Expert Systems with Applications 39 (2012) 8968–8974

Contents lists available at SciVerse ScienceDirect

Expert Systems with Applications

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

Tuning of PID controller based on a multiobjective genetic algorithm applied to a robotic manipulator

Helon Vicente Hultmann Ayala[a], Leandro dos Santos Coelho[b,*]

[a] Electricity Department (DPEL), Power Systems Division (DVSE), LACTEC – Institute of Technology for Development, BR 116 KM 98, n. 8813, Zip code 81531-980 Curitiba, Parana, Brazil

[b] Industrial and Systems Engineering Graduate Program, Pontifical Catholic University of Parana, Imaculada Conceicao, 1155, Zip code 80215-910 Curitiba, PR, Brazil

ARTICLE INFO

Keywords:
Multiobjective optimization
Control systems
Robotic manipulator
Genetic algorithm
PID control

ABSTRACT

Most controllers optimization and design problems are multiobjective in nature, since they normally have several (possibly conflicting) objectives that must be satisfied at the same time. Instead of aiming at finding a single solution, the multiobjective optimization methods try to produce a set of good trade-off solutions from which the decision maker may select one. Several methods have been devised for solving multiobjective optimization problems in control systems field. Traditionally, classical optimization algorithms based on nonlinear programming or optimal control theories are applied to obtain the solution of such problems. The presence of multiple objectives in a problem usually gives rise to a set of optimal solutions, largely known as Pareto-optimal solutions. Recently, Multiobjective Evolutionary Algorithms (MOEAs) have been applied to control systems problems. Compared with mathematical programming, MOEAs are very suitable to solve multiobjective optimization problems, because they deal simultaneously with a set of solutions and find a number of Pareto optimal solutions in a single run of algorithm. Starting from a set of initial solutions, MOEAs use iteratively improving optimization techniques to find the optimal solutions. In every iterative progress, MOEAs favor population-based Pareto dominance as a measure of fitness. In the MOEAs context, the Non-dominated Sorting Genetic Algorithm (NSGA-II) has been successfully applied to solving many multiobjective problems. This paper presents the design and the tuning of two PID (Proportional–Integral–Derivative) controllers through the NSGA-II approach. Simulation numerical results of multivariable PID control and convergence of the NSGA-II is presented and discussed with application in a robotic manipulator of two-degree-of-freedom. The proposed optimization method based on NSGA-II offers an effective way to implement simple but robust solutions providing a good reference tracking performance in closed loop.

1. Introduction

The desired goals in modern control engineering design can be addressed as the resolution of an optimization problem. In general, the state problem is defined as a single objective function expressed as a mathematical function based on some criteria. In many cases, a designer needs to make tradeoffs between disparate and conflicting design objectives. At this point, the definition of a multiobjective optimization (MO) problem can be interesting, given that the field of MO defines the art and science of making such decisions. The MO techniques offer advantages when compared with single objective optimization techniques because they may produce a solution with different trade-offs among different individual objectives, so that the designer can select the best final solution (Martínez, García-Nieto, Sanchis, & Blasco, 2009).

Generally speaking, MO does not restrict to find a unique single solution of a given problem, but a set of solutions called *non-dominated solutions*. Each solution in this set is said to be a *Pareto optimum*, and when they are plotted in the objective space they are collectively known as the *Pareto front*. Obtaining the Pareto front of a given MO problem is the main goal of multiobjective optimization. In this context, Pareto optimality is a measure of efficiency in multicriteria and multiobjective situations.

Most optimization problems in control systems involve the optimization of more than one objective function (Aggelogiannaki, Sarimveis, & Bafas, 2004; Ayala & Coelho, 2008; Carvalho & Ferreira, 1995; Liao & Li, 2002; Liu & Wang, 2000; Serra & Bottura, 2006; Zambrano & Camacho, 2002), which in turn can require a significant computational time to be evaluated. In this context, deterministic techniques are difficult to apply to obtain the set of Pareto optimal solutions of many multiobjective optimization problems,

* Corresponding author. Tel.: +55 41 3271 13 33; fax: +55 41 3271 13 45.
E-mail addresses: hvha@ieee.org (H.V. Hultmann Ayala), leandro.coelho@pucpr.br (L. dos Santos Coelho).

0957-4174/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2012.02.027

so stochastic methods have been widely used and applied. Among them, the use of evolutionary algorithms and other nature-inspired algorithms for solving multiobjective optimization problems has significantly grown in the last years, giving raise to a wide variety of algorithms (Coello, 1999; Coello, Van Veldhuizen, & Lamont, 2002; Deb, 2001; Fonseca & Fleming, 1995; Osyczka, 1985; Van Veldhuizen & Lamont, 2000). Evolutionary Algorithms (EAs) are general-purpose stochastic search methods that use the metaphor of evolution as the key element in the design and implementation of computer-based problem solving systems. EAs offer a number of advantages: robust and reliable performance, global search capability, little or no information requirement, and others.

In recent years, in particular, genetic algorithms (GAs) have been investigated by many authors (see examples in Amjady & Nasri-Rad (2010), Bakirli, Birant, & Kut (2011), Coello (1999), Coello et al. (2002), Deb (2001), Kuroki, Young, & Haupt (2010), Lee & Ahn (2011), Mahmoudabadi & Tavakkoli-Moghaddam (2011), Prakash, Chan, & Deshmukh (2011)). GAs are based on the genetic process of biological organisms such as natural selection and reproduction. Furthermore, GAs are adaptive methods that may be used to solve search and optimization problems. GAs do not guarantee to obtain the optimal solution, but they provide appropriate solutions to a wide range of optimization problems which other deterministic methods find difficult. However, GA possesses advantages that it does not require any gradient information and inherent parallelism in searching the design space, thus making it a robust adaptive optimization technique.

