2008 American Control Conference
Westin Seattle Hotel, Seattle, Washington, USA
June 11-13, 2008

MPC and PID Control Based on Multi-objective Optimization

Adrian Gambier

Abstract— **The design of sophisticated control systems have led in the past ten years to the necessity of satisfying more than one design criterion. Thus, it is natural to think that those criteria can be met in an optimal manner. If several criteria have simultaneously to be optimized, one is in presence of a Multi-Objective Optimization problem.**

In this paper, many efforts to design the most popular control strategies, i.e. PID and MPC, by using multi-objective optimization techniques are reviewed. Both control strategies have dissimilar optimization characteristics and therefore, they can be considered as representative of two different multi-objective optimization problems, which are described including definitions, possible solutions, algorithms and available software implementations.

I. Introduction

In control engineering, Multi-Objective Optimization (MOO) has been used for a long time (pioneer works are for example [24], [40], and the references herein). However, MOO has intensively been applied to obtain optimal control systems in the last ten years. An excellent review of MOO applications in control engineering can be found in [41]. A recently update is given in [19].

PID control or PI control (Proportional, Integral and Derivative) and MPC (Model Predictive Control) can be considered the most popular control strategies as one can infer from the vast available literature. In the case of PID control, a small sample of books is for example [3], [13], [33], [62], [58] and [48]. Well-known books on model predictive control are for instance [9], [27], [29], [39], [42] and [55]. However, none of these books treat the multi-objective control problem. Only in [9] and [42], MOO ideas are shortly mentioned. In [41], it is treated the MOO-PID control but not the MOO-MPC.

In spite of the absence of MOO in the literature, MOO has been applied to design control systems based on PID as well as on MPC strategies in several opportunities. This is, for instance, the case of PI and PID controller design described in [30] as a MOO problem. A simplified goal-attainment formulation of MOO problem is used to tune PI and PID controllers in [37]. The design problem of a robust PID controller with two degrees of freedom based on the partial model matching approach is treated in [35]. In [8], a design procedure for tuning PID controller parameters to achieve a mixed $\mathcal{H}_2/\mathcal{H}_\infty$ optimal performance using genetic algorithms is described. All mentioned MOO problems deal with several objective functions that have to be satisfied by only one controller (SISO or MIMO) in a mono-loop control system. MOO methods for the simultaneous parameter optimization of PID controllers in several interacting control loops can be found for example in [20] and [63] for continuous-time systems and [21] for the discrete-time case.

A MOO framework for MPC has been proposed in [36]. The approach is based on a lexicographic algorithm taking advantage about the fact that for this method, objective functions can be ordered according to pre-established priorities. Another MOO-MPC approach is presented in [65], where the performance index is formulated as MOO problem and the goal attainment method is used to obtain the solution. In [34], a similar approach is used to solve a nonlinear MPC problem by using a NARX model.

Manuscript received September 21, 2007.
A. Gambier is with the Automation Laboratory, University of Heidelberg, B6, No. 26, 68131 Mannheim, Germany (phone: +49 621 181-2783; fax: +49 621 181-2739; e-mail: agambier@ieee.org).

In the current paper, controller design for MPC and PID strategies based on MOO techniques is reviewed including the most used MOO algorithms and software implementations. In Section II, optimal PID and MPC based on a unique performance index are shortly summarized. Section III is dedicated to present the essentials of MOO, i.e. definitions and most relevant methods. Available software for MOO is the subject of Section IV. In Section V, some works on MOO-PID and MOO-MPC are compared. Finally, Section VI is devoted to draw conclusions.

II. Optimal Control System Design

Control design based on the minimization of performance indices is an well established area in control engineering (see e.g. [6] and references herein). Conventional optimal control systems are obtained by optimization of a single performance index, which in general is defined as

$$J = J[\mathbf{e}(k_0),\cdots,\mathbf{e}(k),\mathbf{u}(k_0),\cdots,\mathbf{u}(k),\boldsymbol{\alpha}], \qquad (1)$$

subject to constraints

$$g_e[\mathbf{e}(k_0),\cdots,\mathbf{e}(k),\mathbf{u}(k_0),\cdots,\mathbf{u}(k),\boldsymbol{\alpha}] = 0 \text{ or} \qquad (2)$$

$$h_{ie}[\mathbf{e}(k_0),\cdots,\mathbf{e}(k),\mathbf{u}(k_0),\cdots,\mathbf{u}(k),\boldsymbol{\alpha}] \le 0, \qquad (3)$$

where $\mathbf{e}(\cdot)$ is the control error, $\mathbf{u}(\cdot)$ is the control or decision vector (controller's outputs) and $\boldsymbol{\alpha}$ is a parameter vector (normally containing controller parameters).

According to which argument J is optimized, two different optimization problems can be considered: (*a*) if J is optimized with respect to the parameter vector $\boldsymbol{\alpha}$, i.e.

$$J(\mathbf{e}(k),\mathbf{u}(k),\boldsymbol{\alpha}^{\mathrm{o}}) = \min_{\boldsymbol{\alpha}} J[\mathbf{e}(k_0),\cdots,\mathbf{e}(k),\mathbf{u}(k_0),\cdots,\mathbf{u}(k),\boldsymbol{\alpha}], \quad (4)$$

one is in presence of a *parameter optimization problem* and (1) is a cost function; (*b*) on the contrary, J can also be optimized with respect to the function $\mathbf{u}(k)$, i.e.

$$J(\mathbf{e}(k),\mathbf{u}^{\mathrm{o}}(k),\boldsymbol{\alpha}) = \min_{\mathbf{u}(k)} J[\mathbf{e}(k_0),\cdots,\mathbf{e}(k),\mathbf{u}(k_0),\cdots,\mathbf{u}(k),\boldsymbol{\alpha}]. \quad (5)$$

This is a *problem of calculus of variations* (or multi-stage optimization problem). That is a dynamic optimization problem, where (1) is now a *cost functional* and $\mathbf{u}^{\mathrm{o}}(k)$ is the *optimal control law*. Sometimes optimization problems can be solved analytically leading to an elegant closed solution, but in general optimization problems with constraints can only be solved numerically. Numerical methods to solve parameter optimization problems can be used in a repetitive fashion to solve problems of dynamic optimization. The most important limitation here is that the optimization must normally be carried out on-line. In this case, the problem is also subject to time constraints given by the sampling time.

The optimal PID controller design belongs to the first class of optimization problems, while MPC is a typical case of the second one. In the following subsections, standard approaches for optimal PID and MPC are shortly summarized.

