The use of an evolutionary algorithm for the parameter selection of predictive controllers

R.C. Gutiérrez-Urquídez, G. Valencia-Palomo and O.M. Rodríguez-Elias

Instituto Tecnológico de Hermosillo, Av. Tecnológico S/N, Hermosillo, Mexico.
ro_gutierrez@ith.mx, gvalencia@ith.mx, omrodriguez@ith.mx

Paper received on 11/19/13, Accepted on 01/19/14.

Abstract. In the design of linear predictive controllers (MPC), a problem that has not yet been fully resolved, is how to determine the best strategy for the selection of the tuning parameters in order to obtain good performance and a good feasibility region while maintaining a sensible low computational burden for practical implementation. The main contribution of this paper is to achieve a systematic tuning by use of a multi-objective evolutionary algorithm (MOEA) on predictive control algorithm that has been reparameterized with Laguerre functions. Numerical simulations show that MOEA is a useful tool to obtain consistently good solutions for the selection of MPC tuning parameters.

1 Introduction

Although predictive control (MPC) has its background well established [1, 2], and is widely used, there are still some theoretical and practical problems to be solved. For instance, one key conflict is between feasibility and performance: if a MPC controller is well tuned to provide high performance, feasibility will have (in general) a very small region, unless one use a large number of decision variables (degrees of freedom or d.o.f.), which implies increasing the computational load of the algorithm. On the other hand, if one aims to get better feasibility with a fixed d.o.f., the result will be a detuned controller with relatively poor performance. Several authors have looked at this problem and have proposed different strategies that provide improved quality of predictions and maintain a balance between performance and computational complexity. The first MPC algorithms, such as Dynamic Matrix Control (DMC) [3] or Generalised Predictive Control(GPC) [4], will usually give reasonable performance, but only for large input and output horizons over the settling time. Furthermore, this may not be so effective when the open-loop dynamics of the system is poor or simply if there are state or output constraints. Another drawback is that they do not automatically guarantee stability, thus requiring further tuning considerations [5].

Concern for stability has been a major engine for generating different formulations of MPC. At 1990s, proposals arise to amend this problem of optimal open-loop control, so that closed-loop stability can be guaranteed. The most accepted approach for an a priori recursive feasibility, is the dual-mode MPC paradigm [5, 6], for which, the predictions have two modes: (i) a transient phase containing the d.o.f. and (ii) a terminal mode with guaranteed convergence. Following the dual-mode approach, other work has looked at alternative ways of formulating the d.o.f. for optimisation, for instance by interpolation methods [7]. However, these methods do not currently extend well to large dimensional systems. Another interesting work considered the so-called triple mode strategies [8], where one embeds a smooth transition between a controller with good feasibility and other with good performance into a single model and use the decision variables to improve performance/feasibility further. This work is successful but relies on heavy computation and algebra and thus may be difficult for industrial implementations.

Recently, several works have been proposed for more efficient predictive control algorithms by reparametrization of the d.o.f. of the optimization problem. An alternative is to use orthogonal functions either with Laguerre/Kautz polynomials or through Generalised orthonormal functions [9–11], since they have proven to be very effective for improving the volume of the feasible region with a limited number of d.o.f. with almost no performance loss. And finally, it is also possible to reparameterize the d.o.f. using directional information for the specific control problem [12].

In summary, the recent MPC algorithms differ in the way of reparameterizing the d.o.f., but introducing one or more tuning parameters that has to be selected by the user. The original papers [9–12] fail to give guidelines in this selection, since in most cases, the selection of tuning parameters is still performed based on trial and error simulations. Therefore, the main debate is to establish the best strategy for the selection of controller tuning parameters that allow to efficiently handle the trade-off between feasibility, computational burden and controller performance. This paper provides an alternative that uses multi-objective evolutionary algorithms, in order to achieve a systematic tuning of a MPC. Specifically, the focus of this work is the tuning of a MPC algorithm whose d.o.f.s have been reparameterized using Laguerre functions. This paper is organized as follows: Sections 2 and 3 will give the necessary background about modelling, predictive control, Laguerre optimal predictive control (LOMPC) and multi-objective evolutionary algorithms (MOEAs). Section 4 presents and develops the proposed tuning algorithm for LOMPC with MOEAs. Section 5 gives a numerical example showing the efficacy of the proposed algorithm. Finally, the conclusions are presented in Section 6.