GAs work with a population of individuals (potential solutions of optimization problem), each representing a possible solution to a given problem. Each individual is assigned a fitness score according to the qualification of that solution for the given problem. Individuals are selected from the population and recombined to produce offspring (new solutions) using the crossover and mutation mechanisms comprising the next generation. The GA is applied until that the evolutionary procedure meets a given evolution stopping criteria.

For multi-objective optimization methods, some modification to the simple GA is necessary. Multi-Objective Genetic Algorithm (MOGA) (Fonseca & Fleming, 1993), Vector Evaluated Genetic Algorithm (VEGA) (Schaffer, 1985), Niched Pareto Genetic Algorithm (NPGA) (Horn, Nafpliotis, & Goldberg, 1994) and Non-Dominated Sorting Genetic Algorithm (NSGA) (Srinivas & Deb, 1994) are examples of GA based multi-objective solution methods.

The NSGA proposed by Srinivas and Deb (1994) has been successfully applied to solving many problems, the main criticisms of this approach has been its high computational complexity of nondominated sorting, lack of elitism, and need for specifying a tunable parameter called sharing parameter. Recently, Deb, Pratap, Agarwal, and Meyarivan (2002) reported an improved version of NSGA, which they called NSGA-II, to address all the above issues. NSGA-II is based on Pareto solutions, measuring individual fitness according to their dominance property. The non-dominated individuals in the population are regarded as the fittest, and the dominated individuals are assigned lower fitness values.

The purpose of this work is to extend this methodology for solution of a multiobjective control problem under the framework of NSGA-II approach proposed in Ayala and Coelho (2008). The efficiency of the proposed method is illustrated by solving the tuning of a PID (Proportional-Integral-Derivative) multivariable controller applied to a robotic manipulator of two-degree-of-freedom. In the present work, two objective optimizations were carried out to obtain the PID's design parameters. Simulation results show that the NSGA-II algorithm can evolve good control profiles which result in an acceptable compromise between two (and possibly conflicting) objectives of tracking of position and velocity trajectories.

The remainder of this paper is organized as follows. In Section 2, the fundamentals of robotic manipulator are presented, while Section 3 explains the concepts of multiobjective optimization and the NSGA-II method. Section 4 presents the setup the NSGA-II approach and the simulation results. Lastly, Section 5 outlines the conclusion and future research.

2. Description of robotic manipulator of two-degree of freedom

Equations that characterize the robot dynamic are represented by a set of coupling differential equations, and, there are terms such as: varying inertia, centrifugal and Coriolis torques, load and gravity terms. The movement of the end-effector in a particular trajectory with constraint speed requires a complex set of torque functions to be applied to the actuators in the link of the robotic manipulator. Next, the description of the robot mathematical model is presented.

The manipulator model usually considers the representation of the robotic manipulator dynamic of n-links (in our case $n = 2$) governed by the following equation (Ayala & Coelho, 2008):

$$M(\theta)\ddot{\theta} + C(\theta,\dot{\theta}) + G(\theta) = \tau \tag{1}$$

where $M(\theta) \in \Re^{nxn}$ is the positive definite matrix of the system, $C(\theta,\dot{\theta}) \in \Re^{nx1}$ is the vector that represents the effects of centrifugal and Coriolis torques, $G(\theta) \in \Re^{nx1}$ is the vector of the gravitational torque effect, $\tau \in \Re^{nx1}$ is the vector of the torque of the links, and, $\theta, \dot{\theta}$, and $\ddot{\theta}$ are angular position, velocity and acceleration of the links. The dynamic model of robotic manipulator utilized for evaluation of the controllers is presented in Fig. 1.

The dynamic equations are given by Craig (1996):

$$\tau_1 = m_2 l_2^2(\ddot{\theta}_1 + \ddot{\theta}_2) + m_2 l_1 l_2 c_2(2\ddot{\theta}_1 + \ddot{\theta}_2) + (m_1 + m_2) l_1^2 \ddot{\theta}_1 - m_2 l_1 l_2 s_2 \dot{\theta}_2^2 - 2 m_2 l_1 l_2 s_2 \dot{\theta}_1 \dot{\theta}_2 + m_2 l_2 g c_{12} + (m_1 + m_2) l_1 g c_1 \tag{2}$$

$$\tau_2 = m_2 l_1 l_2 c_2 \ddot{\theta}_1 + m_2 l_1 l_2 s_2 \dot{\theta}_1^2 + m_2 l_1 g c_{12} + m_2 l_2^2(\ddot{\theta}_1 + \ddot{\theta}_2) \tag{3}$$

where $s_1 = \sin(\theta_1)$, $s_2 = \sin(\theta_2)$, $c_1 = \cos(\theta_1)$, $c_2 = \cos(\theta_2)$, and $c_{12} = \cos(\theta_1 + \theta_2)$ and the subscript 1 and 2 denote the parameters of the links 1 and 2, respectively. Parameters utilized in all simulations were: links lengths — $l_1 = 0.8$ m and $l_2 = 0.4$ m, links masses — $m_1 = m_2 = 0.1$ kg, and gravity acceleration $g = 9.81$ m/s^2 (Mital & Chin, 1995). The sampling period is $T_s = 10$ ms and the simulation period is 2 s ($N = 200$ samples). The imposed constraint in torque τ_1 and τ_2 are $[-1000; 1000]$ N m. Signals $\theta_{d,j}$ and $\dot{\theta}_{d,j}$ are desired values of the angular position and velocity of the robotic links, respectively. The position and velocity error vectors are respectively defined by