A. Parameter-optimized PID Controllers

The idea of choosing PID controller parameters by minimizing an integral square cost function is not new. It is already proposed in [54] for a discrete-time adaptive PID controller. The continuous-time case is analogous. The problem then is to find parameters for the digital PID controller so that the performance index

$$J = \sum_{k=1}^{\infty} e^2(k) + \lambda\, \Delta u^2(k) \qquad (6)$$

or equivalently (according to the Parseval's theorem)

$$J = \frac{1}{2\pi j}\oint [e(z)\,e(z^{-1}) + \lambda\,\Delta u(z)\,\Delta u(z^{-1})]z^{-1}dz \qquad (7)$$

is minimized.

The parameter λ is referred to a *Lagrange multiplier* and it can be exactly determined. However, it is usually assumed that λ is a free design parameter by mean of which control signal amplitudes can be softly limited. Thus, the index (6) can be seen as a *weighted sum* objective function. The digital PID control law is given by

$$P(z^{-1})u(z) \;=\; Q(z^{-1})e(z)\,, \qquad (8)$$

where P and Q are the polynomials

$$P(z^{-1}) = 1 - z^{-1} \;\text{ and }\; Q(z^{-1}) \;= q_0 + q_1 z^{-1} + q_2 z^{-2}\;. \quad (9)$$

Variable $e(z)$ is the control error $e(z) = r(z) - y(z)$. Moreover, the coefficients q_i must satisfy the inequality (see [31] for details)

$$\begin{bmatrix} -1 & 0 & 0 \\ 1 & 1 & 0 \\ -1 & -1 & -1 \\ -1 & 0 & 1 \end{bmatrix} \begin{bmatrix} q_0 \\ q_1 \\ q_2 \end{bmatrix} \le \mathbf{0} \qquad (10)$$

in order to hold the step response characteristic of a continuous-time PID controller. If it is assumed that the plant is modelled by

$$A(z)\,y(z) = B(z)\,u(z) \qquad (11)$$

and the controller design is carried out for a step setpoint $r(z^{-1}) = 1/(1 - z^{-1})$. Hence, $e(z)$ and $\Delta u(z)$ are given by

$$e(z) = \frac{A(z)}{(1-z^{-1})A(z) + Q(z)B(z)} \;\text{ and }\; \Delta u(z) = \frac{A(z)Q(z)}{(1-z^{-1})A(z) + Q(z)B(z)} \quad (12)$$

respectively. The unknown coefficients of Q, $\mathbf{q} = [q_0,\, q_1,\, q_2]^{\mathrm{T}}$ are obtained by using a numerical optimization algorithm as e.g. Fletcher-Powell or Hooke-Jeeves. The index (7) can be evaluated by using the algorithm given in [2] (see [18] for the evaluation of this and other square criteria). Parameter λ is normally difficult to select and therefore chosen by trial and error.

B. Model Predictive Control

Model Predictive Control (MPC) comprises a family of controllers signed by the same design philosophy and similar characteristics (see e.g. [9], [42] and [55]). The design is based on three main steps: (*i*) the use of a model to predict the output vector $\mathbf{y}$, N step in the future, (*ii*) the optimization of a quadratic performance index during a finite period of time delimited for the control error by the prediction horizon N and by N_u for the control action, i.e.

$$J = \| \mathbf{e}(k+N) \|_{\mathbf{S}}^2 + \sum_{i=k}^{k+N-1} \| \mathbf{e}(i) \|_{\mathbf{Q}}^2 + \sum_{i=k}^{k+N_u} \| \Delta\mathbf{u}(i) \|_{\mathbf{R}}^2 \qquad (13)$$

(where $\mathbf{S} = \mathbf{S}^{\mathrm{T}}$ and $\mathbf{Q} = \mathbf{Q}^{\mathrm{T}}$ are positive semidefinite matrices and $\mathbf{R} = \mathbf{R}^{\mathrm{T}}$ is a positive definite matrix; $\| \mathbf{p} \|_{\mathbf{M}}^2 = \mathbf{p}^{\mathrm{T}}\mathbf{M}\,\mathbf{p}$, $\mathbf{e}(\cdot) = \mathbf{r}(\cdot)\text{-}\mathbf{y}(\cdot)$, $\Delta\mathbf{u}(\cdot) = \mathbf{u}(\cdot) - \mathbf{u}(\cdot\ \text{-}1)$. $\mathbf{y} \in \mathbb{R}^m$, $\mathbf{r} \in \mathbb{R}^m$ and $\mathbf{u} \in \mathbb{R}^l$ are the output vector, reference vector and input vector, respectively) and (*iii*) the application of the *receding horizon principle*, i.e. all control signals $\mathbf{u}(k) \cdots \mathbf{u}(k+N_u)$ are calculated and only $\mathbf{u}(k)$ is applied, in the next step the sequence is calculated again taking into account new values for $\mathbf{y}$ and $\mathbf{r}$ and again, only the new $\mathbf{u}(k)$ is used and so on. The solution of the unconstrained problem is given by

$$\Delta\mathbf{u}(k) = [\mathbf{I}\;\mathbf{0}\cdots\mathbf{0}][\mathbf{R}_e + \mathbf{G}^{\mathrm{T}}\mathbf{Q}_e\mathbf{G}]^{-1}\mathbf{G}^{\mathrm{T}}\mathbf{Q}_e\,(\bar{\mathbf{r}} - \mathbf{f})\,. \qquad (14)$$

where $\mathbf{Q}_e$ = diag ($\mathbf{S}$, $\mathbf{Q}$, $\cdots$, $\mathbf{Q}$), $\mathbf{Q}_e$ = diag ($\mathbf{R}$, $\cdots$, $\mathbf{R}$), $\mathbf{G}$ is a Toeplitz matrix containing the step response coefficients of the plant, $\bar{\mathbf{r}} = [\mathbf{r}(k)\ \cdots \mathbf{r}(k+N)]^{\mathrm{T}}$ and $\mathbf{f}$ contains terms of the free response of the predictor and depends on the used model.

One of the most important advantages of MPC is the ability to handle explicitly constraints in the optimization process, but in this case the optimization cannot be carried out analytically. The solution has numerically to be found on-line at every time step.

