A MULTIOBJECTIVE GENETIC ALGORITHM APPLIED TO MULTIVARIABLE CONTROL OPTIMIZATION

Helon Vicente Hultmann Ayala
Undergraduate Program at Mechatronics Engineering
Pontifical Catholic University of Paraná
Rua Imaculada Conceição, 1155
80215-901, Curitiba, Paraná, Brazil
helon.ayala@pucpr.br

Leandro dos Santos Coelho
Industrial and Systems Engineering Graduate Program, PPGEPS
Pontifical Catholic University of Paraná
Rua Imaculada Conceição, 1155
80215-901, Curitiba, Paraná, Brazil
leandro.coelho@pucpr.br

Abstract*. Many control problems involve simultaneous optimization of multiple performance measures that are often noncommensurable and competing with each other. Traditionally, classical optimization algorithms based on nonlinear programming or optimal control theory is applied to obtain the solution of such problems using different scalar approaches. The presence of multiple objectives in a problem usually gives rise to a set of optimal solutions, largely known as Pareto-optimal solutions. In this 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 NSGA-II. 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.*

Keywords*: multiobjective optimization, robotic manipulator, genetic algorithm, PID control.*

1. INTRODUCTION

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 *et al*., 2000).

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).

Most optimization problems in control systems (Carvalho and Ferreira, 1995; Li and Wang, 2000; Liao and Li, 2002; Zambrano and Camacho, 2002) involve the optimization of more than one objective function, 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, 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 (Osyczka, 1985; Fonseca and Fleming, 1995; Coello, 1999; Van Veldhuizen and Lamont, 2000; Coello *et al*., 2002; Deb, 2001).

In recent years, in particular, genetic algorithms (GAs) have been investigated by many authors (Coello *et al*., 2002; Deb, 2001). The GA is based on the concepts of natural selection and reproduction. 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.

For multi-objective optimization methods, some modification to simple GA is necessary. Multi-Objective Genetic Algorithm (MOGA) (Fonseca and Fleming, 1993), Vector Evaluated Genetic Algorithm (VEGA) (Schaffer, 1985), Niched Pareto Genetic Algorithm (NPGA) (Horn *et al.*, 1994) and Non-Dominated Sorting Genetic Algorithm (NSGA) (Srinivas and 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 *et al.* (2002) reported an improved version of NSGA, which they called NSGA-II, to address all the above issues.

The purpose of this work is to extend this methodology for solution of a multiobjective control problem under the framework of NSGA-II approach. 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 proposed 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 of NSGA-II and the simulation results. Lastly, section 5 outlines the conclusion and future research.

2. DESCRIPTION OF ROBOTIC MANIPULATOR OF TWO-DEGREE OF FREEDOM (2-DOF)

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 torque, 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:

$$M(q)\ddot{q} + C(q,\dot{q}) + G(q) = t \qquad (1)$$

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

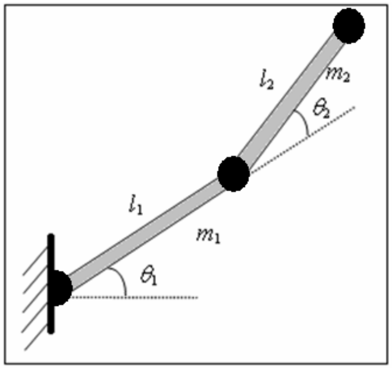


Figure 1. Geometry of robotic manipulator of 2-DOF.

The dynamic equations are given by (Craig, 1996):

$$t_1 = m_2 l_2^2(\ddot{q}_1 + \ddot{q}_2) + m_2 l_1 l_2 c_2(2\ddot{q}_1 + \ddot{q}_2) + (m_1 + m_2)l_1^2\ddot{q}_1 - \mathrm{m}_2 l_1 l_2 s_2 \dot{q}_2^2 - 2m_2 l_1 l_2 s_2 \dot{q}_1 \dot{q}_2 + m_2 l_2 g c_{12} + (m_1 + m_2)l_1 g c_1 \qquad (2)$$

$$t_2 = m_2 l_1 l_2 c_2 \ddot{q}_1 + m_2 l_1 l_2 s_2 \dot{q}_1^2 + m_2 l_1 g c_{12} + m_2 l_2^2(\ddot{q}_1 + \ddot{q}_2) \qquad (3)$$

where s_1=sin(q_1), s_2=sin(q_2), c_1=cos (q_1), c_2=cos(q_2), e c_{12}=cos(q_1+q_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 and Chin, 1995). The sampling period is T_s=10 msec and the simulation period is 2 sec (N=200 samples). The imposed constraint in torque t_1 and t_2 are [-1000; 1000] Nm. Signals $q_{\mathrm{d,j}}$ and $\dot{q}_{\mathrm{d,j}}$ are desired values of the angular position and velocity of the robotic links, respectively. The position and velocity error vectors are defined by