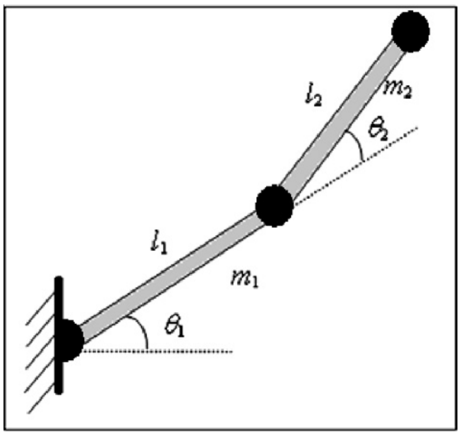

Fig. 1. Geometry of robotic manipulator of two-degree of freedom.

$$e_j(t) = \theta_j(t) - \theta_{d,j}(t), \quad j = 1,2 \tag{4}$$

$$v_j(t) = \dot{\theta}_j(t) - \dot{\theta}_{d,j}(t), \quad j = 1,2 \tag{5}$$

In this paper, the form of the multivariable PID control is calculated by

$$\tau_j(t) = K_{p,j}e_j(t) + K_{v,j}v_j(t) + K_{i,j}\int_o^t d(s)ds, \quad j = 1,2 \tag{6}$$

where $K_{p,j}$, $K_{v,j}$ and $K_{i,j}$ are the positive diagonal matrices relative to the position proportional, velocity (differential of position), and integral terms of error, respectively. The discretization of Eq. (6), provides a discrete equation of the controller given by

$$\Delta\tau_j(t) = (K_{p,j} + 0.5T_sK_{i,j})e_j(t) - (K_{p,j} - 0.5T_sK_{i,j})e_j(t-1) + K_{v,j}v_j(t) + K_{v,j}v_j(t-1), \quad j = 1,2 \tag{7}$$

$$v_j = [\theta_{d,j}(t) - \theta_{d,j}(t-1) - \theta_j(t) + \theta_j(t-1)]/T_s, \quad j = 1,2 \tag{8}$$

3. Fundamentals of multiobjective optimization, GAs and NSGA-II

The presence of multiple objectives in an optimization problem, in principle, gives rise to a set of optimal solutions (known as Pareto-optimal solutions), instead of a single optimal solution. In the absence of any further information, one of these Pareto-optimal solutions cannot be said to be better than the other. This demands a user to find as many Pareto-optimal solutions as possible (Deb, 2001).

The Pareto optimal solutions to a multiobjective optimization problem often distribute very regularly in both the decision space and the objective space. A problem that arises however is how to normalize, prioritize and weight the contributions of the various objectives in arriving at a suitable measure. Also these objectives can interact or conflict with each other, increasing one can reduce others in turn and this can happen in nonlinear ways. Most of the classical Operational Research methods of obtaining solutions or approaching the Pareto front (including the multicriterion decision-making methods) focus on the first stage of ranking the objectives, i.e. trying to reduce the design space to a more easily managed mathematical form (since most such problems are far too complex to enumerate and evaluate all the possible combinations in any reasonable time) (Khare, 2002).

A single objective optimization algorithm will normally be terminated upon obtaining an optimal solution. However, for most realistic the multi-objective problems, there could be a number of optimal solutions. Suitability of one solution depends on a number of factors including user's choice and problem environment, and hence finding the entire set of optimal solutions may be desired. The next section presents the fundamentals of MO.

3.1. Fundamentals of MO

Mathematically, a general multiobjective optimization problem contains a number of objectives to be minimized and (optional) constraints to be satisfied. In this case, a multiobjective optimization problem consists of optimizing a vector of functions:

$$\begin{aligned} &Opt(F(x)) = (f_1(x), f_2(x), \ldots, f_k(x)) \\ &\text{subject to}: \quad g_i(x) \leqslant 0, \; i = 1,2,\ldots,q, \\ &\qquad\qquad\quad h_j(x) = 0, \; j = 1,2,\ldots,r, (q + r = m), \end{aligned}$$

where $x = (x_1, x_2, \ldots, x_n)^T \in \boldsymbol{X}$ is the solution vector or decision variables, $\boldsymbol{X}$ is the set of feasible solutions, $F(x)$ is a vector of objectives, $f_i : \mathrm{IR}^n \to \mathrm{IR}$, $i = 1,2,\ldots,k$ are the objective functions and g_j, $h_j : \mathrm{IR}^n \to \mathrm{IR}$, $i = 1,\ldots,q$, $j = 1,\ldots,r$ are the constraint functions of the problem.

These functions $f_1(x), f_2(x), \ldots, f_k(x)$, usually in conflict with each other, are a mathematical description of performance criteria. The meaning of optimum is not well defined in this context, so it is difficult to have a vector of decision variables that optimizes all the objectives simultaneously. Therefore, the concept of *Pareto optimality* is used.

The concept of optimality in single objective is not directly applicable in multiobjective optimization problems. For this reason a classification of the solutions is introduced in terms of Pareto optimality, according to the following definitions (Zitzler, Laumanns, & Bleuler, 2002). In terms of minimization of objective functions:

Definition 1. *Pareto optimal*: A solution vector $x^* \in X$ is Pareto optimal solution iff $\neg\exists x \in X$: $f_i(x) \leqslant f_i(x^*) \wedge f(x) \neq f(x^*)$; $\forall i = \{1,2,\ldots,k\}$. These solutions are also called true Pareto solutions.