III. Multi-objective Optimization

A. General Definitions

Multi-objective optimization (also known as multi-performance, multi-criteria or vector optimization) was introduced by V. Pareto ([49]) and it can be defined as the problem of finding a vector of decision variables (or parameters), which satisfies constraints and optimizes a vector field, whose elements represent objective functions. In general, the MOO problem can be formulated as follows:

Find	$\mathbf{u} = [u_1 \cdots u_l]^{\mathrm{T}}$ or $\boldsymbol{\alpha} = [\alpha_1 \cdots \alpha_{np}]^{\mathrm{T}}$
to optimize	$\mathbf{J}(\mathbf{u},\boldsymbol{\alpha}) = [J_1(\mathbf{u},\boldsymbol{\alpha}) \cdots J_{nf}(\mathbf{u},\boldsymbol{\alpha})]^{\mathrm{T}}$
with respect	to $\mathbf{u}$ or to $\boldsymbol{\alpha}$
subject to	$g_i(\mathbf{u},\boldsymbol{\alpha}) \le 0$ and $h_j(\mathbf{u},\boldsymbol{\alpha}) = 0$,
for	$i = 1, \cdots, ng,\ j = 1, \cdots, nh$,

where $\mathbf{J} \in \vartheta \subset \mathbb{R}^{nf}$ is the objective vector field, $\mathbf{u} \in \mathbf{U} \subset \mathbb{R}^l$ is a vector of decision variables, $\boldsymbol{\alpha} \in \Lambda \subset \mathbb{R}^{np}$ is a vector of fixed parameters, nf is the number of objective functions, ng is the number of inequality constraints and nh is the number of equality constraints. Optimize means here either minimize or maximize depending on the application. If all J_i are convex, all inequality constraints g_i are convex and all equality constraints h_j are affine, the vector optimization problem is also called *convex*.

Contrary to single-objective optimization (SOO), there is for MOO no single global solution and it is often necessary to determine a set of points that all fit a predetermined definition for the optimum. Usually, it is accepted as multi-objective optimality the definition given by Pareto ([49]), which is stated as follows:

Definition **1**: A point, $\mathbf{u}^\circ \in \mathbf{U} \subset \mathbb{R}^l$, is *Pareto optimal* with respect to $\mathbf{U}$ iff there does not exist another point, $\mathbf{u} \in \mathbf{U}$, such that $\mathbf{J}(\mathbf{u},\alpha) \le \mathbf{J}(\mathbf{u}^\circ,\alpha)$ and $J_i(\mathbf{u},\alpha) < J_i(\mathbf{u}^\circ,\alpha)$ for at least one function, i.e. there is no way to improve upon a Pareto optimal point without increasing the value of at least one of the other objective functions. Notice that the definition given above can be applied to the parameter vector $\boldsymbol{\alpha}$ instead of the vector of decision variables $\mathbf{u}$ and this is also valid for the next definitions.

Sometimes, it is useful to have a definition for a suboptimal point that is easier to be reached by the algorithms and simultaneously is sufficient '*good*' for practical applications. This is obtained e.g. from the Weakly Pareto Optimality:

Definition **2**: A point, $\mathbf{u}^\circ \in \mathbf{U} \subset \mathbb{R}^l$, is *weakly Pareto optimal* iff there does not exist another point $\mathbf{u} \in \mathbf{U}$, such that $\mathbf{J}(\mathbf{u},\alpha) < \mathbf{J}(\mathbf{u}^\circ,\alpha)$.

In other words, a point is weakly Pareto optimal if there is no other point that improves all of the objective functions simultaneously. Hence, Pareto optimal points are weakly Pareto optimal, but weakly Pareto optimal points may not be Pareto optimal. For any given problem, there may be an infinite number of Pareto optimal points, which constitute the *Pareto optimal set*, i.e.

$$\wp \triangleq \left\{ \mathbf{u}^\circ \in \mathbf{U} \subseteq \mathbb{R}^l \;|\; \neg\exists \mathbf{u} \in \mathbf{U}, \forall i\, J_i(\mathbf{u},\boldsymbol{\alpha}) \le J_i(\mathbf{u}^\circ,\boldsymbol{\alpha}) \wedge \exists i : J_i(\mathbf{u},\boldsymbol{\alpha}) < J_i(\mathbf{u}^\circ,\boldsymbol{\alpha}) \right\}. \quad (15)$$

The Pareto optimal set $\wp \subset \mathbf{U}$ has an image in the criterion space ϑ, which is denoted here as $\wp f$ and is called *Pareto front*.

Definition **3**: For a given vector objective function $\mathbf{J}(\mathbf{u},\alpha)$ and a Pareto- optimal set $\wp$, the Pareto front is defined as follows:

$$\wp f \triangleq \left\{ \mathbf{J}(\mathbf{u}_i, \boldsymbol{\alpha}) \in \vartheta \subseteq \mathbb{R}^{nf} \;|\; \mathbf{u}_i \in \wp, \forall\, i \right\}, \qquad (16)$$

An important point in the criterion space is the *utopia point*, which is defined in the following:

Definition **4**: A point, $\mathbf{J}^\circ$ is a *utopia point* iff $J_i^\circ = \min J_i\,(\mathbf{u}, \boldsymbol{\alpha})$ with respect to $\mathbf{u}$ for each $i = 1, \cdots, nf$.

The set of Pareto-optimal solutions is also called *non-inferior*, *non-dominated*, *admissible*, or *efficient* solutions. When the non-dominated vectors are collectively plotted in the criterion space, they constitute the Pareto front. Fig. 1 illustrates with a three/two dimensional example all concepts introduced above.

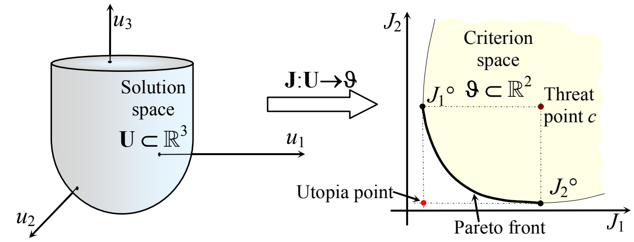

Fig. 1. Evaluation mapping of the multiobjective problem

B. Decision Making and Control Performance

All points of the Pareto front are equally acceptable solution of the vector optimization problem. However, it is necessary to obtain only one point in order to be able to implement the controller. This selection is carried out by a *decision maker*. Although decision making is a crucial aspect in the design, it seldom mentioned in the MOO control literature. However, decision making is a world of its own and therefore only a general idea is given here.

The decision making can be undertaken from two different points of view: *i*) by including additional criteria such that at the end only one point satisfies all of them, and *ii*) by considering one point that represents a fair compromise between all used criteria.