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

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

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

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

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

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

$$v_j = [q_{d,j}(t) - q_{d,j}(t-1) - q_j(t) + q_j(t-1)]/T_s, \; j=1,2 \tag{8}$$

3. FUNDAMENTALS OF MULTIOBJECTIVE OPTIMIZATION AND NSGA-II APPROACH

3.1. Genetic Algorithms

Evolutionary algorithms include 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 operation (as in gene recombination) and mutation (Podlena and Hendtlass, 1998). There are three principal operations in a GA (Hong *et al*., 2002):

(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 Figure 2.

```
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
```

Figure 2. Pseudocode of a basic GA.

3.2. Multiobjective Optimization

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.

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:

$$Opt\ (F(x) = (f_1(x), f_2(x), \ldots, f_k(x))$$

$$\text{subject to: } g_i(x) \leq 0,\ i = 1,2,\ldots,q,$$
$$h_j(x) = 0,\ j = 1,2,\ldots,r,\ \ (q + r = m),$$

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, $f_i : \mathrm{IR}^n \rightarrow \mathrm{IR}$, $i = 1, 2,.., k$ are the objective functions and $g_j;$, $h_j : \mathrm{IR}^n \rightarrow \mathrm{IR}$, $i = 1,\ldots,q$, $j = 1,\ldots, r$ 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 *et al.*, 2002). In terms of minimization of objective functions:

Definition 1. *Pareto optimal*: A solution vector $x^* \in \boldsymbol{X}$ is Pareto optimal solution iff
$\neg\exists\ x \in \boldsymbol{X}$: $f_i(x) \leq f_i(x^*) \wedge f(x) \neq f(x^*)$; $\forall i = \{1,2,\ldots, k\}$. 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
$f_i(x^1) \leq f_i(x^2) \wedge \exists j$: $f_j(x^1) \leq f_j(x^2)$; $\forall i,j = \{1,2,\ldots, k\}$. 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 Figure 3.

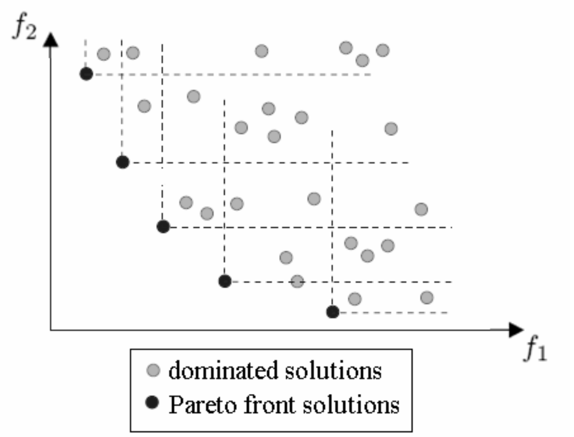


Figure 3. The Pareto front of a set of solutions in a bi-objective space.

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 multicriterion 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 Deb (2001) and Coello *et al*. (2002), namely that the solutions found must be: (i) close to the Pareto optimal front, and (ii) diverse.

This is also illustrated in Figure 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 must also be understood how dominance has influenced the diversity of the solutions.