2 Model predictive control and Laguerre functions

2.1 Optimal predictive control (OMPC)

Model predictive control (MPC) has had a peculiar evolution; it was initially developed in industry, at 70's, and later was taken by academic sector. Early predictive controllers were based in heuristic algorithms until the research community established their theoretical support. Predictive Control, also known as *receding control horizon* refers to a set of control algorithms and techniques which lead to the design of controllers with a similar structure; with information based on past inputs and outputs of the plant, and making use of the process model, an optimal control input is obtained by minimizing an objective function (cost function) in a time interval named control horizon. The common elements in MPC controllers are:

- **The process model.** The mathematical model is used to generate system predictions. In particular, this model must show the dependence of the output on the current measured variable and the current/future inputs. A discrete-time state-space model is assumed, which has the following form:

$$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k; \quad \mathbf{y}_k = \mathbf{C}\mathbf{x}_k + \mathbf{D}\mathbf{u}_k; \tag{1}$$

  with $\mathbf{x}_k \in \mathbb{R}^n$, $\mathbf{y}_k \in \mathbb{R}^l$, $\mathbf{u}_k \in \mathbb{R}^m$ which are the state vectors, the measured output and the plant input respectively. This work also adopts an independent model approach with optimal feedback $\mathbf{K}$. Let $\mathbf{w}_k$ the output of the independent model, hence, the estimated disturbance is $\hat{\mathbf{d}}_k = \mathbf{y}_k - \mathbf{w}_k$ [3]. Disturbance rejection and offset free tracking will be achieved using the offset form of state feedback [13, 14], that is:

$$\mathbf{u}_k - \mathbf{u}_{ss} = -\mathbf{K}(\mathbf{x}_k - \mathbf{x}_{ss}), \tag{2}$$

  where $\mathbf{x}$ is the state of the independent model and $\mathbf{x}_{ss}$, are estimated values of the stady-states giving no offset; these depend upon the model parameters and the disturbance estimate [14].
- **The performance index** (objective function or cost function). This is used to quantify the deviation of the measured output with respect to the desired output and the control effort. A way of defining the cost function is by using deviation variables to penalize the deviation of absolute input on the expected steady state input:

$$\mathbf{J}_k = \sum_{i=0}^{n_y} (\mathbf{x}_{k+i} - \mathbf{x}_{ss})^T \mathbf{Q} (\mathbf{x}_{k+i} - \mathbf{x}_{ss}) + \sum_{i=0}^{n_{u-1}} (\mathbf{u}_{k+i} - \mathbf{u}_{ss})^T \mathbf{R} (\mathbf{u}_{k+i} - \mathbf{u}_{ss}), \tag{3}$$

  with $\mathbf{Q}$ and $\mathbf{R}$ positive definite state and input cost weighting matrices, n_y is the prediction horizon and n_u, the control horizon. Because in practice all real processes are subject to constraints, these need to be considered in the cost function. One of the major selling points of MPC is its ability to do online constraint handling in a systematic fashion, retaining to some extent the stability margins and performance of the unconstrained law. Let the system be subject to constraints of the form:

$$\left.\begin{array}{c} \mathbf{u}_{min} \leq \mathbf{u}_k \leq \mathbf{u}_{máx}; \\ \Delta\mathbf{u}_{min} \leq \mathbf{u}_{k+1} - \mathbf{u}_k \leq \Delta\mathbf{u}_{máx}; \\ \mathbf{y}_{min} \leq \mathbf{y}_k \leq \mathbf{y}_{máx}. \end{array}\right\} \forall k. \tag{4}$$

- **The optimization algorithm** determines the optimal control input that minimizes the cost function imposed by a set of constraints. Minimizing the cost function subject to both current and future constraints and obtaining control action, the optimization results in a quadratic program (QP).
- **The receding horizon.** Although the optimal trajectory of future control signal is completely described within the moving horizon window, the actual control input to the plant only takes the first sample of the control signal, $\mathbf{u}_k$, while neglecting the rest of the trajectory [15]. The horizon selected for predictions should include all significant dynamics; otherwise performance may be poor and important events may be unobserved.

The key idea of Optimal MPC (OMPC) [5] is to embed into the predictions the unconstrained optimal behaviour and handle constraints by using perturbations about this. The closed-loop paradigm uses perturbations as d.o.f. for the optimal control law without constraints. Disturbance rejection and offset free tracking will be achieved using the offset form of state feedback of (2) [2]. For convenience, d.o.f. can be reformulated in terms of a new variable $\mathbf{c}_k$. Hence, assuming $\mathbf{K}$ is the optimal feedback, the input predictions are defined as follows:

$$\mathbf{u}_{k+i} - \mathbf{u}_{ss} = \begin{cases} -\mathbf{K}(\mathbf{x}_{k+i} - \mathbf{x}_{ss}) + \mathbf{c}_{k+i}; & i \in \{1, 2, ...n_c - 1\} \\ -\mathbf{K}(\mathbf{x}_{k+i} - \mathbf{x}_{ss}); & i \in \{n_c, n_c + 1, ...\} \end{cases}, \qquad (5)$$

where the perturbations $\mathbf{c}_k$ are the d.o.f. for optimization; conveniently sumarised in vector: $\underset{\rightarrow}{\mathbf{c}}_k = [\mathbf{c}_k^T, \mathbf{c}_{k+1}^T, ..., \mathbf{c}_{k+n_c-1}^T]^T$. Input predictions (5) and state associated (1) which satisfies constraints (4): $\mathbf{M}\mathbf{x}_k + \mathbf{N}\underset{\rightarrow}{\mathbf{c}}_k \leq f(k)$. In practice, if an unconstrained optimal prediction may violate a constraint defined in (4), prediction class more suitable shall be used according (5). The OMPC algorithm can be summarized [2]:

$$\underset{\rightarrow}{\mathbf{c}}_k^* = \arg\min_{\underset{\rightarrow}{\mathbf{c}}_k} \underset{\rightarrow}{\mathbf{c}}_k^T \mathbf{W}_j \underset{\rightarrow}{\mathbf{c}}_k \quad s.a.\ \mathbf{M}_j \mathbf{x}_k + \mathbf{N}_j \underset{\rightarrow}{\mathbf{c}}_k \leq f(k). \qquad (6)$$

Use $\underset{\rightarrow}{\mathbf{c}}_k^*$ to construct the input (5).