The first option allows introducing additional criteria that could be oriented to improve the control performance. For example, if the Pareto front corresponds to $\mathbf{J} = (J_{ISE}, J_{ISC})$, eq. (19), the final solution can be obtained for the point that leads to the minimum overshoot or the maximum rise time. It is also possible to start a min-max optimization problem with domain in the Pareto set in order to find the solution for the maximum rise time and the minimum overshoot. In [64], it is selected the point whose parameters minimize the structured singular value μ such that the obtained controller is the most robust contained within the Pareto set.

The second option does not introduce more information for the decision making and only a fair point for all indices is searched. The ideal case would be to reach the utopia point. However, it is difficult to optimize all individual objective functions independently and simultaneously. Moreover, the utopia point $\mathbf{J}°$ is normally unattainable and it is not in ϑ. Thus, it is only possible to find a solution that is as close as possible to the utopia point. Such solution is called *compromise solution* (*CS*) and is Pareto optimal.

Two more efficient procedures to select a fair point are *cooperative negotiation* ([22]) and *bargaining games* ([61]). Solutions of a bargaining game lead to some practical procedure for choosing a unique point (see Fig. 2). For example (see [61]), the *Nash* solution of the game (*NS*) corresponds to a point of the Pareto set which yields the largest rectangle (*c*, *B, NS, A*), the *Kalai-Smorodinsky* solution (*KS*) is situated at the intersection of the Pareto front and the straight line, which connects the threat point and the utopia point, and the *egalitarian* solution (*ES*) yields the point given by the intersection of the Pareto front and a 45°-line through the threat point.

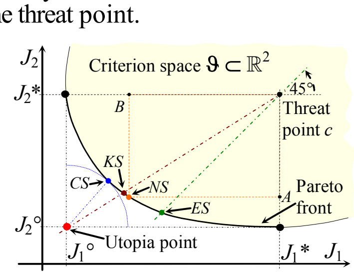

Fig. 2. Different criteria for the decision making

However, the complexity due to the large number of objective functions and temporal deadlines, within which the optimization must normally be accomplished, reduce the applicability of classical game theory techniques to design optimal bargaining models for decision making.

C. Most Important MOO Methods

At the present, a very huge number of methods to solve MOO problems can be found in the specialized literature. Broad reviews can be found e.g. in [1] and in [43]. In this work, only the most important methods, which have been used to solve PID/MPC control problems, will be included for sake of space.

Methods to solve MOO problems can be classified according to a wide spectrum of characteristics (see [44]). In Fig. 3, a classification based on [14] is given. The two main groups are: (a) *Scalarization methods* and (b) the *Pareto methods*.

Scalarization methods require the formation of an overarching objective function aggregating contributions from all components of the objective vector, normally by using coefficients, exponents, constraint limits, etc. and then methods for single objective optimization are used to find a unique solution. They are very efficient and fast to find a unique solution. On the other hand, they do not always give acceptable solutions because of interest conflicts of design objectives. Moreover, these methods can converge to a local optimum and therefore they are not able to find the global solution. Finally, it is not always clear for the user, how to express the preferences for the scalarization process.

Pareto methods first find the optimal solutions space and then a unique optimal solution from the Pareto optimal set is chosen by a decision maker. Thus, these methods keep the elements of the objective vector separate throughout the optimization process and use the concept of dominance to distinguish between inferior and non-inferior solutions. The advantage of Pareto methods consists in the fact that the preferences can be expressed once the optimization is already carried out, keeping the different objective functions separately. Therefore, they are able to take care of all conflicting design objectives individually but compromising them concurrently. However, the search process requires a very high computation burden and the convergence can be very slow.

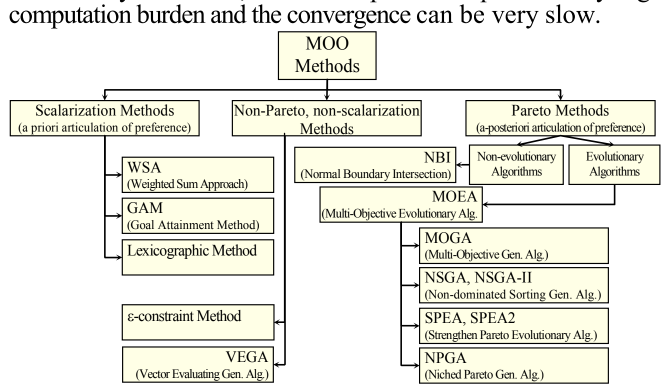

Fig. 3. General classification of MOO solving methods.

Remark: In addition, it is possible to find methods *without articulation of preference* and methods with *progressive articulation of preference*, also called *iterative methods* (see [1] for details).

In general, it is difficult to recommend a particular method and the choice mostly depends on the application. As a rule, some authors ([43]) suggest to select first methods, which guarantee necessary and sufficient conditions for Pareto optimality. Then, methods those guarantee only sufficient conditions and finally other ones.

1) Weighted Sum Approach

This is probably the most widely used MOO method. It consists in assigning a non-negative weight γ_i to each of the i objective functions, so that the overarching scalar objective function can be expressed as

$$J = \boldsymbol{\gamma}^{\mathrm{T}} \mathbf{J}(\mathbf{u}, \boldsymbol{\alpha}) \,, \quad (17)$$

where $\boldsymbol{\gamma} = [\gamma_1 \cdots \gamma_{nf}]^{\mathrm{T}}$ is the weight vector. The other variables were already defined in Subsection A. Weights are chosen in such a form that $\sum |\gamma_i| = 1$. Moreover, the objective functions have to be normalized since not all objectives have the same range of values. The optimal solution is given by

$$J^{\circ} = J(\mathbf{u}^{\circ}, \boldsymbol{\alpha}) = \min_{\mathbf{u}} \boldsymbol{\gamma}^{\mathrm{T}} \mathbf{J}(\mathbf{u}, \boldsymbol{\alpha}) \,. \quad (18)$$

The most important advantage of this method is the transformation of the vector objective function in a single-objective function, such that traditional optimization methods can be used. The problem is the setting of the weights: On one hand, results are sensitive to weights ratio and they are difficult to be chosen. On the other hand, weights indicate the relative importance of the corresponding objective function but they do not mean priorities. Hence, if the optimization process cannot be completed for all objective functions, the method does not indicate, in which sequence objective functions may be discarded. Moreover, the method presents difficulties in case of non-convex problems.

