# MULTICRITERIA CONTROL SYSTEM DESIGN USING AN INTELLIGENT EVOLUTION STRATEGY WITH DYNAMICAL CONSTRAINTS BOUNDARIES

**To Thanh Binh, Ulrich Korn**,

*Institute of Automation, University of Magdeburg, Germany*
*E-mail: binh@infaut.et.uni-magdeburg.de*
*E-mail: korn@infaut.et.uni-magdeburg.de*

**Abstract.** This paper represents a multivariable controller design methodology which is based on the solution of the multiobjective optimization problem using a multiobjective evolution strategy with dynamical constraints boundaries. The evolution strategy allows to handle simultaneously all the types of multiple design objectives and constraints (hard and soft), and to achieve the good approximation of the complete trade–off set between the design objectives while satisfying all the given constraints. This evolution strategy is implemented in the MATLAB–based * environment and it can be used to solve many optimization problems (Binh *et al.*, 1996).

* MATLAB is the Trademark of the MathWorks, Inc.

**Keywords.** Mutation, Reproduction, Selection, Pareto–optimal Set and Dynamical Constraints Boundaries.

## 1. MOTIVATION

The design of many multivariable control systems is essentially a multiobjective design problem, in that the design objectives reflect the need to achieve the economic performance of the system, specifications on the product quality and equipment safety, etc. These practical requirements are normally converted into mathematical expressions, stated either as *objectives* or *constraints* (Boyd and Barratt, 1991). The objectives – the functions of the variables to be optimized, for example, settling times and rise times of the control outputs, zero steady–state tracking errors, requirements on the pole assigment in specified regions, explicit closed–loop performance and robustness by the analytical optimization methods as $\mathcal{L}QG$, $\mathcal{H}_\infty$ mixed–sensitivity and $\mu$–synthesis, need to be simultaneously achieved (normally minimized). Constraints, on the other hand, can often be described either as the hard requirements, which must be satisfied before the optimization of the remaining, or the soft one, which should be maintained within specific bounds and their temporary violations are allowed in order to satisfy the other objectives. In control systems, the closed–loop system stability is an example of the hard constraints, because most performance measures are not defined for unstable systems. In the most cases, the soft constraints can be expressed in terms of function inequalities, for example, upper or lower bounds on the output or the manipulated variables.

If we describe quantitatively all these objectives as a set of $N$ design objective functions $f_i(\underline{x}) \quad \forall i = \overline{1, N}$, where $\underline{x}$ denotes the $n$-dimensional vector of design parameters, the design problem could be formulated as a *multiobjective optimization* problem:

$$\min_{\underline{x}} \underline{f}(\underline{x}) = \min_{\underline{x}} \ (f_1(\underline{x}), f_2(\underline{x}), \cdots, f_N(\underline{x})).$$

Here, the objective variable $\underline{x}$ must be in an universe $\Omega$ of the $n$-dimensional space $\mathbb{R}^n$, which is determined by the soft and hard design constraints, that means,

$$\Omega = \Omega_{(\text{hard})} \bigcap \Omega_{(\text{soft})},$$

where $\Omega_{(\text{hard})}$ is the feasible region for the hard constraints and can be expressed in terms of any functions of the objective variable $\underline{x}$; $\Omega_{(\text{soft})}$ is the feasible region for the soft constraints, i. e.

$$\Omega_{(\text{soft})} = \{\underline{x} \in \mathbb{R}^n : \ c_i^{(\text{lower})} \leq c_i(\underline{x}) \leq c_i^{(\text{upper})}, \forall i = \overline{1, L}\}.$$

The scalar value $L$ is the number of the soft constraints.