OMPC algorithm has implied LQR theory and is able to find a global optimum on the objective function. If one chooses a value for $\mathbf{K}$ in (5) to become a optimal Linear-Quadratic-Regulator(LQR)[5], the feasible region depends only on the class of prediction and hence also the number of free movements, that is, n_c.

Remark 21 *The optimization of* (6) *can require a large n_c (d.o.f.) to obtain both good performance and a large feasible region.*

Definition 21 *Maximum Admissible Set (MAS). A common method to achieve recursive feasibility is to find the region of the state space where positively invariant sets ensure the action of an unconstraint control law but satisfy all constraints in the future. This achieved using the dual-mode paradigm. And the greatest invariant set possible for use as the terminal state set is referred as Maximum Admissible Set (MAS) [6, 2]. For a linear discrete system, observable, pre-stabilized by a gain $\mathbf{K}$ of state feedback, associated with a set of constraints* (4)*, there exists a set, MAS, finite and where the constraints are satisfied for all future time intervals: $MAS = \{\mathbf{x}_k \in \mathbb{R}^n \mid \mathbf{M}\mathbf{x}_k \leq \mathbf{d}\}$.*

Definition 22 *Maximal Controllable Admissible Set ($MCAS$). It is also posible to define a region in $\mathbf{x}$ in which it is possible to find a $\mathbf{c}_k$ such that at the future trajectory satisfying the constraints: $MCAS = \{\mathbf{x}_k \in \mathbb{R}^n \exists \underset{\rightarrow}{\mathbf{c}}_k \in \mathbb{R}^{n_c m} \quad s.t. \quad \mathbf{M}\mathbf{x}_k + \mathbf{N}\underset{\rightarrow}{\mathbf{c}}_k \leq \mathbf{d}\}$; and this is named Maximal Controllable Admissible Set ($MCAS$).*

2.2 LOMPC: Laguerre polynomials and OMPC

The fundamental weakness of OMPC algorithms is that the d.o.f. are parameterized as individual values at specific samples and have an impact over just one sample and thus have a limited impact on feasibility. If the initial state is far away from the MAS associated to $\mathbf{c}_k = 0$, the n_c steps will be insufficient to move into the MAS. Laguerre OMPC (LOMPC) is a dual-mode MPC algorithm where the d.o.f. within the input predictions are parameterized in terms of Laguerre polynomials rather than using the more normal standard basis set. The algorithm proposes to replace common decision variables $\mathbf{u}_k$ and $\mathbf{c}_k$ by Laguerre polynomials L_i in OMPC. It has been shown that with the reparameterization of d.o.f. get an increase in feasibility region (MCAS) of controller LOMPC regarding equal number of d.o.f. in OMPC. The $z-$transform of discrete Laguerre polynomials are defined as follows:

$$\Gamma_i(z) = \sqrt{1-a^2}\frac{(z^{-1}-a)^{n-1}}{(1-az^{-1})^n}; \qquad 0 \le a \le 1. \tag{7}$$

With the inverse $z-$ transform of $\Gamma_n(z,a)$, denoted by $l_{(k,n)}$, the Laguerre functions set are the vector: $\mathbf{L}_k = \{l_{k,1},\, l_{k,2},\, ...,\, l_{k,n}\}^T$. The size of the $\mathbf{A}_L$ matrix, is $n \times n$; and it is a function of the parameters a, $\beta = 1 - a^2$ and initial condition $\mathbf{L}_0$, so that:

$$\mathbf{L}_{k+1} = \underbrace{\begin{bmatrix} a & 0 & 0\,0\cdots \\ \beta & a & 0\,0\cdots \\ -a\beta & \beta & a\,0\cdots \\ a^2\beta & -a\beta & \beta\,a\cdots \\ \vdots & \vdots & \vdots\,\vdots\,\ddots \end{bmatrix}}_{A_L}\mathbf{L}_k, \quad \mathbf{L}_0 = \sqrt{1-a^2}\left[1,\, -a,\, a^2,\, \ldots\right]^T. \tag{8}$$