Definition 2. *Pareto dominance*: A solution vector x^1 is said dominate another feasible solution x^2 (denote this relationship $x^1 \succ x^2$) iff $f_i(x^1) \leqslant f_i(x^2) \wedge \exists j$: $f_j(x^1) \leqslant f_j(x^2)$; $\forall i,j = \{1,2,\ldots,k\}$. If there are no solutions which dominate x^1, then x^1 is non-dominated.

Definition 3. *Pareto set*: A set of non-dominated feasible solutions $\{x^* | \neg\exists\, x: x \succ x^*\}$ is said to be a Pareto set.

Definition 4. *Pareto front*: The set of vectors in the objective space that are image of a Pareto set, i.e. $\{F(x^*) | \neg\exists x: x \succ x^*\}$. A representation of the Pareto front for a bi-objective space is presented in Fig. 2.

In many cases, one cannot expect to be able to determine the set of all Pareto-efficient solutions *exactly*, at least not for problem instances of a realistic magnitude. Accordingly, one needs to establish approximations of a sufficiently high quality for this set.

3.2. Genetic algorithms

Evolutionary computation field includes various methods, which are also called evolutionary computation methods. These include: genetic algorithms, evolution strategies, evolutionary programming, differential evolution and genetic programming. The GA is a paradigm which is based on both gene recombination and the Darwinian "survival of the fittest" theory (Goldberg, 1989). A group of individuals in some environment have a higher probability to reproduce if their fitness (their ability to thrive in this environment) is high. Offspring are created via a crossover

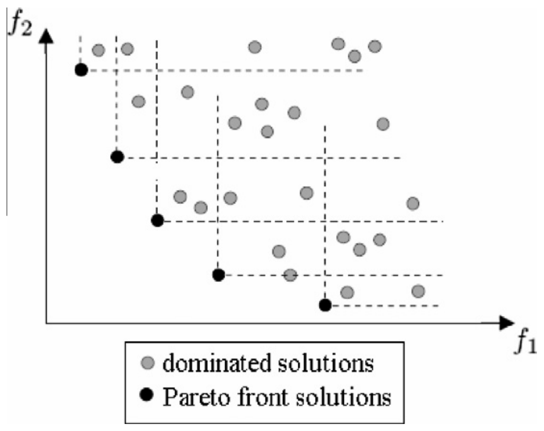

Fig. 2. The Pareto front of a set of solutions in a bi-objective space.

```
Determination of control parameters of GA
Begin
  Initialize randomly a population of individuals (potential solutions) using uniform distribution
  Evaluate the fitness of each individual
  Generation counter, $k = 1$;
  While (not stopping-condition) do
    Implement GA operators (selection, crossover, and mutation) for generate new individuals
    Evaluate the fitness of each new individual
    Replace the old individuals with the new individuals
  End While
End Begin
```

Fig. 3. Pseudocode of a basic GA.

operation (as in gene recombination) and mutation. There are three principal operations in a GA (Podlena & Hendtlass, 1998):

(i) The *crossover* operation generates offspring from two chosen individuals in the population, by exchanging some bits in the two individuals. The offspring thus inherit some characteristics from each parent;
(ii) The *mutation* operation generates offspring by randomly changing one or several bits in an individual. Offspring may thus possess different characteristics from their parents. The mutation operator may be considered to be an important element for solving the premature convergence problem, since it serves to create random diversity in the population;
(iii) The *selection* operation chooses some offspring for survival according to predefined rules. This keeps the population size within a fixed constant and puts good offspring into the next generation with a high probability.

The pseudocode of a GA is presented in Fig. 3.

3.3. Multiobjective optimization using evolutionary algorithms

Evolutionary algorithms are becoming increasingly valuable in solving realistic engineering problems. Most of these problems deal with sufficiently complex issues that typically conflict with each other, thus requiring multiobjective analysis to assist in identifying compromise solutions.

In a typical multiobjective optimization problem, there exists a family of equivalent solutions that are superior to the rest of the solutions and are considered equal from the perspective of simultaneous optimization of multiple (and possibly competing) objective functions. Such solutions are called noninferior, nondominated or Pareto-optimal solutions, and are such that no objective can be improved without degrading at least one of the others, and, given the constraints of the model, no solution exist beyond the true Pareto front. The goal of multiobjective algorithms is to locate the (whole) Pareto front.

Multiple-Objective Evolutionary Algorithms (MOEAs) is the term employed in the evolutionary multicriteria optimization field to refer to as a group of EA formulated to deal with multiobjective optimization problem. The goal of MOEAs actually consists of two parts as mentioned in Coello et al. (2002) and Deb (2001), namely that the solutions found must be: (i) close to the Pareto optimal front, and (ii) diverse.

This is also illustrated in Fig. 4, where it is clear how solutions near the Pareto optimal front first are sought followed by a search for diversity along the front. The first requirement can be obtained using the conventional concept of dominance and does not have a need for any niching or crowding measures. A good algorithm would thus be able to find a set of solutions as close to the Pareto optimal front as possible. However, the second requirement can be more difficult to obtain. In order to obtain a diverse set, it must be specified what can be considered as a set of diverse solutions, but it

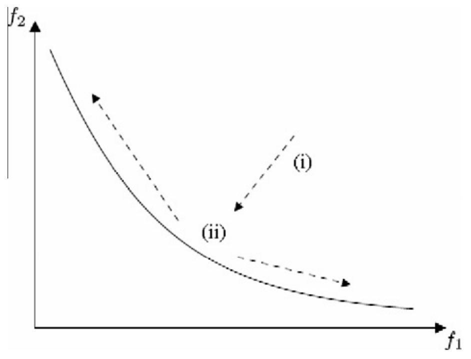

Fig. 4. The goal for MOEAs is to find a diverse set of solutions near the Pareto optimal front by first founding the front and then creating diversity along it.

must also be understood how dominance has influenced the diversity of the solutions.