By the solution of the multiobjective optimization with constraints the following problems can often be met:

- The design objectives are in conflict, so that it is not possible to improve any of the objectives without deteriorating at least one of the other objectives. This is known as the *concept of pareto optimality* (Goldberg, 1989). Using this concept, the solution of the multiobjective optimization can be seen as the set of pareto–optimal solutions – the set of nondominated or noninferior vectors in the objective function space. The first problem is *how to get the pareto–optimal set* for each multiobjective optimization problem. It can be solved by using the evolution strategy with the given feasible starting point for the optimization (Binh, 1994), (Kahlert, 1991), (Binh and Korn, 1996*a*).
- The second problem is concerned with the *choosing a starting point* for the optimization that satisfies all the given constraints (that means a point of an unknown universe $\Omega$). In the most cases the soft constraints are so strictly that a starting point satisfying *only* the hard constraints can be chosen. The optimization with the well–known evolution strategy can not be started until the feasible starting point (it is in the feasible region $\Omega$) is found.

Both these design situations motivate the development of the new multivariable controller design methodology using the multiobjective evolution strategy with the dynamical constraints boundaries. Instead of the very long searching for the feasible starting point, the optimization should be executed in the universe $\Omega_{(\text{hard})}$ in two following steps:

- Search for the feasible region – the subspace of $\Omega_{(\text{hard})}$.
- When the desired feasible region is reached, the optimization begins to search for the set of pareto–optimal solutions in $\Omega$.

This evolution strategy is introduced in the next section, and based on the main genetic mechanisms: *Mutation, Reproduction and Selection*, and on the ranking according to the *actual concept of pareto optimality*. This evolution strategy guarantees the approximation of the pareto–optimal solutions by the population in the further next generations better than in the current generation. Some examples to illustrate the efficiency of the evolution strategy are shown in section 3.

## 2. THE MAIN ALGORITHM OF THE EVOLUTION STRATEGY

### 2.1 *Representation of the individual*

The major difference between this evolution strategy and the well–known evolution strategy in (Binh, 1994), (Binh and Korn, 1996*a*), (Binh and Korn, 1996*b*) lies in the representation of an individual.

Here, an individual includes four properties as below:

- the so-called *life environment* $\underline{w}$,
- the objective variable $\underline{x} = (x_1, x_2, \cdots, x_n)$,
- the strategy parameter vector $\underline{s} = (s_1, s_2, \cdots, s_n)$
- the corresponding objective function vector $\underline{f}(\underline{x})$,

that means:

$$\mathcal{I}nd = (\underline{w}, \underline{x}, \underline{s}, \underline{f}).$$

The strategy parameter vector $\underline{s}$ has the same dimension as the objective vector $\underline{x}$ and describes the "*personal experiences* " of each individual by the reproducing its offsprings. It is interesting that at the begin of the evolution all individuals have very little experiences (i. e. all elements of the vector $\underline{s}$ are small),but in every generation, the better experiences of the individuals can be accumulated and improved (i. e. all elements of the vector $\underline{s}$ are enlarged). When the global minimum or the pareto–optimal set is found, the vector $\underline{s}$ must become small to guarantee a concentration with the high density of the current population on it.

The *life environment* is the new property of the individuals. The addition of this property into an individual comes from the following motivation. The subspace $\Omega_{(\text{soft})}$ can be seem the ideal life environment for all individuals. But in the few first generations the individuals can live in the real current life environment that is more different to the ideal life environment. Therefore, the individuals usually have to find out the way to improve their life environment until the ideal life environment will be reached. This property is used:

- to make the decision if the individual is viable,
- to create the life environment for the current population,
- and to rank the current population.

Because only the soft constraints, expressed in terms of the inequalities, are mentioned here, the subspace $\Omega_{(\mathrm{soft})}$ can be seen like a hyperparallelogram in the space of the soft constraints. By this way, the current life condition of an individual is described also as a hyperparallelogram defined by the current soft–constraints boundaries. The current life condition of an individual is said to be *ideal* if the given soft–constraints are reached by the individual, that means $\Omega_{(\mathrm{soft})}$ includes its hyperparallelogram. Mathematically, the informations about the position of the hyperparallelogram for an individual can be saved in the vector $\underline{w}$ consisting of the current lower and corresponding upper values of the soft–constraints, also:

$$\underline{w} = (c_1^{(\mathrm{lower})}, \cdots, c_L^{(\mathrm{lower})}, c_1^{(\mathrm{upper})}, \cdots, c_L^{(\mathrm{upper})}).$$

Then, the *life environment* of the current population can be characterized as a hyperparallelogram, so named $\Omega_{(\mathrm{soft})}^{(\mathrm{current})}$, including all hyperparallelograms for every individual and $\Omega_{(\mathrm{soft})}$.