The basic concept of OMPC is preserved [5, 2], that is the predictions take the form of (5) and optimality dynamics included in the predictions. However, a key difference is that the disturbance (terms $\mathbf{c}_k$) is defined with Laguerre polynomials instead of taking the d.o.f. individually. The relevant link between Laguerre and predicted values of $\underrightarrow{\mathbf{c}}_k$, are summarized in the following equation:

$$\underrightarrow{\mathbf{c}}_k = [\mathbf{c}_k^T, \ldots, \mathbf{c}_{k+n_c-1}^T]^T, \quad \underrightarrow{\mathbf{c}}_k(L) = \underbrace{[\mathbf{L}_0^T, \mathbf{L}_1^T, \ldots]^T}_{H_L}\underrightarrow{\eta}_k = \mathbf{H}_L\underrightarrow{\eta}_k. \tag{9}$$

Now, with $\mathbf{L}_{k+1} = \mathbf{A}_L * \mathbf{L}_k$, the decision variable is $\underrightarrow{\eta}_k$; and substituting predictions in (8), get the LOMPC optimization, which is:

$$\underrightarrow{\eta}_k^* = \arg\min_{\underrightarrow{\eta}_k} \underrightarrow{\eta}_k^T \left\{ \sum_{i=0}^{\infty} \mathbf{A}_L^i \mathbf{L}_0 \mathbf{S} \mathbf{L}_0^T \mathbf{A}_L^{i\,T} \right\} \underrightarrow{\eta}_k = \underrightarrow{\eta}_k^T \mathbf{S}_L \underrightarrow{\eta}_k \tag{10}$$

$$s.t.\ \mathbf{M}\mathbf{x}_k + \mathbf{N}\mathbf{H}_L\underrightarrow{\eta}_k \le \mathbf{d}. \tag{11}$$

3 Multi-Objective Evolutionary Algorithms (MOEA)

Evolutionary algorithms (EA) allows flexible representation of decision variables and performance evaluation; also are robust and methodological tool for search and optimization, with the ability to work in environments that include discontinuities, time variance, bad behavior, multi-modality, uncertainty and noise. The EA are increasingly accepted in the control community, their applications are mainly in the off-line design and online optimization [16].

EA are systematic methods for solving search and optimization problems in which apply the same methods of the biological evolution, based on selection, reproduction and mutation of the population. These algorithms transform a set of individual mathematical objects using operations that are modeled according to the Darwinian principle of reproduction and survival of the fittest [17].

The simplest form of an optimization problem is to consider the existence of a single criterion or objective in the search for a solution. However, when there are multiple objectives to optimize, there is no single solution, but rather a set of compromise solutions together. These problems are called Multi-objective Problem (MOP). Most EA multi-objective optimization techniques use the Pareto concept. Generally, must be found a vector of decision variables, $\overrightarrow{\mathbf{x}}^{*}$, which optimizes a vector of objective functions, $\overrightarrow{\mathbf{f}}(x)$, and satisfies certain inequality constraints, $\mathbf{g}_i(x)$, or equal, $\mathbf{h}_i(x)$ [18, 19]. Both the objective functions and constraints are functions of the decision variables. Several types of MOEAs exist; each uses different mechanisms of selection, crossover and elitism in particular, but for this paper we use the algorithm NSGA-II (Nondominated Sorting Genetic Algorithm II), proposed by [20]. NSGA-II is an improved version of NSGA that modifies the mechanism for diversity preserving and incorporates an explicit mechanism for elitism; leave the use of Sharing distance of the NSGA to use Crowding tournament selection operator as a method of diversity preservation.

4 Tuning LOMPC

In the work previously presented, it is shown that the reparameterization of d.o.f. has advantages over conventional approaches. However, there is still no systematic way to define the controller tuning parameters, this mainly due to the compromise between the objectives to be optimized. It is clear that the problem to be solved is a multi-objective optimization, where the search space is fairly large, which justifies the use of MOEAs. The purpose of this work is to develop a method to systematically choose the optimal values of the tuning parameters, a and n_c, of an MPC whose d.o.f. have been reparameterized with Laguerre functions; in order to guarantee the best trade-off between feasibility, performance and computational load. Therefore, there are two decision variables a and n_c and three optimizing conditions:

1. Maximize the feasibility region, $\max f_{a,n_c}(\vartheta)$.
2. Minimize the performance loss, $\min f_{a,n_c}(\beta)$.
3. Minimize the computational burden, $\min f_{a,n_c}(\varrho)$.

Also the constraints associated with the parameter selection must be added to the multi-objective optimization problem:

$$0 \le a \le 1; \qquad 1 \le n_c \le n_{c,max}; \qquad n_c \ \textit{integer}. \tag{12}$$

4.1 Feasibility evaluation (ϑ)

In order to estimate the normalized volume, first is defined a polytope $\mathbf{P}_{opt}$ as the global MCAS of OMPC with a large number of degrees of freedom, able to represent the largest feasible region that can be obtained by the controller (usually $n_c \geq 20$) [11]: $\mathbf{P}_{opt} = \{(x, c) \mid \mathbf{M}\mathbf{x}_k + \mathbf{N}\underrightarrow{c}_k \leqslant \mathbf{d}\}$. Also it is defined a polytope $\mathbf{P}_{H_L}$ corresponding to proposed parameterization MCAS: $\mathbf{P}_{H_L} = \{(x, \eta) \mid \mathbf{M}_L x + \mathbf{N}_L \underrightarrow{\eta} \leq \mathbf{d}\}$; where $\mathbf{P}_{H_L}$ is the polytope sliced by the parameterization matrix $\mathbf{H}_L$. The volume of $\mathbf{P}_{opt}$ and $\mathbf{P}_{H_L}$ polytopes, represent the feasible regions or feasible volumes for each type of algorithm.