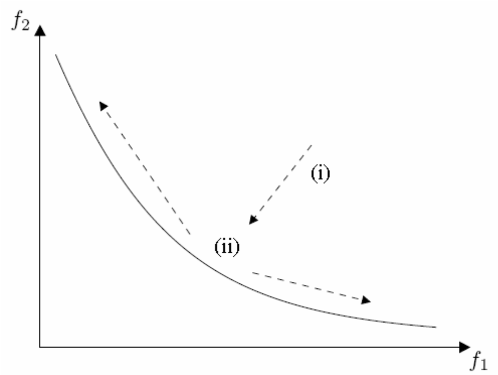


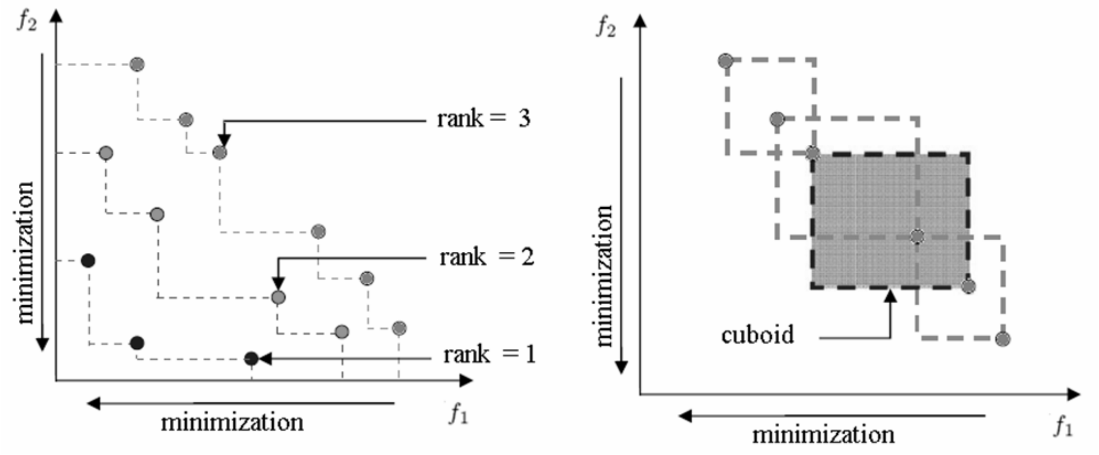
Figure 4. The goal for MOEAs is to find a diverse set of solutions near the Pareto optimal front by first founding the front and then creating diversity along it.

3.3. 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).

Figure 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 (Figure 5(b)). This way extreme solutions and less crowded areas are encouraged (Salazar *et al.*, 2006).



(a) Rank concept (b) Cuboid concept

Figure 5. Concepts used by NSGA-II (Salazar *et al.*, 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:

(1) Generation $t = 0$;
(2) Generate a uniformly distributed parent population of size P;
(3) Evaluate the individuals and sort the population based on the nondomination;
(4) Assign each solution a rank equal to its nondomination level (minimization of fitness is assumed);
(5) Use the usual binary tournament selection;
(6) Use the Simulated Binary Crossover operator and polynomial mutation (Deb and Agarwal, 1995) to create an offspring population of size P;
(7) Combine the offspring and parent population to form extended population of size $2P$;
(8) Sort the extended population based on non-domination;
(9) Fill new population of size P with the individuals from the sorting fronts starting from the best;
(10) 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;
(11) Update the number of generations, $t = t + 1$;
(12) Repeat the steps (3) to (11) until a stopping criterion is met.

4. SIMULATION RESULTS

The objective of PIDs 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

$$q_{d,j}(t) = a_0 + a_1t + a_2t^2 + a_3t^3\,, \qquad j=1,2 \tag{9}$$

where the constraints are

$$\dot{q}_{df,j}\,(t) = a_1 + 2a_2t_f + 3a_3t^{\,2}_{\,f}\,, \qquad j=1,2 \tag{10}$$

$$\ddot{q}_{df,j}\,(t) = 2a_2 + 6a_3t_f\,, \qquad j=1,2 \tag{11}$$

where $q_{d,j}$ is the instantaneous desired position, q_0 and $\dot{q}_0$ are the adopted values for position, velocity and acceleration in initial time, respectively. In this context, $q_{df,j}$ and $\dot{q}_{df,j}$ are the final desired values for the position ($q_{df,1}$=1 rad and $q_{df,2}$=2 rad in t=2 s and $\square_{df,1}$=0.5 rad and $q_{df,2}$=4 rad in final time, t_f=4 s) and velocity ($\dot{q}_{\text{df},1}=\dot{q}_{\text{df},2}$=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 population of NSGA-II is 40 individuals (binary representation of genetic algorithm) and evolutionary cycle has stopping criterion stipulated in 200 generations.