Using this representation and the ranking algorithm (see section 2.2), the life environment of the current population will be improved in every next generation by increasingly diminishing the current hyperparallelogram into $\Omega_{(\mathrm{soft})}$. In other words, we have the optimization process with the *moving* or *dynamical* boundaries of constraints (namely soft–constraints).

### 2.2 *Ranking the individuals*

In the evolution strategy it usually needs to choose the better individual from two (by the mutation and the reproduction) or from many individuals (by the selection). Without loss of generality, we consider only the ranking between two individuals in the universe $\Omega_{(\mathrm{hard})}$. It can easily be generalized for the comparision of more individuals and for the universe $\Omega$.

To do it the following selection criterions are mentioned:

*Criterion 1.* An individual is said to be viable iff it satisfies the current soft–constraints $\Omega_{(\mathrm{soft})}^{(\mathrm{current})}$.

*Criterion 2.* When both the viable individuals either do not satisfy or satisfy the given soft–constraints simultaneuosly, the ranking is based on the *actual concept of pareto optimality*, i. e. choosing only the non–inferior individuals. In another case, the better individual is the individual, which satisfies the given soft–constraints.

In other words, the priority by the selection is as follows:

(1) viability,
(2) satisfaction the given soft–constraints,
(3) non–inferiority.

### 2.3 *The main mechanism*

The algorithm for the mutation, the reproduction and the selection are quite same as by the well–known evolution strategy, but the comparision between the individuals is based on the above ranking scheme.

The mutation allows to evolve the own strategy parameters $\underline{s}$ of an individual so that number of its viable offsprings is biggest and they can lie in the down–climbing direction. By this way, the current population runs very quickly into the local minimum.

The reproduction is performed on the strategy parameters as well as on the objective variables using the well–known two–points–crossover recombination.

By the selection the pareto–optimality in the objective function space is used. The algorithm concludes the assigning rank 1 to the non–dominated individuals and rank 2 to the other of the current population. After some generations, the current population reaches a neighbourhood of the set of pareto–optimal solutions and the number of the non–inferior individuals is, therefore, big enough. A new problem with the approximation of a continuum set (namely the set of pareto–optimal solutions) by the finite set (the current population with the finite population size) appears here. Generally, the best solution for this problem is not found. This situation leads to the great unstability of the population in the next generations (that means the population will continueosly be moving on the surface of the pareto–optimal set). From this reason, a global picture of the pareto–optimal solutions can not be achieved in every generation.

To avoid it and to create the uniform distribution of the current population on the trade–offs surface we recommend the following selection algorithm:

*Algorithm 1.* Let

$$\underline{f}^{(\min)} = (\min f_1, \cdots, \min f_N) \;=\; (f_1^{(\min)}, \cdots, f_N^{(\min)})$$
$$\underline{f}^{(\max)} = (\max f_1, \cdots, \max f_N) \;=\; (f_1^{(\max)}, \cdots, f_N^{(\max)}),$$

where the minimum and maximum operators are performed along each coordinate axes of the objective function space for all individuals of the population.
Then, the current trade–offs surface is bound in the hyperparallelogram $\mathcal{H}$ defined by $\underline{f}^{(\min)}$ and $\underline{f}^{(\max)}$.

- Dividing each interval $[f_i^{(\min)}, f_i^{(\max)}]$ into $N_{pop}$ small sections $\delta_i$, i. e.:

$$\delta_i = \frac{f_i^{(\max)} - f_i^{(\min)}}{N_{pop}}.$$

- In the $i$-th coordinate axes of the objective function space, the best individual in each of $\dfrac{N_{\mathrm{pop}}}{N+k}$ the first sections is selected, where $k$ is an integer number.