3.4. NSGA-II approach

NSGA-II differs from a simple genetic algorithm only in the way the selection operator works. The efficiency of NSGA-II lies in the way multiple objectives are reduced to a single fitness measure by the creation of number of fronts, sorted according to nondomination.

NSGA-II is a computationally efficient algorithm implementing the idea of a selection method based on classes of dominance of all the solutions. It incorporates an elitist and a rule for adaptation assignment that takes into account both the rank and the distance of each solution regarding others (sharing mechanism for solution diversification).

Fig. 5(a) shows what is meant by rank in a minimization case. The value of adaptation is equal to its rank. When comparing two solutions belonging to the same rank, extreme solutions prevail over not extreme ones. If both solutions are not extreme, the one with the bigger crowding distance (i.e. the perimeter of the cuboid calculated between the two nearest neighbors) wins (Fig. 5(b)). This way extreme solutions and less crowded areas are encouraged (Salazar, Rocco, & Galván, 2006).

A NSGA-II for finding a well-distributed and well-converged set of Pareto-optimal solutions is presented in this work. The NSGA-II algorithm used in this work may be stated as (Ayala & Coelho, 2008):

(i) Generation $t = 0$.
(ii) Generate a uniformly distributed parent population of size P.
(iii) Evaluate the individuals and sort the population based on the nondomination.
(iv) Assign each solution a rank equal to its nondomination level (minimization of fitness is assumed).
(v) Use the usual binary tournament selection.
(vi) Use the Simulated Binary Crossover operator and polynomial mutation (Deb & Agarwal, 1995) to create an offspring population of size P.
(vii) Combine the offspring and parent population to form extended population of size $2P$.
(viii) Sort the extended population based on non-domination.
(ix) Fill new population of size P with the individuals from the sorting fronts starting from the best.
(x) Invoke the crowding-distance method to ensure diversity if a front can only partially fill the next generation. The crowding-distance method maintains diversity in the population and prevents convergence in one direction.
(xi) Update the number of generations, $t = t + 1$.
(xii) Repeat the steps (iii) to (xi) until a stopping criterion is met.

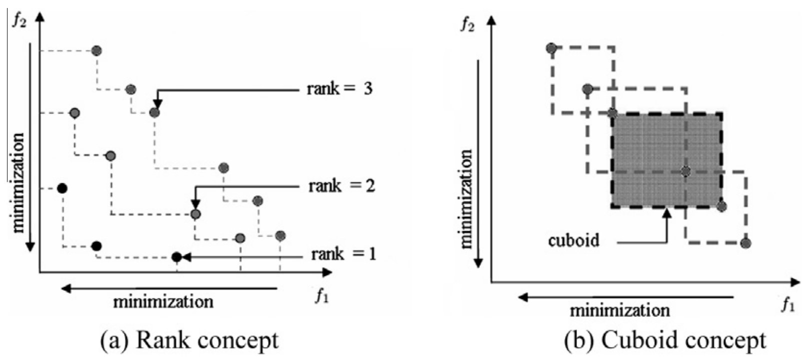

Fig. 5. Concepts used by NSGA-II (Salazar et al., 2006).

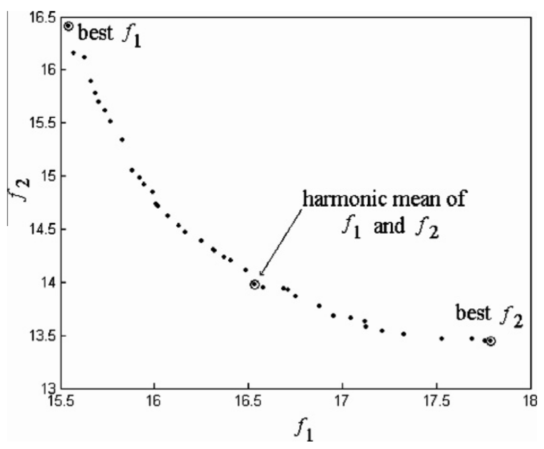

Fig. 6. The Pareto front obtained by NSGA-II.

4. Simulation results

The objective of PID controllers optimization is to minimize the effect of the position and velocity in the links. In realized simulations the robot dynamic is solved by 4th order Runge–Kutta method. In this case, the NSGA-II deals with the controllers tuning in order to satisfy a cubic interpolation polynomial joint that represents the trajectory to be followed by the manipulator (Craig, 1996), and given by

$$\theta_{d,j}(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3, \quad j = 1,2 \tag{9}$$

where the constraints are

$$\dot{\theta}_{df,j}(t) = a_1 + 2a_2 t_f + 3a_3 t_f^2, \quad j = 1,2 \tag{10}$$

$$\ddot{\theta}_{df,j}(t) = 2a_2 + 6a_3 t_f, \quad j = 1,2 \tag{11}$$

where $\theta_{d,j}$ is the instantaneous desired position, θ_0 and $\dot{\theta}_0$ are the adopted values for position, velocity and acceleration in initial time, respectively. In this context, $\theta_{df,j}$ and $\dot{\theta}_{df,j}$ are the final desired values for the position ($\theta_{df,1} = 1$ rad and $\theta_{df,2} = 2$ rad in $t = 2$ s and $\theta_{df,1} = 0.5$ rad and $\theta_{df,2} = 4$ rad in final time, $t_f = 4$ s) and velocity ($\dot{\theta}_{df,1} = \dot{\theta}_{df,2} = 0$ rad/s in $t = 2$ s and $t_f = 4$ s), respectively.