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:

$$f_1 = \sum_{t=1}^{tf} \left| q_1(t) - q_{d,1}(t) \right| + \sum_{t=1}^{tf} \left| q_2(t) - q_{d,2}(t) \right| \tag{12}$$

$$f_2 = \sum_{t=1}^{tf} \left| t_1(t) - t_1(t-1) \right| + \sum_{t=1}^{tf} \left| t_2(t) - t_2(t-1) \right| \tag{13}$$

Figure 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 multiobjective optimization as shown in Figure 6.

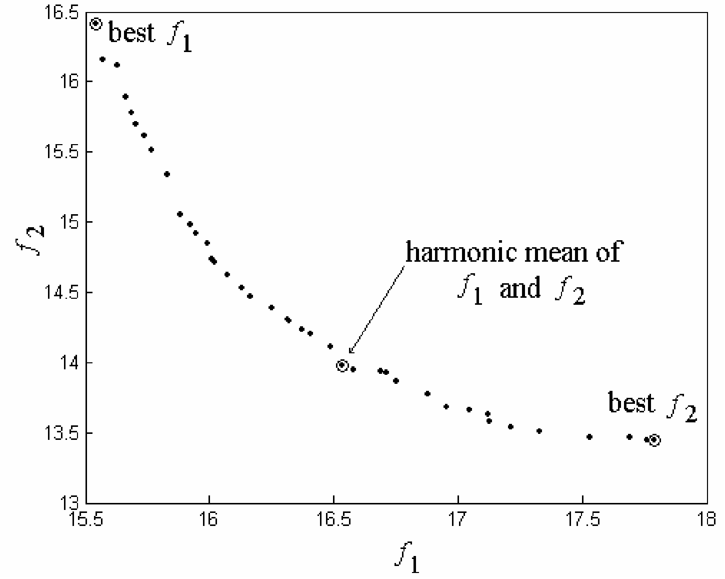


Figure 6. The Pareto front obtained by NSGA-II.

It can be seen that the design objective have been satisfied as shown in Figure 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 Figures 7, 8 and 9, respectively.

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

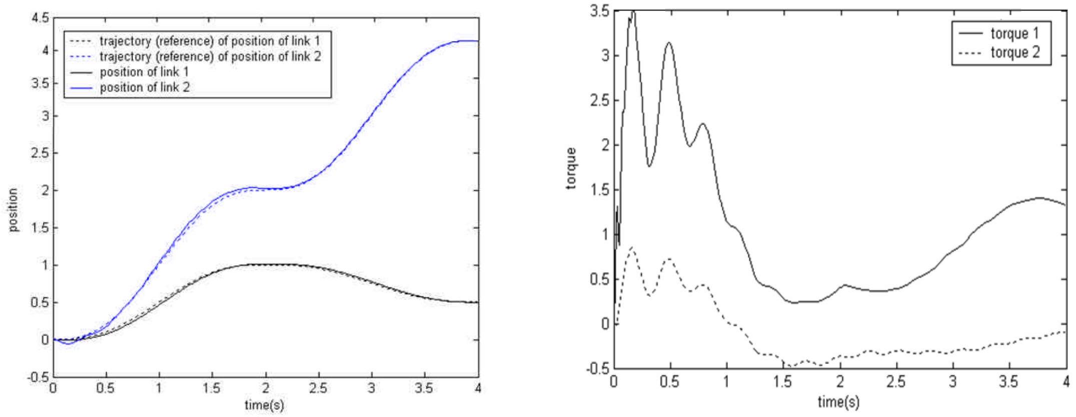


Figure 7. Response of robotic manipulator in closed loop with PID control (best f_1).