## 3. SOME DEMONSTRATIONS

In this section, we would like to illustrate the efficiency of the new evolution strategy by the optimization of some mathematical functions. Because their solutions are well–known, then it can be good understood how to handle the multiple objectives and constraints by the evolution strategy. By the application for the multiobjective control system design, to solve the so–called IFAC–93 Benchmark Problem (Whidborne and etc., 1995), we have got the results which are as good as in (Binh and Korn, 1996*a*) and (Binh and Korn, 1996*b*) but better than in (Whidborne and etc., 1995).

### 3.1 *The Multi Modal Function*

The objective function is described as follows:

$$f(x_1,x_2) = \begin{cases} (x_1-3)^2+(x_2-3)^2+8 \\ \quad \text{for } 0 \le x_1 \le 6; 0 \le x_2 \le 6 \\ (x_1-3)^2+(x_2-9)^2+5 \\ \quad \text{for } 0 \le x_1 \le 6; 6 \le x_2 \le 12 \\ (x_1-3)^2+(x_2-15)^2+4 \\ \quad \text{for } 0 \le x_1 \le 6; 12 \le x_2 \le 18 \\ (x_1-9)^2+(x_2-3)^2+7 \\ \quad \text{for } 6 \le x_1 \le 12; 0 \le x_2 \le 6 \\ (x_1-9)^2+(x_2-9)^2+9 \\ \quad \text{for } 6 \le x_1 \le 12; 6 \le x_2 \le 12 \\ (x_1-9)^2+(x_2-15)^2+3 \\ \quad \text{for } 6 \le x_1 \le 12; 12 \le x_2 \le 18 \\ (x_1-15)^2+(x_2-3)^2+1 \\ \quad \text{for } 12 \le x_1 \le 18; 0 \le x_2 \le 6 \\ (x_1-15)^2+(x_2-9)^2+6 \\ \quad \text{for } 12 \le x_1 \le 18; 6 \le x_2 \le 12 \\ (x_1-15)^2+(x_2-15)^2+2 \\ \quad \text{for } 12 \le x_1 \le 18; 12 \le x_2 \le 18 \end{cases}$$

with the hard constraints:

$$0 \le x_1 \le 18 \text{ and } 0 \le x_2 \le 18$$

and the soft constraints:

$$0 \le x_1 \le 6 \text{ and } 12 \le x_2 \le 18.$$

This optimization problem has the following special properties:

- the objective function has 8 local minima and the global minimum at the point $\underline{x} = (15,3)$.
- not the point $(15,3)$ with the smallest value $(f=1)$ of the objective function but the point $\underline{x} = (3,15)$ with the objective function value $f = 4$ must be the global solution of the above optimization. In other words, the global minimum of the optimization without the soft–constraints does not identify to the one with the above soft–constraints.

Starting at the point $\underline{x} = (15,1)$ lying outside and far from the feasible region:

$$\{\underline{x} \in \mathbb{R}^2 :\ 0 \le x_1 \le 6 \text{ and } 12 \le x_2 \le 18\},$$

the current population should know that the point $\underline{x} = (15,3)$ is only a temporary goal until at least an (feasible) individual reachs the feasible region. The starting population does not satisfy the given soft–constraints, i. e. it is not viable. To guarantee that it can be viable, the current soft–constraints have to be enlarged to the region $\mathcal{H}_0$:

$$\{\underline{x} \in \mathbb{R}^2 :\ 0 \le x_1 \le 18 \text{ and } 0 \le x_2 \le 18\}.$$

The evolution strategy shows that some feasible individuals can be found after 8 generations and the correspondent soft–constraints region $\mathcal{H}_8$:

$$\{\underline{x} \in \mathbb{R}^2 :\ 0 \le x_1 \le 16 \text{ and } 2 \le x_2 \le 18\}$$

is smaller than $\mathcal{H}_0$. In the 20-th generation the feasible region is found. The search for the desired global minimum of the optimization problem begins at the 21-th generation, and successfully ends in the 36-th generation (see the Figure 1).