The search range of PID parameters is [0; 350] for the gains $K_{p,1}$ and $K_{p,2}$, and [0; 50] for gains $K_{v,1}$, $K_{v,2}$, $K_{i,1}$ and $K_{i,2}$. The current work adopts a population size of NSGA-II with 40 individuals (binary representation of genetic algorithm) and evolutionary cycle has stopping criterion stipulated in 200 generations.

Table 1
Simulation results using NSGA-II.

Results	$K_{p,1}$	$K_{i,1}$	$K_{v,1}$	$K_{p,2}$	$K_{i,2}$	$K_{v,2}$	f_1	f_2
Best f_1	184.86	49.26	9.10	11.63	16.20	0.11	15.54	16.42
Best f_2	184.72	49.41	8.85	11.29	16.21	0.27	17.79	13.44
Best harmonic mean	184.76	49.68	8.94	11.46	16.54	0.20	16.54	13.97

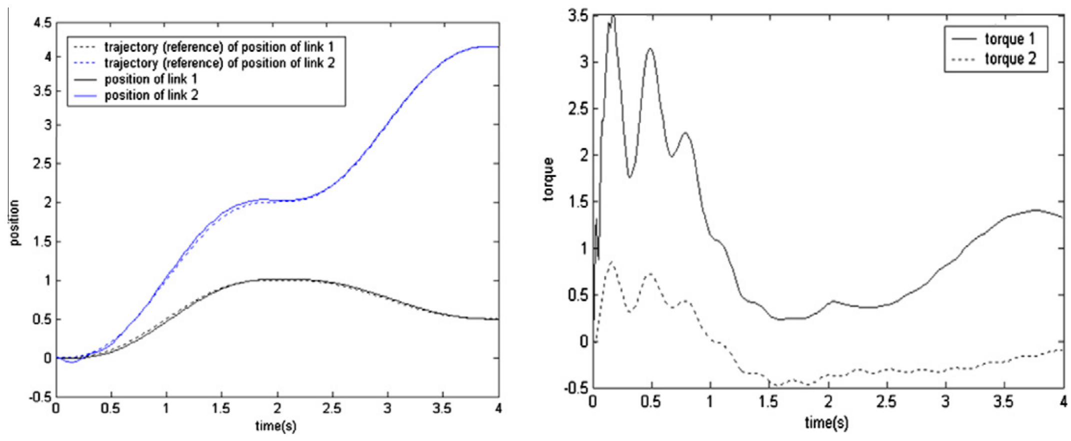

Fig. 7. Response of robotic manipulator in closed loop with PID control (best f_1).

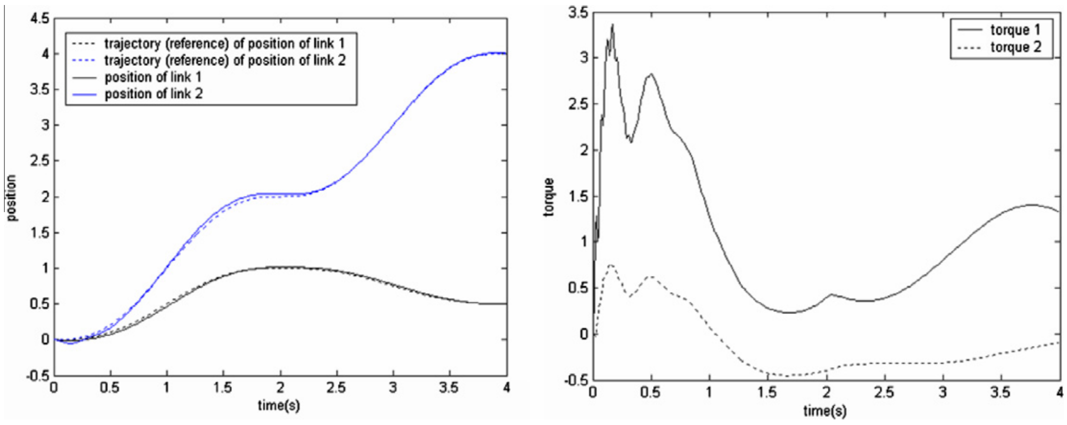

Fig. 8. Response of robotic manipulator in closed loop with PID control (best f_2).

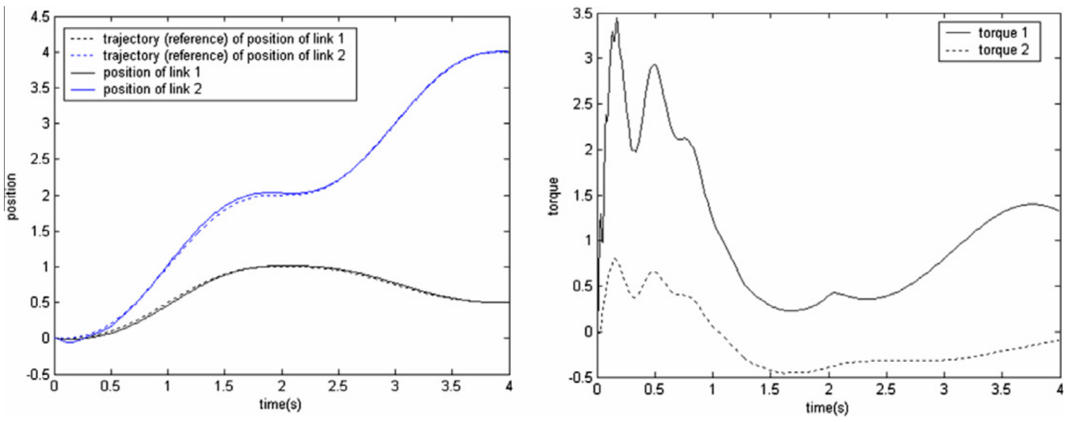

Fig. 9. Response of robotic manipulator in closed loop with PID control (harmonic mean of f_1 and f_2).