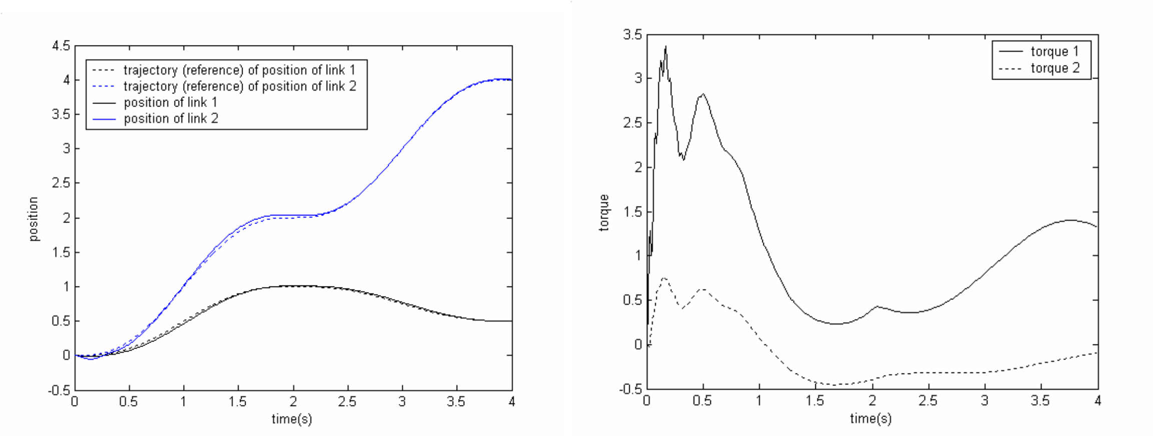


Figure 8. Response of robotic manipulator in closed loop with PID control (best f_2).

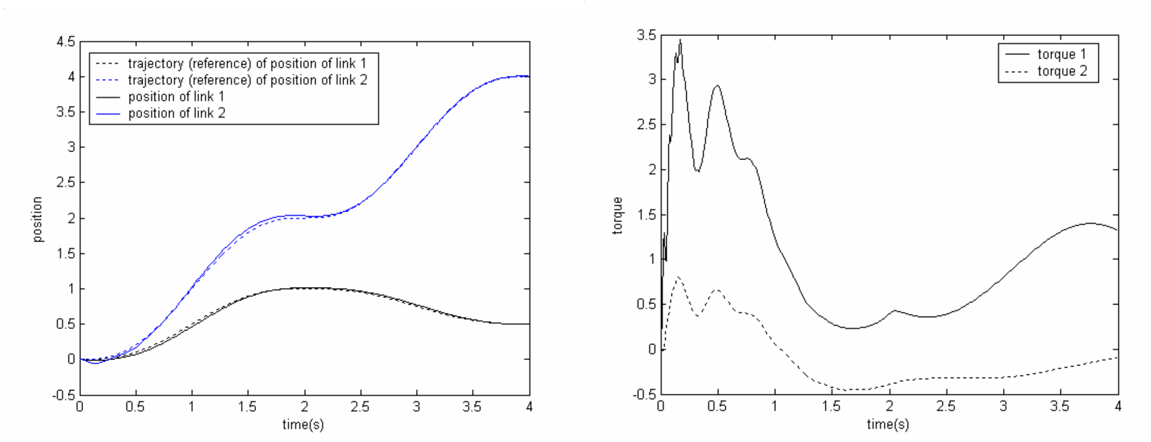


Figure 9. Response of robotic manipulator in closed loop with PID control (harmonic mean of f_1 and f_2).

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 (Åström and Hägglund, 1995).

The 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 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. The idea behind NSGA-II is that a ranking selection method is used to emphasize current nondominated points and sharing function method is used to maintain diversity in the population. 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 Figures 6 to 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 PID design.

6. ACKNOWLEDGEMENTS

This work was supported by the National Council of Scientific and Technologic Development of Brazil — CNPq — under Grant 309646/2006-5/PQ. The author would also like to thank the reviewers for their insightful comments and suggestions.

7. REFERENCES

Carvalho, J.R.H. and Ferreira, P.A.V., 1995, "Multiple-Criterion Control: A Convex Programming Approach", Automatica, Vol. 31, pp. 1025-1029.

Coello, C.A.C., 1999, "A Comprehensive Survey of Evolutionary-Based Multiobjective Optimization", Knowledge and Information Systems, Vol. 1, No. 3, pp. 269-308.

Coello, C.A.C., Van Veldhuizen, D.A. and Lamont, G.B., 2002, "Evolutionary Algorithms for Solving Multi-Objective Problems", Kluwer Academic Publishers, New York, NY, USA.

Craig, J.J., 1996, "Introduction to Robotics: Mechanics & Control", Chapter 6, Addison-Wesley.

Deb, K. and Agarwal, R.B., 1995, "Simulated Binary Crossover for Continuous Search Space", Complex Systems, Vol. 9, pp. 115-148.