### 3.2 *The bi–objective optimization*

We consider the following bi–objective optimization with:

$$f_1(x_1,x_2) = x_1^2 + x_2^2;$$
$$f_2(x_1,x_2) = (x_1-5)^2 + (x_2-5)^2;$$

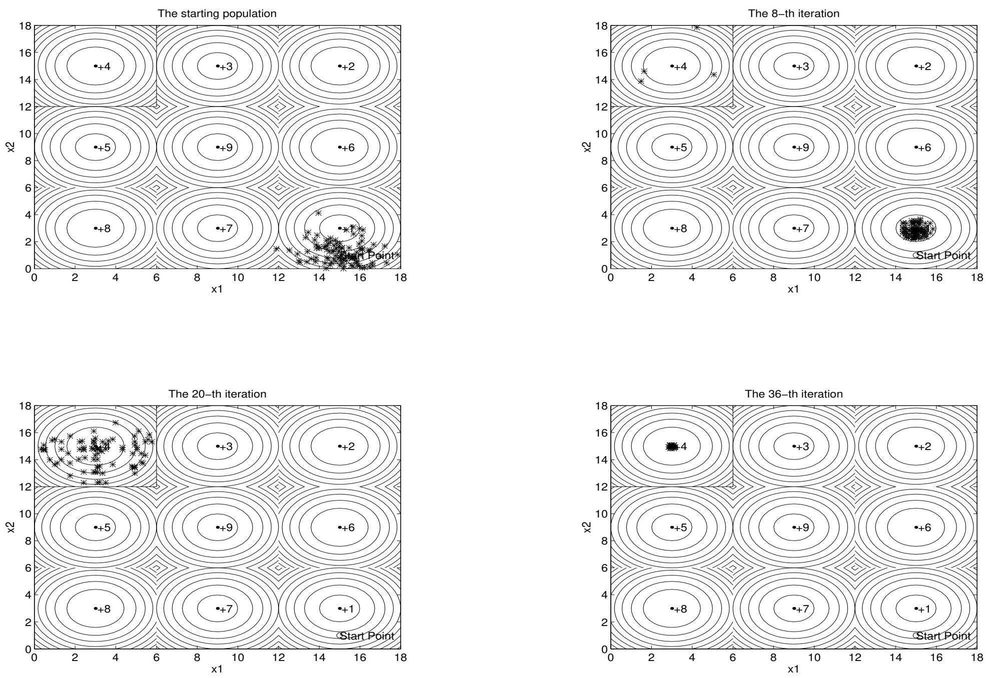


Fig. 1. Optimization process for the multi modal function

with the hard constraints:

$$0 \leq f_1 \leq 1100 \text{ and } 0 \leq f_2 \leq 1000$$

and the soft constraints:

$$-5 \leq x_1 \leq 10 \text{ and } -5 \leq x_2 \leq 10$$

The optimization proccess with the starting point at $\underline{x} = (-10, 30)$ is shown in the Figure 2. It is clear, every individual is good able to learn how to get the better solution in each optimization situation by adaptiv changing its own strategy parameter. The feasible region can, therefore, very quickly be found.

## 4. CONCLUSION

The new evolution strategy allows the users:

- to overcome the missing a priori knowledge about the structure of the feasible region that is very often met in the system control design. Independent from the starting point, the evolution strategy tries to search the better life conditions for every population and gets the ideal life condition after some generations. That means the current individuals and population are able to adapt themselves to various optimization situations.
- to get the global survey about the structure of the pareto–optimal set.
- to choose easily the best solution from it by using a lot of the dialogue methods in the MATLAB-based environment (Binh *et al.*, 1996).

By the solution of a lot of optimization problems: both the scalar and the multi–objective, it is well known that the evolution strategy possess essentially many good properties: high robustness to get a feasible region and to find the global minimum or the set of pareto–optimal solutions (Binh *et al.*, 1996).