The aim of the NSGA-II in the PID controller tuning is the minimization of two f objectives: (i) minimization of positions errors of links 1 and 2, and (ii) minimization of variation of two control signals (torques). In this case, the objective functions f for minimization are given by:

$$f_1 = \sum_{t=1}^{tf} |\theta_1(t) - \theta_{d,1}(t)| + \sum_{t=1}^{tf} |\theta_2(t) - \theta_{d,2}(t)| \tag{12}$$

$$f_2 = \sum_{t=1}^{tf} |\tau_1(t) - \tau_1(t-1)| + \sum_{t=1}^{tf} |\tau_2(t) - \tau_2(t-1)| \tag{13}$$

Fig. 6 shows the Pareto front of simulation result using NSGA-II, and the results indicate that our approach is a viable alternative. The NSGA-II algorithm was able to find the Pareto front with good distribution of the solutions, the valuable characteristics in multi-objective optimization as shown in Fig. 6.

It can be seen that the design objective have been satisfied as shown in Fig. 6. However, from such responses it is difficult to determine the relative merits of one controller against another over the entire population.

The results of best f_1, best f_2, and best harmonic mean of objectives f_1 and f_2 of PIDs tuning using NSGA-II are presented in Table 1 and Figs. 7–9, respectively.

5. Conclusion and future research

The controllers used to control processes can have several structures. The choice of the structure determines how well the plant can be controlled. The plant poses some guidelines and restrictions on the controller. The choice of controller determines the best possible performance achieved of a well-tuned controller.

PID controllers are used extensively in the industry as an all-in-all controller, mostly because it is an intuitive control algorithm (Aström & Hagglund, 1995). The popularity and widespread use of PID controllers are attributed primarily to their simplicity and performance characteristics. The PID controller has three parts: the proportional part is proportional to the error, the integral part removes the steady-state error and the derivative part reduces the overshoot. The weights of the controller's actions are adjusted with the P, I and D gains.

In the optimization tuning method the tuning of a parameterized controller such as a multivariable PID controller is based on an optimization criterion. Many optimization tuning problems involve simultaneous optimization of multiple performance measures that are often noncommensurable and competing with each other. Historically, optimization techniques have dealt with multiple objectives by combining them into one objective function composed of the weighted sum of individual objectives, or by transforming one objective into a single response function while using others as constraints.

This paper presented the tuning of two PID controllers through NSGA-II. NSGA-II is a computationally efficient algorithm implementing the idea of a selection method based on classes of dominance of all the solutions. It uses a fast non-dominated ranking algorithm and a parameter-less sharing mechanism for solutions diversification.

Simulation numerical results of multivariable PID control and convergence of the NSGA-II were discussed with application in a robotic manipulator of two-degree-of-freedom. The proposed optimization method based on NSGA-II offers an effective way to implement simple but robust solutions providing a good reference tracking performance in closed loop as shown in Figs. 7–9.

Perspectives for improving the input–output behavior and robustness of PID design should addressed combining other paradigms of computational intelligence for treatment of the coupling between variables in multivariable PID design. Furthermore, comparative studies with other evolutionary optimization approaches (Coelho & Pessôa, 2011; Iruthayarajan & Baskar, 2009; Iruthayarajan & Baskar, 2010) for multivariable processes must be approached.

Acknowledgements

This work was supported by the National Council of Scientific and Technologic Development of Brazil — CNPq — under Grants No. 303963/2009-3/PQ and 'Fundação Araucária' under Grant No. 14/2008–416/09-15149 for the financial support of this work.

References

Aggelogiannaki, E., Sarimveis, H., & Bafas, G. (2004). Multiobjective constrained mpc with simultaneous closed loop identification for MIMO processes. *Computer Aided Chemical Engineering, 18*, 523–528.

Amjady, N., & Nasri-Rad, H. (2010). Solution of nonconvex and nonsmooth economic dispatch by a new adaptive real coded genetic algorithm. *Expert Systems with Applications, 37*(7), 5239–5245.

Aström, K. J., & Hagglund, T. (1995). PID controllers: theory, design, and tuning. Research Triangle Park, NC, USA, Instrument Society of America.

Ayala, H. V. H., & Coelho, L. S. (2008). A multiobjective genetic algorithm applied to multivariable control optimization. In *ABCM symposium series in mechatronics* (Vol. 3, section VII.03, pp. 636–645).

Bakirli, G., Birant, D., & Kut, A. (2011). An incremental genetic algorithm for classification and sensitivity analysis of its parameters. *Expert Systems with Applications, 38*(3), 2609–2620.

Carvalho, J. R. H., & Ferreira, P. A. V. (1995). Multiple-criterion control: A convex programming approach. *Automatica, 31*(7), 1025–1029.

Coelho, L. S., & Pessôa, M. W. (2011). A tuning strategy for multivariable PI and PID controllers using differential evolution combined with chaotic Zaslavskii map. *Expert Systems with Applications, 38*(11), 13694–13701.