2) Goal Attainment Method ([23])

This method is formulated as follows:

find	$\mathbf{u}$, ξ
to minimize	ξ
subject to	$J_i(\mathbf{u},\alpha) - \gamma_i\, \xi \leq g_i$, for $i = 1, \ldots, nf$,

where $J_i(\mathbf{u},\alpha)$ are objective functions, γ_i are weights indicating the relative importance of each objective function, g_i are the goals, which have to be reached, and ξ is an unrestricted scalar.

This method presents similar properties as the weighted sum approach, where the most important difficulty is to select the weights, which, in turn, do not correspond to priorities.

3) Lexicographic method ([4])

In this method, objective functions are arranged in order of importance by the user. The optimum solution $\mathbf{u}^{\circ}$ is then obtained by minimizing the objective functions, starting with the most important one and proceeding according to the order of importance assigned to the objectives. Thus, the following optimization problems are solved one at a time:

minimize	$J_i(\mathbf{u},\alpha)$ respect to $\mathbf{u} \in \mathbf{U}$
subject to	$J_j(\mathbf{u},\alpha) \leq J_j(\mathbf{u}_j^{\circ},\alpha)$,
for	$j = 1, 2, \cdots, i\text{-}1,\ i > 1,\ i = 1, 2, \cdots, nf.$

The subscript i represents here not only the objective function number but also the priority assigned to the objective, and $J_j(\mathbf{u}_j^{\circ},\alpha)$ represents the optimum of the jth objective function, found in the jth iteration. Notice that after the first iteration $J_j(\mathbf{u}_j^{\circ},\alpha)$ is not necessarily the same as the independent minimum of $J_j(\mathbf{u},\alpha)$, because of the introduction of new constraints.

The advantage of this method is that there is only one optimum for a given lexicographic order and it is very easy to implement because of the sequential optimization. Its disadvantage is that normally objectives with lower priorities will not be satisfied. Thus, priority assignment, which has to be done a priori, is crucial in order that the methods would be successful.

4) Multiobjective Evolutionary Algorithms

Evolutionary optimization algorithms simulate the survival of the fittest in biological evolution by means of algorithms. The renewal of a population (entire set of variables that represent a group of potential solution points) is based on the so called 'genetic operators': *recombination* (out of two points of the population picked out so that a new point is generated, e.g. by averaging), *mutation* (single, randomly selected digits of a newly generated point are substituted by a realization of a random variable) and *selection* (out of the union of the original population and the newly generated points, which are taken over into the new population with the best *fitness*). There are many algorithms to implement this functions (see [38] and [66] for comparative reviews).

Evolutionary algorithms can be applied to solve SOO problems as well as MOO problems. The MOO case was first studied in [26]. Today, there are many MOEAs distinguished mainly by the algorithms for the population ranking in the fitness assignment. The most important are: MOGA (Multiple Objective Genetic Algorithm, [17]), *NSGA-II* (Non-dominated Sorting Genetic Algorithm, [15]), *SPEA2* (Strength Pareto Evolutionary Algorithm, [67]), *NPGA-II* (Niched Pareto Genetic Algorithm, [16]).

A very important advantage of the MOEAs is that they do not need information about the objective-functions derivatives and they can solve non-convex problems. Moreover, the algorithms are relatively robust and they do not require solving a sequence of single-objective problems and it is a parallel search technique. However, Pareto optimality is not a concept embedded in the fundaments of evolutionary algorithms. Consequently, it could be that a Pareto optimal solution born and then, by chance, also it dies out. An additional drawback is the intensive computational burden required until the solution is obtained.

5) Other Algorithms

Finally, the NBI method (Normal Boundary Intersection) was proposed in [12]. It is a very fast Pareto method that does not belong to the family of evolutionary algorithms. The VEGA (Vector Evaluated Genetic Algorithm, [56]) is also an algorithm for MOO based on a genetic algorithm but it is neither a scalarization nor Pareto based algorithm.

IV. Available Software Implementations

In order to be able to implement multi-objective optimal control systems it is reasonable to ask about the state of the art of the development of software for such task. A short review about available software can be found in [44]. Updated information is also given in the web (for example in [11] and [45]).

Because the MATLAB™ software system is a very common environment for control engineers and students and also for the sake of space, this review is limited to code available for this package. Available toolboxes for MOO can be divided in two groups: commercial code and free code. Both will be presented in the following two subsections without intending to be exhaustive because this field changes dynamically.

A. Commercial Code for MOO Problems

Matlab itself bring support for MOO by means of its *Optimization Toolbox*™ ([5]). It includes the weighted sum method, the ε-constraint method and the goal attainment method. Another general purpose development environment in Matlab for optimization problems is TOMLAB ([28]). Its support for MOO is given through FORTRAN routines that are acceded by means of a MEX interface. However, this software does not support Pareto methods. Implemented methods are e.g. weighted sum, ε-constraint, hierarchical optimization, min-max and global criterion. In [32], the software package MOPS (Multi-Objective Parameter Synthesis) is presented. The MOO support is based on min-max MOO, which is solved by reformulating it as a non-linear programming problem. Matlab is also supported by a MEX interface.

A comprehensive implementation of evolutionary algorithms in Matlab is given by the GEA Toolbox (GEATbx, [51]), where multi-objective ranking of MOGA ([17]) and goal attainment are completely implemented.

B. Free Code for MOO Problems

Free Matlab code for MOO problems, which is not genetic-algorithm based, is difficult to find. In fact, only the NBI Toolbox ([12]) could be found by the author. However, this toolbox needs the Optimization Toolbox and therefore it cannot be considered completely free.

On the contrary, several free toolboxes for MOO are available for evolutionary algorithms. In [57], a small but complete package that implements the NSGA-II algorithm and some examples is available. Moreover, a complete package named SGALAB including many algorithms (SPEA2, NSGA-II, VEGA, MOGA, etc.) is currently being developed. A beta version can be downloaded from [10]. The drawback of this software consists in the fact that the Matlab source code is not available with exception of the examples. In addition, a small package based on NSGA is available in [53].

Finally, the MOEA toolbox described in [60] is no longer available either on the web or by writing to the author.

