Engineering Applications of Artificial Intelligence 64 (2017) 52–66

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai

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

ARTICLE INFO

Article history:
Received 11 May 2016
Received in revised form
11 March 2017
Accepted 9 May 2017

MSC:
00-01
99-00

Keywords:
Multi-objective optimisation
Controller tuning
Hardware in the loop
Flight control system.

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 always 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 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.

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 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

* Corresponding authors.
E-mail addresses: jevecar@upvnet.upv.es (J. Velasco Carrau), g.reynosomeza@pucpr.br (G. Reynoso-Meza), sergarro@isa.upv.es (S. García-Nieto), xblasco@isa.upv.es (X. Blasco).

http://dx.doi.org/10.1016/j.engappai.2017.05.005
0952-1976/

practical sense. According to this, assessing robustness and reliability 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., 2014a) 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

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

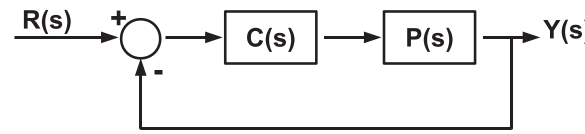

Fig. 1. Basic control loop.

- 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., 2014a) for design objectives are:

- Maximum value of sensitivity function

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

- Integral of the absolute error value

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

- Total variation of control action

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

where $r(t)$, $y(t)$, $u(t)$ are the reference, measured variable and control action in time t. Eqs. (2) and (3) are commonly used for setpoint response and load disturbance, while for example Eq. (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.