The volume calculation of a high dimensional polytope is a complex task and the computing time for these polytopes can be prohibitive; consequently, this paper approximates the volume by computing the average distance from the origin to the boundary of the associated MCAS (radius). First select a large number of equi-spaced (by solid angle) or random directions in the state space i.e. $x = [x_1, ..., x_n]$ and then, the distance from the origin to the boundary of MCAS is determined by solving a linear programming (LP) for each direction x_i selected. Greater distances imply bigger feasible region. The objective function for evaluating the normalized feasible volume is then:

$$\vartheta = \frac{vol(\mathbf{P}_{H_L})}{vol(\mathbf{P}_{opt})}. \tag{13}$$

4.2 Performance evaluation (β)

The performance evaluation it is done by realizing the calculation for the n-points x_i selected, they are represented by the optimized values of the associated cost function, i.e. $J_{opt}(x_i)$ and $J_{H_L}(x_i)$. To ensure fairness in comparison of these values, scaling is used (setting) in one direction x_i given [11]. The objective function for evaluating the normalized performance is:

$$\beta = \frac{1}{n}\sum_{i=1}^{n} \frac{\mathbf{J}_{H_L}(x_i)}{\mathbf{J}_{opt}(x_i)}. \tag{14}$$

4.3 Computational load evaluation (ϱ)

It is demonstrated that the re-parameterization of d.o.f. proposed in LOMPC algorithm is able to achieve great feasibility regions while maintaining an acceptable local optimality within a relatively low computational complexity compared to conventional OMPC approaches. However, this reduction in d.o.f. not necessarily results in a reduction of the complexity of optimization and therefore the computational load, since the resulting quadratic programming of reassignment is denser (heavier) than for OMPC [11]. So, how one can determine the minimum number of d.o.f., which gives the best performance and the largest feasible region? One alternative is to compare the online computational load for LOMPC and OMPC as a function of the number of floating point operations per second (flops) required for each algorithm. For OMPC, the computational complexity is linear with respect to the horizon length and cubic respect to state and the input dimension, so that:

$$\varrho_{(OMPC)} = n_c + {n_x}^3 + {n_u}^3 \quad (flops). \tag{15}$$

For LOMPC, their computational load is cubic in number of d.o.f., the state and input dimensions [11]:

$$\varrho_{(LOMPC)} = {n_c}^3 + {n_x}^3 + {n_u}^3 \quad (flops). \tag{16}$$

5 Numerical example

Consider the following discrete-time state-space model with constraints:

$$\mathbf{x}_{k+1} = \begin{bmatrix} 0.6 & -0.4 \\ 1.0 & 1.4 \end{bmatrix} \mathbf{x}_k + \begin{bmatrix} 0.20 \\ 0.05 \end{bmatrix} \mathbf{u}_k; \quad \mathbf{y}_k = \begin{bmatrix} 1.0 & -2.0 \end{bmatrix} \mathbf{x}_k;$$

$$-1.5 \le \mathbf{u}_k \le 0.8; \quad |\Delta \mathbf{u}_k| \le 0.4, \quad \mathbf{x}_k \le 5; \quad \mathbf{Q} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}; \quad \mathbf{R} = 2.$$

The objective is to compare OMPC ($n_c = 3$), OMPCopt ($n_c = 20$) with LOMPC tunned using EA. This example utilizes the Matlab Multi-objective Optimization Toolbox that allows the use of a variant of NSGA-II [21]. The EA were run using a search space generated by varying $n_c \in \{2, 3, 4, 5\}$ and $a \in [0, 1]$. The Pareto front obtained is shown in figure 1. Table 1 summarizes the LOMPC tuning parameters obtained with EA and the objective functions evaluated with these. This shows that performance and feasibility improve as n_c is increased, and as expected, the computational load too. However, within the various tuning parameters found with the EA, it can be selected any combination of these and achieve the best trade-off between the above objectives. Table 1 also includes the respective calculated values for OMPC using the same d.o.f. for comparison purposes. Figures 2, 3 and 4 show the feasibility, performance and computational load evaluation for each pair of tuning parameters with EA and indicate clearly where there are located the optimal values on the LOMPC search space.