Deb, K., 2001, "Multi-Objective Optimization using Evolutionary Algorithms" Wiley-Interscience Series in Systems and Optimization. John Wiley & Sons.

Deb, K., Pratap, A., Agarwal, S. and Meyarivan, T., 2002, "A Fast and Elitist Multi-Objective Genetic Algorithm: NSGA-II", IEEE Transactions on Evolutionary Computation, Vol. 6, No. 2, pp. 182-187.

Fonseca, C. and Fleming, P.J., 1993, "Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization", Proceedings of the 5th International Conference on Genetic Algorithms, pp. 416-423.

Fonseca, C.M. and Fleming, P.J., 1995, "An Overview of Evolutionary Algorithms in Multiobjejctive Optimization", Evolutionary Computation, Vol. 3, No. 1, pp. 1-16.

Goldberg, D.E., 1989, "Genetic Algorithms in Search, Optimization, and Machine Learning", Adison-Wesley, USA.

Hong, T.-P., Wang, H.-S., Lin, W.-Y. and Lee, W.-Y., 2002, "Evoluton of appropriate crossover and mutation operators in a genetic process, Applied Intelligence, Vol. 16, pp. 7-17.

Horn, J., Nafpliotis, N. and Goldberg, D.E., 1994, "A Niched Pareto Genetic Algorithm for Multi-Objective Optimization", Proceedings of the 1st IEEE Conference on Evolutionary Computation, IEEE World Congress on Computational Intelligence, Orlando, FL, USA, Vol. 1, pp. 82–87.

Khare, V., 2002, "Performance Scaling of Multi-Objective Evolutionary Algorithms", Master thesis, School of Coputer Science, The University of Birmingham, Birmingham, UK.

Liao, L.-Z. and Li, D., 2002, "Adaptice Differential Dynamic Programming for Multiobjective Optimal Control", Automatica, Vol. 38, pp. 1003-1015.

Liu, W. and Wang, G., 2000, "Auto-Tuning Procedure for Model-Based Predictive Controller", Proceedings of IEEE International Conference on Systems, Man, and Cybernetics, Nashville, Tennessee, USA, Vol. 5, pp. 3421-3426.

Mital, D.P. and Chin, L., 1995, "Intelligent Control Applications with Neural Networks", Intelligent control systems: theory and applications, Gupta, M. M. e Sinha,. N. K. (eds), Chapter 18, Piscataway: IEEE Press, USA, pp. 479-514.

Osyczka, A., 1985, "Multicriteria Optimization for Engineering Design", Academic Press.

Podlena, J. R. and Hendtlass, T., 1998, "An Accelerated Genetic Algorithm", Applied Intelligence, Vol. 8, pp. 103-111.

Salazar, D., Rocco, C.M. and Galván, B. J., 2006, “Optimization of Constrained Multiple-Objective Reliability Problems using Evolutionary Algorithms”, Reliability Engineering and System Safety, Vol. 91, No. 9, pp. 1057-1070.

Schaffer, J.D., 1985, “Multiple Objective Optimization with Vector Evaluated Genetic Algorithms”, Proceedings of the First International Conference of Genetic Algorithms, Pittsburgh, PA, USA, pp. 93-100.

Srinivas, N. and Deb, K., 1994, “Multi-Objective Function Optimization Using Nondominated Sorting Genetic Algorithms”, Evolutionary Computation, Vol. 2, No. 3, pp. 221-248.

Van Veldhuizen, D.A. and Lamont, G.B., 2000, “Multiobjective Evolutionary Algorithms: Analyzing the State-of-the-Art”, Evolutionary Computation, Vol. 8, No. 2, pp. 125-147.

Zambrano, D. and Camacho, E. F., 2002, “Application of MPC with Multiple Objective for a Solar Refrigeration Plant”, Proceedings of the IEEE International Conference on Control Applications, Glasgow, Scotland, UK, pp. 1230-1235.

Zitzler E., Laumanns M. and Bleuler S., 2002, “A Tutorial on Evolutionary Multiobjective Optimization”, Workshop on Multiple Objective MetaHeuristics (MOMH 2002), Paris, France.

8. RESPONSIBILITY NOTICE

The authors are the only responsible for the printed material included in this paper.