gunnarsson. A benchmark problem for robust feedback control of a flexible manipulator. *Control Systems Technology, IEEE Transactions on*, 17(6):1398–1405, 2009.

H. Panagopoulos and K. Åström. Pid control design and h-infinity loop shaping. *International Journal of Robust and Nonlinear Control*, 10(15):1249–1261, 2000. doi: 10.1002/1099-1239(20001230)10:15⟨1249::AID-RNC514⟩3.0.CO;2-7.

A. Perera, R. Attalage, K. Perera, and V. Dassanayake. A hybrid tool to combine multi-objective optimization and multi-criterion decision making in designing standalone hybrid energy systems. *Applied Energy*, 107:412 – 425, 2013. ISSN 0306-2619. doi: https://doi.org/10.1016/j.apenergy.2013.02.049.

S. Pourzeynali, S. Salimi, and H. E. Kalesar. Robust multi-objective optimization design of tmd control device to reduce tall building responses against earthquake excitations using genetic algorithms. *Scientia Iranica*, 20(2):207 – 221, 2013. ISSN 1026-3098. doi: https://doi.org/10.1016/j.scient.2012.11.015.

G. Reynoso-Meza, J. Sanchis, X. Blasco, and M. Martínez. Multiobjective design of continuous controllers using differential evolution and spherical pruning. In C. D. C. et Al., editor, *Applications of Evolutionary Computation, Part I*, volume LNCS 6024, pages 532–541. Springer-Verlag, 2010.

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*, 221:124 – 141, 2013a. ISSN 0020-0255. doi: 10.1016/j.ins.2012.09.049.

G. Reynoso-Meza, S. García-Nieto, J. Sanchis, and X. Blasco. Controller tuning using multiobjective optimization algorithms: a global tuning framework. *IEEE Transactions on Control Systems Technology*, 21(2):445–458, 2013b.

G. Reynoso-Meza, J. Sanchis, X. Blasco, and S. García-Nieto. Multiobjective evolutionary algortihms for multivariable PI controller tuning. *Applied Soft Computing*, 24:341 – 362, 2014a.

G. Reynoso-Meza, J. Sanchis, X. Blasco, and M. Martínez. Controller tuning using evolutionary multi-objective optimisation: current trends and applications. *Control Engineering Practice*, 1:58 – 73, 2014b.

G. Reynoso-Meza, J. Sanchis, X. Blasco, and R. Z. Freire. Evolutionary multi-objective optimisation with preferences for multivariable pi controller tuning. *Expert Systems with Applications*, 51:120–133, 2016a.

G. Reynoso-Meza, J. Sanchis, X. Blasco, and M. Martínez. Preference driven multi-objective optimization design procedure for industrial controller tuning. *Information Sciences*, 339:108–131, 2016b.

T. Samad and G. Stewart. Perspectives on innovation in control systems technology: compatibility with industry practices. *Control Systems Technology, IEEE Transactions on*, 21(2):284–288, 2013.

H. S. Sanchez and R. Vilanova. Multiobjective tuning of pi controller using the nnc method: Simplified problem definition and guidelines for decision making. In *Emerging Technologies & Factory Automation (ETFA), 2013 IEEE 18th Conference on*, pages 1–8. IEEE, 2013.

H. S. Sánchez, F. Padula, A. Visioli, and R. Vilanova. Tuning rules for robust fopid controllers based on multi-objective optimization with fopdt models. *ISA transactions*, 66:344–361, 2017.

R. F. Stengel and C. I. Marrison. Robustness of solutions to a benchmark control problem. *Journal of guidance, control, and dynamics*, 15(5):1060–1067, 1992.

P. Stewart, D. Stone, and P. Fleming. Design of robust fuzzy-logic control systems by multi-objective evolutionary methods with hardware in the loop. *Engineering Applications of Artificial Intelligence*, 17(3):275–284, 2004.

R. Storn and K. Price. Differential evolution: A simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, 11:341 – 359, 1997.

T. Tušar and B. Filipič. Visualization of pareto front approximations in evolutionary multiobjective optimization: A critical review and the prosection method. *IEEE Transactions on Evolutionary Computation*, 19, 2014. ISSN 1089-778X. doi: 10.1109/TEVC.2014.2313407.

M. A. Valdebenito and G. I. Schuëller. A survey on approaches for reliability-based optimization. *Structural and Multidisciplinary Optimization*, 42(5): 645–663, Nov 2010. doi: 10.1007/s00158-010-0518-6.

J. Velasco and S. G. Nieto. Unmanned aerial vehicles model identification using multi-objective optimization techniques. *IFAC Proceedings Volumes*, 47(3): 8837–8842, 2014.

J. Velasco-Carrau, S. García-Nieto, J. V. Salcedo, and R. H. Bishop. Multi-objective optimization for wind estimation and aircraft model identification. *Journal of Guidance, Control, and Dynamics*, 39(2):372–389, 2015.

Z. Wang, Q. Su, and X. Luo. A novel htd-cs based pid controller tuning method for time delay continuous systems with multi-objective and multi-constraint optimization. *Chemical Engineering Research and Design*, 115:98 – 106, 2016. ISSN 0263-8762. doi: https://doi.org/10.1016/j.cherd.2016.09.025.

A. White, G. G. Zhu, and J. Choi. Hardware-in-the-loop simulation of robust gain-scheduling control of port-fuel-injection processes. *Control Systems Technology, IEEE Transactions on*, 19(6):1433–1443, 2011.

P. Woźniak. Preferences in multi-objective evolutionary optimisation of electric motor speed control with hardware in the loop. *Applied Soft Computing*, 11 (1):49–55, 2011.

D. Yurchenko and P. Alevras. Stability, control and reliability of a ship crane payload motion. *Probabilistic Engineering Mechanics*, 38:173–179, 2014.

A.-A. Zamani, S. Tavakoli, and S. Etedali. Fractional order pid control design for semi-active control of smart base-isolated structures: A multi-objective cuckoo search approach. *ISA Transactions*, 67:222 – 232, 2017. ISSN 0019-0578. doi: https://doi.org/10.1016/j.isatra.2017.01.012.