Table 1. Tuning parameters, a y n_c, obtained with the NSGA-II algorithm in LOMPC

Tuning parameter		Average radius to MCAS, $J\vartheta$		Performance $J\beta$		Computational load (flops) $J\varrho$	
nc	a	OMPC	LOMPC	OMPC	LOMPC	OMPC	LOMP
2	0.6208	0.4522	0.6339	1.0	1.0526	11	17
3	0.7915	0.5503	0.7345	1.0	1.0528	12	36
3	0.7916	0.5503	0.7345	1.0	1.0528	12	36
4	0.6548	0.6265	0.7833	1.0	1.0528	13	73
4	0.6688	0.6265	0.7827	1.0	1.0528	13	73

Using one of the selected tuning options, i.e. $a = 0.7915, n_c = 3$, figure 5 shows the plots of the controller simulation for: (i) an initial state in the MCAS of both, OMPC and

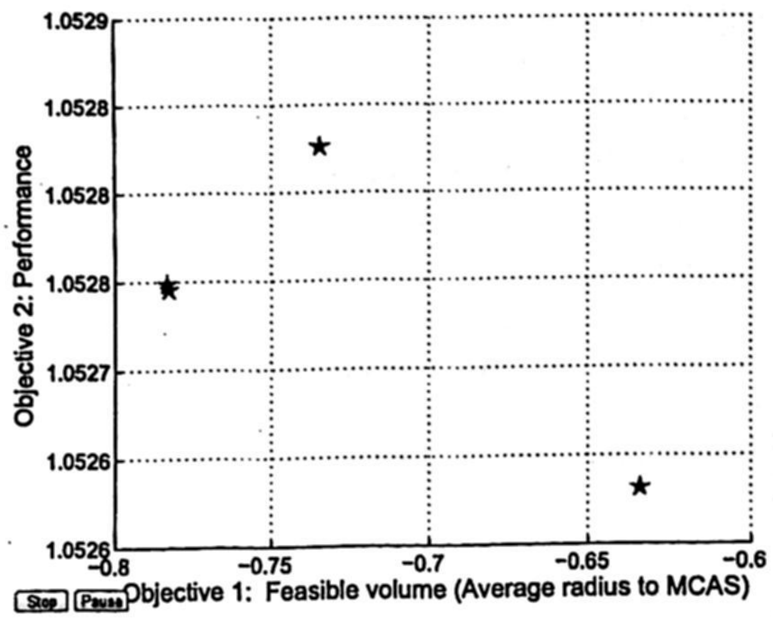

Fig. 1. Pareto front.

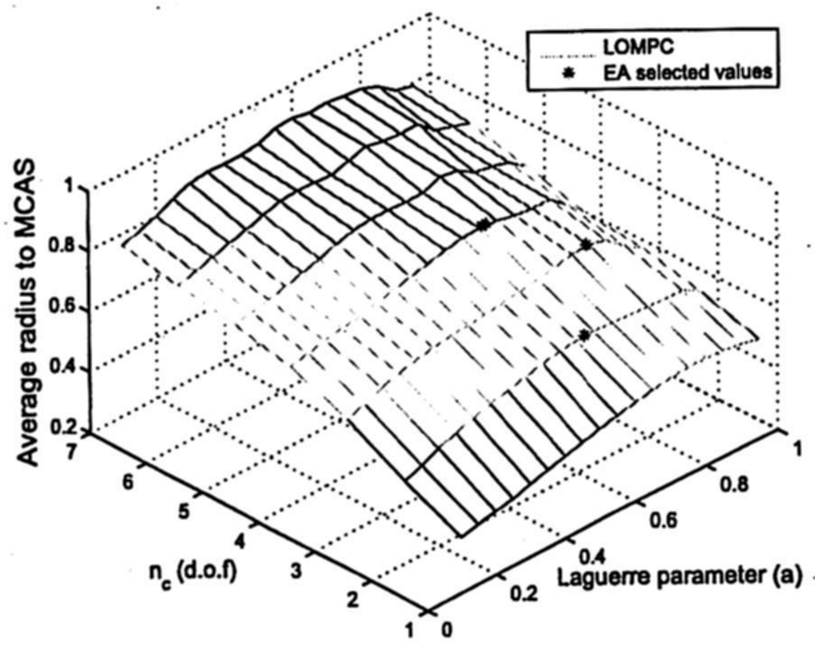

Fig. 2. Feasibility function of average radius to MCAS

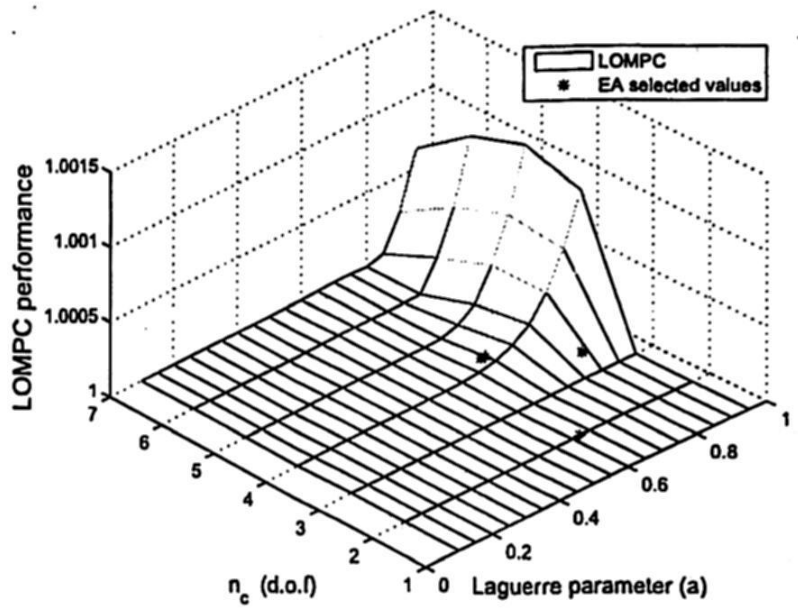

Fig. 3. Performance evaluation.