Coello, C. A. C. (1999). A comprehensive survey of evolutionary-based multiobjective optimization. *Knowledge and Information Systems, 1*(3), 269–308.

Coello, C. A. C., Van Veldhuizen, D. A., & Lamont, G. B. (2002). *Evolutionary algorithms for solving multi-objective problems*. New York, NY, USA: Kluwer Academic Publishers..

Craig, J. J. (1996). *Introduction to robotics: Mechanics & control*. Addison-Wesley (Chapter 6).

Deb, K. (2001). *Multi-objective optimization using evolutionary algorithms. Wiley-Interscience Series in Systems and Optimization*. John Wiley & Sons.

Deb, K., & Agarwal, R. B. (1995). Simulated binary crossover for continuous search space. *Complex Systems, 9*(2), 115–148.

Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A fast and elitist multi-objective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation, 6*(2), 182–187.

Fonseca, C., & Fleming, P. J. (1993). Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization. In *Proceedings of the 5th international conference on genetic algorithms* (pp. 416–423).

Fonseca, C. M., & Fleming, P. J. (1995). An overview of evolutionary algorithms in multiobjective optimization. *Evolutionary Computation, 3*(1), 1–16.

Goldberg, D. E. (1989). *Genetic algorithms in search, optimization, and machine learning*. USA: Addison-Wesley.

Horn, J., Nafpliotis, N., & Goldberg, D. E. (1994). A niched pareto genetic algorithm for multi-objective optimization. In *Proceedings of the 1st IEEE conference on evolutionary computation, IEEE world congress on computational intelligence*, Orlando, FL, USA (Vol. 1, pp. 82–87).

Iruthayarajan, M. W., & Baskar, S. (2009). Evolutionary algorithms based design of multivariable PID controller. *Expert Systems with Applications, 36*(5), 9159–9167.

Iruthayarajan, M. W., & Baskar, S. (2010). Covariance matrix adaptation evolution strategy based design of centralized PID controller. *Expert Systems with Applications, 37*(8), 5775–5781.

Khare, V. (2002). Performance scaling of multi-objective evolutionary algorithms. Master thesis. School of Computer Science, The University of Birmingham, Birmingham, UK.

Kuroki, Y., Young, G. S., & Haupt, S. E. (2010). UAV navigation by an expert system for contaminant mapping with a genetic algorithm. *Expert Systems with Applications, 37*(6), 4687–4697.

Lee, S., & Ahn, H. (2011). The hybrid model of neural networks and genetic algorithms for the design of controls for internet-based systems for business-to-consumer electronic commerce. *Expert Systems with Applications, 38*(4), 4326–4338.

Liao, L. Z., & Li, D. (2002). Adaptive differential dynamic programming for multiobjective optimal control. *Automatica, 38*(6), 1003–1015.

Liu, W., & Wang, G. (2000). Auto-tuning procedure for model-based predictive controller. In *Proceedings of IEEE international conference on systems, man, and cybernetics*, Nashville, Tennessee, USA (Vol. 5, pp. 3421–3426).

Mahmoudabadi, A., & Tavakkoli-Moghaddam, R. (2011). The use of a genetic algorithm for clustering the weighing station performance in transportation – A case study. *Expert Systems with Applications, 38*(9), 11744–11750.

Martínez, M., García-Nieto, S., Sanchis, J., & Blasco, X. (2009). Genetic algorithms optimization for normalized normal constraint method under Pareto construction. *Advances in Engineering Software, 40*(4), 260–267.

Mital, D. P., & Chin, L. (1995). Intelligent control applications with neural networks. In M. M. Gupta & N. K. e Sinha (Eds.), *Intelligent control systems: Theory and applications* (pp. 479–514). USA: IEEE Press. Chapter 18, Piscataway.

Osyczka, A. (1985). *Multicriteria optimization for engineering design*. Academic Press.

Podlena, J. R., & Hendtlass, T. (1998). An accelerated genetic algorithm. *Applied Intelligence, 8*(2), 103–111.

Prakash, A., Chan, F. T. S., & Deshmukh, S. G. (2011). FMS scheduling with knowledge based genetic algorithm approach. *Expert Systems with Applications, 38*(4), 3161–3171.

Salazar, D., Rocco, C. M., & Galván, B. J. (2006). Optimization of constrained multiple-objective reliability problems using evolutionary algorithms. *Reliability Engineering and System Safety, 91*(9), 1057–1070.

Schaffer, J. D. (1985). Multiple objective optimization with vector evaluated genetic algorithms. In *Proceedings of the first international conference of genetic algorithms*, Pittsburgh, PA, USA (pp. 93–100).

Serra, G. L. O., & Bottura, C. P. (2006). Multiobjective evolution based fuzzy PI controller design for nonlinear systems. *Engineering Applications of Artificial Intelligence, 19*(2), 157–167.

Srinivas, N., & Deb, K. (1994). Multi-objective function optimization using nondominated sorting genetic algorithms. *Evolutionary Computation, 2*(3), 221–248.

Van Veldhuizen, D. A., & Lamont, G. B. (2000). Multiobjective evolutionary algorithms: Analyzing the state-of-the-art. *Evolutionary Computation, 8*(2), 125–147.

Zambrano, D., & Camacho, E. F. (2002). Application of MPC with multiple objective for a solar refrigeration plant. In *Proceedings of the IEEE international conference on control applications*, Glasgow, Scotland, UK (pp. 1230–1235).

Zitzler, E., Laumanns, M., & Bleuler, S. (2002). A tutorial on evolutionary multiobjective optimization. In *Workshop on multiple objective MetaHeuristics (MOMH 2002)*, Paris, France.