## 5. REFERENCES

Binh, T.T. (1994). Eine Entwurfsstrategie für Mehrgrößensysteme zur Polgebietsvorgabe. PhD thesis. Institute of Automation, University of Magdeburg, Germany.

Binh, T.T. and U. Korn (1996*a*). An evolution strategy for the multiobjective optimization. To appear in Mendel96 Conference, in Brno , Juni 1996.

Binh, T.T. and U. Korn (1996*b*). Robuster Reglerentwurf für das IFAC93 Benchmark Problem mittels einer globalen Mehr-Ziel-Evolutionsstrategie. To appear in *Automatisierungstechnik*.

Binh, T.T., U. Korn and J. Kliche (1996). *Evolution Strategy Toolbox for use with MATLAB*. Techni-

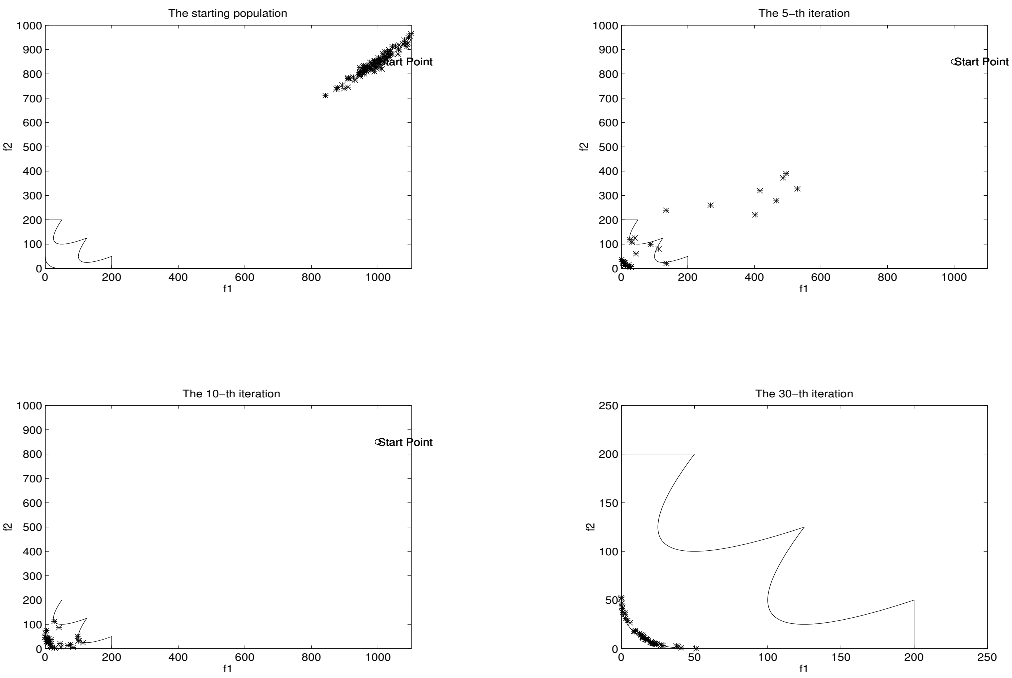


Fig. 2. Bi–objective optimization process

cal report. Institute of Automation, University of Magdeburg, Germany.

Boyd, S.P. and C.H. Barratt (1991). *Linear Controller Design: Limits of Performance*. Prentice Hall, Englewood Cliffs, New Jersey 07632. Stanford University.

Goldberg, D.E. (1989). *Genetic algorithms in Search, Optimization, and Machine Learning*. 1. ed.. Addision–Wesley Publishing Company, Inc.. New York, England, Bonn, Tokyo.

Kahlert, J. (1991). *Vektorielle Optimierung mit Evolutionsstrategien und Anwendung in der Regelungstechnik*. Forschungsbericht VDI Reihe 8 Nr.234.

Whidborne, J.F. and etc. (1995). Robust control of an unknown plant – the IFAC–93 benchmark. *Int. J. Control* **61**(3), 589–640.