V. Multi-objective Control System Design

The controller design based on the optimization of performance indices (6) and (13) is actually the solution of a MOO problem since (6) and (13) can be considered as weighted sums of objective functions. The a-priori selection of weights yields a unique solution like a method with a-priori articulation of preferences. If the weighted sums are decomposed in its components, the vector performance indices

$$\mathbf{J}=\left[\sum_{k=1}^{\infty} e^{2}(k) \;\middle|\; \sum_{k=1}^{\infty} \Delta u^{2}(k)\right]^{\mathrm{T}} \text{ and } \tag{19}$$

$$\mathbf{J}=\left[\|\mathbf{e}(k+N)\|_{\mathbf{S}}^{2}+\sum_{i=k}^{k+N-1}\|\mathbf{e}(i)\|_{\mathbf{Q}}^{2} \;\middle|\; \sum_{i=k}^{k+N_u}\|\boldsymbol{\Delta}\mathbf{u}(i)\|_{\mathbf{R}}^{2}\right]^{\mathrm{T}} \tag{20}$$

are obtained, revealing the multi-objective nature of the problem. Moreover, the MPC with constraints can also be analyzed as a MOO problem because constraints can normally be formulated as additional objective functions with priorities. The aim of this Section is to describe some works, where MOO techniques are used for a particular goal. The summery is presented in Table I.

A. Multi-objective PID Control

Maybe one of the most important contributions in the field of MOO-PID is presented in [50], where several classic measures in time domain (overshoot, rise time, settling time) together with IAE index (Integral absolute error) are simultaneously optimized by using an evolutionary algorithm. A similar approach is presented in [30] but the solution is obtained by using a gradient MOO algorithm. In [59], the idea was just to eliminate weighting factors by using a two-objective optimization problem for the continuous time case. The solution is then found by using a genetic algorithm. In [52], a similar approach for the continuous-time case but for an objective function based on $|u|$ is proposed.

Finally, a PI controller is optimized in [37] by using special objective functions (Weighted Integral Square Error, WISE, gain and phase margins), which are derived for first order plus delay time (FOPDT) models and integrator plus delay time (IPDT) models. The MOO problem is solved by using the Goal Attainment Method.

B. Multi-objective Predictive Control

A MOO framework for MPC has been proposed in [36] and applied in [47]. The approach is based on a lexicographic algorithm taking advantage about the fact that for this method, objective functions can be ordered according to pre-established priorities. As example, three objective functions are given in addition to the standard cost index J_4 given by (13): the *size of the largest constraint violation*

$$J_1(\mathbf{g}) \triangleq \max\{0; g_1(\mathbf{u},\mathbf{x}); g_2(\mathbf{u},\mathbf{x}); \;\cdots; g_{ng}(\mathbf{u},\mathbf{x})\}, \tag{21}$$

where $\mathbf{g}$ models the constraints, $\mathbf{x}$ is the vector of state variables and $\mathbf{u}$ the control vector, the *weighted sum of constraint violations*

$$J_2(\mathbf{g}) \triangleq \mathbf{g}^{+}(\mathbf{u},\mathbf{x})^{\mathrm{T}}\mathbf{M}\,\mathbf{g}^{+}(\mathbf{u},\mathbf{x})+\boldsymbol{\nu}^{\mathrm{T}}\mathbf{g}^{+}(\mathbf{u},\mathbf{x}), \tag{22}$$

with $\mathbf{g}^{+}(\mathbf{u},\mathbf{x}) \triangleq \max\{0, g_i(\mathbf{u},\mathbf{x})\}$, $\mathbf{M} > \mathbf{0}$ and $\boldsymbol{\nu} > \mathbf{0}$, and the *largest element in index set of violated constraints*

$$J_3(\mathbf{g}) \triangleq \begin{cases} 0 & \text{if } \mathbf{g}(\mathbf{u},\mathbf{x}) \le \mathbf{0} \\ \max_i \{i \mid g_i(\mathbf{u},\mathbf{x}) > 0\} & \text{otherwise} \end{cases}. \tag{23}$$

This MOO approach for MPC contributes to improve the feasibility of the algorithm since constraints can be relaxed according to on-line assignable priorities when their satisfaction is not strictly necessary. The cost, which must be paid for this advantage, is an increased computational burden. This topic has still to be studied. Other approaches of MOO-MPC are given in [7] and in [65] in order to implement a control system for nonlinear system; in [25] to obtain robustness and in [46] to satisfy objectives with different priorities.

VI. Concluding Remarks and Final Discussion

In this paper, a short overview about MOO methods and their application to PID and MPC is presented. In general, the multi-objective formulation of control problems seems to be a very attractive approach in order to improve control systems in many directions and it appears as a promising methodology for the near future. However, the state of the art of algorithms for MOO allows concluding that at the present time not all optimal control problems can be formulated as a MOO problem: Multi-objective parameter optimization problems can efficiently be solved by the existing algorithms since the optimization is habitually carried out off-line. This is, for example, the case of parameter optimization of PID controllers.

TABLE I
SHORT SUMMERY ABOUT THE MOST IMPORTANT APPLICATION OF MOO IN CONTROL ENGINEERING

References	Aim for using MOO	Most important objective functions	Used MOO	Real-Time
MOO-PID				
[50]	Stabilize the system + minim. performance measures	Overshoot, rise time, settling time, IAE	NSGA-II	yes
[52]	Control of a non-linear plant with a fixed-gain PID	ISE + ISC	SPEA2	yes
[30]	Simultaneously satisfaction of many objectives	Overshoot, rise time	Gradient	No
[37]	Simultaneously satisfaction of many objectives	Gain and phase margins, WISE	Goal Attainment	No
[59]	Elimination of weighting factors	ISE + ISDC	MOGA	yes
MOO-MPC				
[36], [47]	Fault tolerant control	Duration of constraint relaxations, square norm of deviations, size of largest constraints violation	Lexicographic	No
[7]	Alternative to a multi-model control scheme	Eq. (13) for each neural network (nonlinear system)	WARGA, NSGA	No
[46]	To satisfy objectives with different priorities	Absolute values of deviations of outputs and control signals	Not mentioned	No
[65]	Control of nonlinear systems with linear controllers	Sum of square errors, sum square of Δu	Goal Attainment	No
[34]	Solving a problem of application	Energy consumption, filtration time of pulse jet fabric filters	Goal Attainment	No
[25]	Robust control system	Eq. (13) for each linearized model of the nonlinear system	Weighted sum	No

On the contrary, controllers, which calculate the control action on-line like MPC, can today use MOO techniques in case of systems with slow dynamic as many applications of process control. Electromechanical as well as mechatronical systems will probably have to wait for more powerful hardware and more efficient algorithms.

Thus, the real-time use of MOO algorithms is, in general, not possible at the moment. Moreover, decision making for control applications is also a field that needs more ideas and considerable more research.