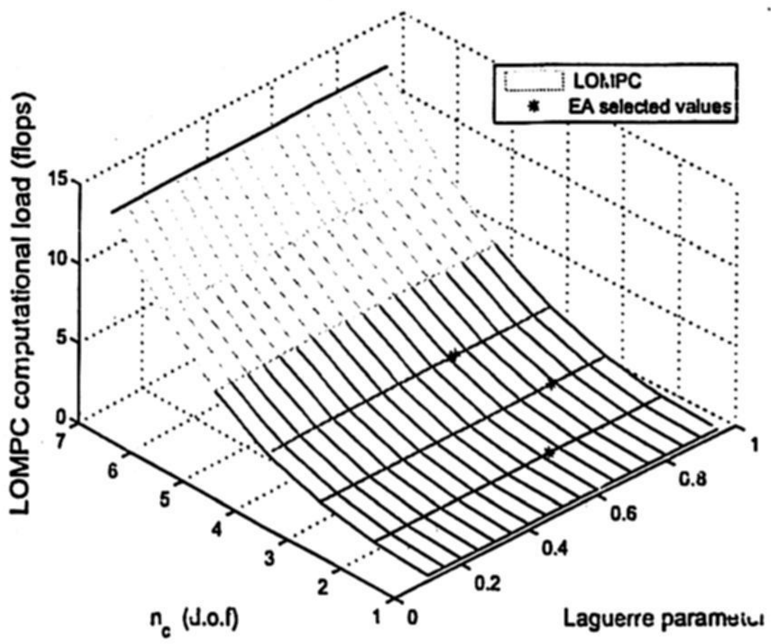

Fig. 4. Computational load evaluation.

LOMPC (left); (ii) an initial state in the MCAS of LOMPC, but outside of the MCAS of OMPC (right), where the latter is infeasible. Once again, it is clear that LOMPC with EA is better than OMPC under similar conditions (d.o.f.) and very similar to the global optimum.

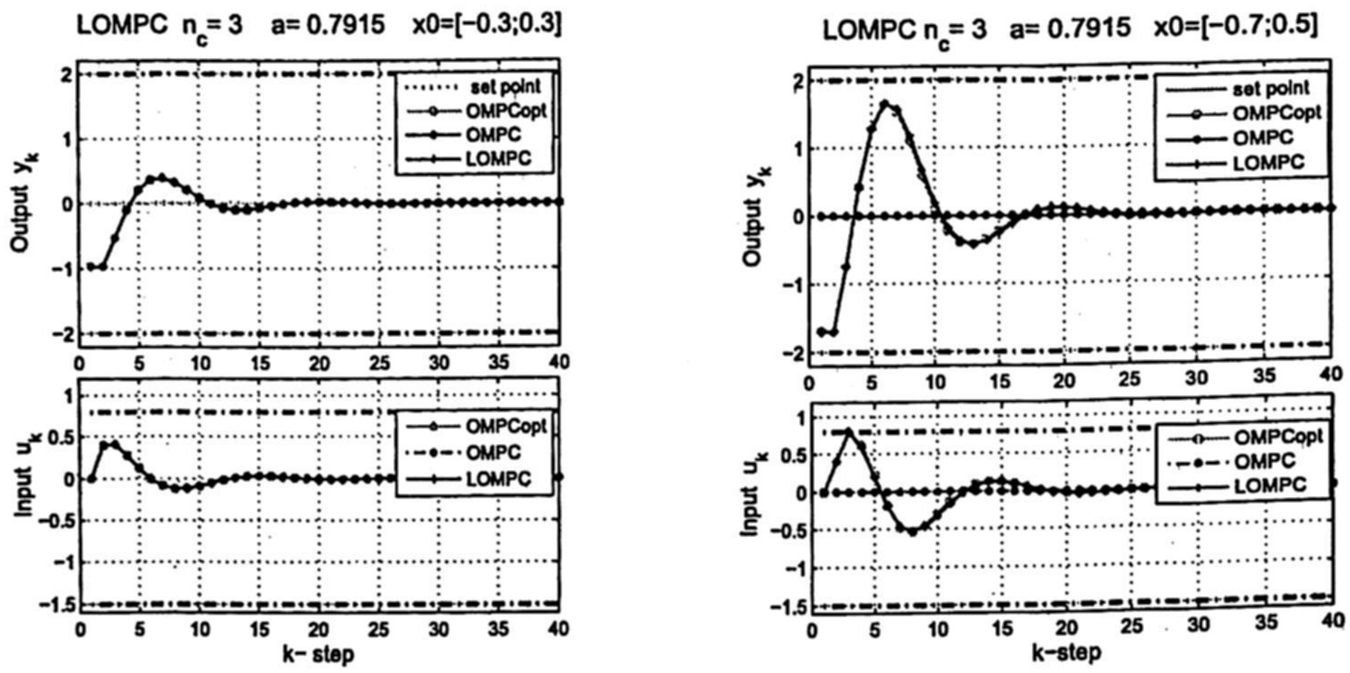

Fig. 5. Controllers simulation with different initial conditions.