REFERENCES

[1] Andersson, J., A survey of multiobjective optimization in engineering design. Technical report LiTH-IKP-R-1097, Department of Mechanical Engineering, Linköping University, Linköping, Sweden, 2000.

[2] Åström, K.J., E.I. Jury and R.G. Agniel, A numerical method for the evaluation of complex integrals. IEEE Transactions on Automatic Control, AC-15, 468–471, 1970.

[3] Åström, K. J., and T. Hagglund, Advanced PID Control. ISA - The Instrumentation, Systems, and Automation Society, 2005.

[4] Ben-Tal, A., Characterization of Pareto and lexicographic optimal solutions. Lecture Notes in Economics and Mathematical Systems, G. Fandel, and T. Gal (Eds.), **177**, 1-11, Springer, Berlin, 1980.

[5] Branch, M.A. and A. Grace, MATLAB Optimization Toolbox User's Guide, Version 1.5, The Mathworks, 1996.

[6] Bryson A. E., Optimal Control - 1950 to 1985, *IEEE Control System Magazine*, **16**, 26-33, 1996.

[7] Bouani, F., K. Laabidi, , and M. Ksouri, Constrained nonlinear multi-objective predictive control. *Proc. of the IMACS Multiconference on Computational Engineering in Systems Applicationsm* (*CESA*), October 4-6, Beijing, 2006.

[8] Calistru, C. N., Mixed $\mathcal{H}_2/\mathcal{H}_\infty$ PID robust control via genetic algorithms and MATHEMATICA facilities. Proc. of *the 2nd European Symposium on Intelligent Techniques*, Crete, 1999.

[9] Camacho, E. F. and C. Bordons *Model predictive control.* Springer: Advanced Textbooks in Control and Signal Processing Series, London, 2004**.**

[10] Chen, Y., SGALAB 1003: Genetic algorithm toolbox. *Matlab Central File Exchange*: http://www.mathworks.com/matlabcentral/, 2006.

[11] Coello-Coello, C. A., *EMOO website*: http://www.lania.mx/~ccoello/EMOO, 2006.

[12] Das, I. and J. Dennis, Normal-Boundary Intersection: A new method for generating Pareto optimal points in multicriteria optimization problems. *SIAM Journal on Optimization*, **8**, 631-657, 1998.

[13] Datta, A., M. T. Ho and S. P. Bhattacharyya, *Structure and synthesis of PID controllers*. Springer Verlag, New York, 2000.

[14] de Weck, O. L., Multiobjective optimization: history and promise. Keynote Paper. The 3rd China-Japan-Korea Joint Symposium on Optim. of Structural and Mechanical Systems, Kanazawa, Japan, October 30-November 2, 2004.

[15] Deb, K., S. Agrawal, A. Pratab and T. Meyarivan. A fast elitist non- dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. In: Lecture Notes in Computer Science, (M. Schoenauer, K. Deb, G. Rudolph, X. Yao, E. Lutton, J. J. Merelo, and H-P. Schwefel, Eds.), Springer, No. 1917, 849-858, 2000.

[16] Erickson, M., A. Mayer and J. Horn. The niched pareto genetic algorithm 2 applied to the design of ground water remed. systems. In EMO, 681-695, 2001.

[17] Fonseca, C. M., Multiobjective genetic algorithms with application to control engineering problems. Ph.D. Thesis, University of Sheffield, Sheffield, U.K., 1995.

[18] Gambier, A., Parametric optimization for practical control systems design. *Proc. of the 2007 IEEE Conf. on Control Applications,* Singapore, October 1-3, 2007.

[19] Gambier, A., and E. Badreddin, Multi-objective optimal control: An overview. *Proc. of the 2007 IEEE Conf. on Control Applic.,* Singapore, October 1-3, 2007.

[20] Gambier, A., A. Wellenreuther and E. Badreddin, A new approach to design multi-loop control systems with multiple controllers. *Proc. of the 45th IEEE Conference on Decision and Control*, San Diego, 2006.

[21] Gambier, A., A. Wellenreuther and E. Badreddin. Multi-loop control system design: A game theoretical approach for computer-controlled systems. *Proc. of the* 46th *IEEE CD*, New Orleans, December 12-14, 2007.

[22] Gatti, N. and F. Amigoni, An approximate Pareto optimal cooperative negotiation model for multiple continuous dependent issues. Proc. of the *2005 IEEE/WIC/ACM Int. Conf. on Intelligent Agent Technology*, Compiègne, France, 565 – 571, September 19-22, 2005.

[23] Gembicki, F.W., Vector optimization for control with performance and parameter sensitivity indices, Ph.D. Thesis, Case Western Reserve University, Cleveland, Ohio, 1974.

[24] Giesy, D. Calculation of Pareto-optimal solutions to multiple-objective problems using threshold-of-acceptability constraints. *IEEE Transactions on Automatic Control*, **23**, 1114-1115, 1978.

[25] Giovanini, L. and M. Grimble, Robust predictive feedback control for constrained systems. International Journal of Control, Automation, and Systems, **2**, 407-422, December 2004.

[26] Goldberg, D. E. Genetic algorithms in search, optimization, and machine learning. Addison-Wesley, Reading, Massachusetts, 1989.

[27] Goodwin, G. C., M. M Seron and J. A. De Doná, *Constrained control and estimation: An optimisation approach*. Springer: Communications and Control Engineering Serie, London, 2005.

[28] Holmström, K.. The TOMLAB Optimization Environment in Matlab. Advanced Modeling and Optimization, **1**, 47-69, 1999.

[29] Huang, S., K. K. Tan and T. H. Lee, *Applied predictive control*. Springer: Advances in Industrial Control Serie, London, 2002.

[30] Hutauruk, N.B.C. and M. Brown. Directed multi-objective optimization for controller design. Proc. *International Conference on Instrumentation, Communic. and Information Tech.*, Bandung, Indonesia, August 3-5, 751-756, 2005.

[31] Isermann R., Digital Control Systems (2nd Edn.). Spring-Verlag, 1991.

[32] Joos, H-D. J. Bals, K. Schnepper, A. Varga, A multiobjective optimization-based software environment for control system design. Proceedings of the *IEEE International Conference on Control Applications,* 1242-1247, Glasgow, 2002.

[33] Johnson, M. A. and M. H. Moradi (Eds.), PID control: New identification and design methods. Springer Verlag, London, 2005.

[34] Ju, J., M.-S. Chiu and C. Tien, "Multiple-objective based model predictive control of pulse jet fabric filters, IChemE Journal Chemical Engineering Research and Design, **78**, 581-589, 2000.