6 Conclusions

Correct selection of the tuning parameters of the controller guarantees the best balance between closed-loop performance, feasible region and computational load. The main contribution of this work is to achieve a systematic tuning for predictive control algorithm, LOMPC. In this paper, it has been shown that MOEA can be a useful tool for obtaining one or more solutions to select the tuning parameters of an efficient MPC algorithm even when the search space is large and when there are included different objectives compromised to each other. It can be verified that this selection offer a proper balance in the trade-off between the three objectives and outperform OMPC in those objectives, under the same conditions and even for systems with constraints. However, this are just preliminary results and future work will include other efficient MPC approaches and systems with more challenging behavior.

References

1. Camacho, E.F. and Bordons, C.: Control predictivo: Pasado, presente y futuro. Revista Iberoamericana de Automática e Informática Industrial **1:3**, 5–28 (2004).
2. Rossiter, J.A.: Model Predictive Control: A practical aproach. Taylor and Francis (2005)
3. Cutler, C.R. and Ramarker, B.C.: Dynamic matrix control - a computed algorithm. (1979).

4. Clarke, D.W. ; Mohtadi, C. and Tuffs, P.S.: Generalized predictive control: the basic algorithm. Automatica **23,** 137–148 (1987).
5. Scokaert, P.O.M. and Rawlings, J.B.: Constrained linear quadratic regulation. IEEE Transactions on Automatic Control **43:8,** 1163–1168(1998).
6. Mayne, D.Q. ; Rawlings, J.B. ; Rao, C.V. and Scokaert, P.O.: Constrained model predictive control: Stability and optimality. Automatica **36:6,** 789–814, (2000).
7. Bacic, M. ; Cannon M. ; Lee, Y.I. and Kouvaritakis, B.: General interpolation in mpc and its advantages. IEEE Transactions on Automatic Control **48:6,** 1092–1096 (2003).
8. Imsland, L. ; Rossiter, J.A. ; Pluymers, B. and Suykens, J.: Robust triple mode mpc. International Journal of Control **81:4,** 679–687 (2008).
9. Valencia-Palomo, G. and Rossiter, J.A.: Using laguerre functions to improve efficiency of multi-parametric predictive control. In: Proceedings of the American Control Conference, Baltimore, USA (2010)
10. Khan, B. ; Valencia-Palomo, G. and Rossiter, J.A.: Exploiting kautz functions to improve feasibility in mpc. In: Automatic Control and Systems Eng. IFAC World Congress, Milán, Italia (2011)
11. Khan, B. and Rossiter, J.A.: Alternative parameterisation within predictive control: a systematic selection. International Journal of Control **86:8,** 1397–1409 (2013).
12. Valencia-Palomo, G. ; Rossiter, J.A. ; Jones, C.N. and Gondhalekar, R.: Alternative parameterisations for predictive control: how and why? In: American Control Conference, San Francisco, CA, USA (2011)
13. Muske, K.R. and Rawlings, J.B.: Model predicive control with linear models. American Institute of Chemical Engineers (AIChE) **39:2,** 262–287 (1993).
14. Rossiter, J.A.: A global approach to feasibility in linear mpc. In: Proceedings of the International Control Conference, Glasgow, Scotland (2006)
15. Maciejowski, J.M.: Predictive control with constraints. Pearson Education (2006)
16. Fleming, P.J. and Purshouse, R.C.: Evolutionary algorithms in control systems engineering: a survey. Control Engineering Practice **10,** 1223–1241 (2002).
17. Fogel, D.B.: Evolutionary Computation: Toward a New Philosophy of Machine Intelligence. 3 edn. John Wiley and Sons, Inc., New York, USA (2006)
18. Coello-Coello, C.A. ; Lamont, G.B. and Van-Veldhuizen, D.A.: Evolutionary Algorithms for Solving Multi-Objective Problems. 2 edn. Genetic and Evolutionary Computation, New York,USA (2007)
19. Zitzler, E.: Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications. PhD thesis, Institut für Technische Informatik und Kommunikationsnetze; Swiss Federal Institute of Technology Zurich, Computer Engineering and Networks Laboratory (1999)
20. Deb, K. ; Agrawal, S. ; Pratab, A. and Meyarivan, T.: A fast elitist non-dominated sorting geneticalgorithm for multi-objective optimization: Nsga-ii. In: Proceedings of the Parallel Problem Solving from Nature VI Conference. 849–858 (2000).
21. Deb, K. ; Pratap, A. ; Agarwal, S. and Meyarivan, T.: A fast and elitist multiobjetive genetic algorithm nsga-ii. IEEE Transactions on Evolutionary Computation **6:2,** 182–197(2002).