[35] Kawabe, T. and T. Tagami. A new genetic algorithm using Pareto partitioning method for robust partial model matching PID design with two degrees of freedom. Proceedings of *the 3rd International Computer Science Conventions Symposia on Intelligent Industrial Automation* (IIA'99) *and Soft Computing* (SOCO'99), 562-567, Genova, 1999.

[36] Kerrigan E. and J. Maciejowski, Designing model predictive controllers with prioritised constraints and objectives. Proceedings of the *IEEE Symposium on Computer Aided Control System Design*, 33-38, September, 2002.

[37] Kookos, I. K., K.G. Arvanitis, G. Kalogeropoulos, *PI controller tuning via multiobjective optimization*. Proceedings of *the 7th Mediterranean Conference on Control and Automation*, Haifa, Israel, - , 1999.

[38] Kunkle, D., A summary and comparison of MOEA algorithms. Internal Report, College of Computer and Information Science Northeastern University, 2005.

[39] Kwon, W. H. and S. Han, *Receding horizon control: Model predictive control for state models*. Springer: Advanced Textbooks in Control and Signal Processing Serie, London, 2005.

[40] Lin, J. G., *Multi-Objective Problems: Pareto-optimal solutions by method of proper equality constraints*. *IEEE Trans. on Aut. Con.* **AC-21**, 641-650, 1976.

[41] Liu, G. P., J. B. Yang and J.F. Whidborne. *Multiobjective Optimisation and Control.* Research Studies Press Ltd., Exeter, 2003.

[42] Maciejowski, J. M., *Predictive control with constraints***.** Prentice Hall, 2002.

[43] Marler, R.T. and J.S. Arora, Survey of multi-objective optimization methods for engineering. Struct Multidisc Optim, **26**, 369-395, 2004.

[44] Miettinen, K. M., Nonlinear Multiobjective Optimization. *Kluwer Pub.,* 1999.

[45] Miettinen, K. M., Links connected to optimization, operations research and other interesting areas. *http://www.mit.jyu.fi/miettine/lista.html*, 2006.

[46] Nuñez-Reyes, A., C. Scheffer-Dutra, C. Bordons, Comparison of different predictive controllers with multi-objective optimization. Application to an olive oil mill. Proceedings of the *IEEE International Conference on Control Applications,* 1242-1247, Glasgow, 2002.

[47] Ocampo, C., A., *Model predictive control of complex systems including fault tolerance capacities: Application to sewer networks*. Ph.D. Thesis, Technical University of Catalonia, Catalonia, Spain, 2007.

[48] O'Dwyer, A., *Handbook of PI and PID controller tuning rules*. Imperial College Press, London, 2003.

[49] Pareto, V., Manuale di Economia Politica, Societa Editrice Libraria, Milano, Italy, 1906. Translated into English by A.S. Schwier as Manual of Political Economy, Macmillan, New York, 1971.

[50] Pedersen, G. K. M. and Z. Yang, Multi-objective PID-controller tuning for a magnetic levitation system using NSGA-II. Proc. of the Genetic and Evolutionary Computation Conference, 1737-1744, Seattle, July 8-12, 2006.

[51] *Pohlheim, H. Evolutionary Algorithms - Methods, Operators, Practical Examples* (in German). Berlin, Heidelberg, New York: Springer-Verlag, 1999.

[52] Popov, A., A. Farag, H. Werner, Tuning of a PID controller using a multi-objective optimization technique applied to a neutralization plant. *Proc. of the 45th IEEE Conference on Decision and Control*, Seville, 7139-7143, 2005.

[53] Popov, A., Genetic Algorithms for Optimization, User Manual for the GAmin toolbox for Genetic Algorithms Optimization for MATLAB. Only on-line at http://www.geocities.com/potopi/downloads, 2005.

[54] Radke, F., and R. Isermann, A parameter-adaptive PID controller with stepwise parameter optimization, Automatica, 449-451, 1987.

[55] Rossiter, J. A., *Model-based predictive control: A practical approach*. Crc Press: Control Series, Boca Raton, 2003.

[56] Schaffer, J. D., Multipleobjective optimization with vector evaluated genetic algorithms. *Proceedings of the 1st International Conference on Genetic Algorithms and their App.*, Hillsdale: Lawrence Erl-baum Associates, 93-100, 1985.

[57] Seshadri, A., NSGA-II: A multi-objective optimization algorithm. *Matlab Central File Exchange*, http://www.mathworks.com/matlabcentral/, 2006.

[58] Silva, G. J., A. Datta and S. P. Bhattacharyya, *PID controllers for time delay systems*. Birkhäuser, Cambridge, MA, 2003.

[59] Tagami, T., T. Kawabe and K. Ikeda, Multi-objective design scheme for robust I-PD controller. P*roc. of IASTED International Conference on Modelling, Identification, and Control*, Grindelwald, Switzerland, 128-131, 2004.

[60] Tan, K. C., E. F. Khor and T. H. Lee, Multiobjective evolutionary algorithms and applications. *Springer, Advanced Information and Knowledge Proc.*, 2005.

[61] Thomson, W., Cooperative models of bargaining. *Handbook of Game Theory with Economic App.*, in: R.J. Aumann & S. Hart (ed.), **2**, 1237-1284, Elsevier.

[62] Visioli, A., Practical PID Control. Springer, London, 2006.

[63] Wellenreuther, A., A. Gambier and E. Badreddin. Parameter tuning of multiple controllers for multiloop systems using game theory. Preceedings of *the 6th Asian Control Conference*, 773-779**,** Bali, 18-21 July, 2006.

[64] Wellenreuther, A., A. Gambier and E. Badreddin, Optimal multi-loop control design of a reverse osmosis desalination plant with robust performance. *17th IFAC World Congress* (IFAC WC 2008), Seoul, July 6-11, 2008.

[65] Zambrano, D., and E.F Camacho, Application of MPC with multiple objective for a solar refrigeration plant. Preceedings of *the IEEE Conference on Control Applications*, 1230-1235, Glasgow. Septiembre. 2002.

[66] Zitzler, E., K. Deb and L.Thiele, Comparison of multiobjective evolutionary algorithms: Empirical results. Evolutionary Comp. Journal, **8**, 125–148, 2000.

[67] Zitzler E., M. Laumanns and L. Thiele. SPEA2: Improving the Strength Pareto evolutionary algorithm. Tech. Report 103, CH-892 Zurich, Switzerland, 2001.