Applied Soft Computing 31 (2015) 326–338

Contents lists available at ScienceDirect

Applied Soft Computing

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

Systematic selection of tuning parameters for efficient predictive controllers using a multiobjective evolutionary algorithm

R.C. Gutiérrez-Urquídez [a], G. Valencia-Palomo [a,*], O.M. Rodríguez-Elias [a], L. Trujillo [b]

[a] Instituto Tecnológico de Hermosillo, Av. Tecnológico y Periférico Poniente s/n, 83170, Mexico
[b] Instituto Tecnológico de Tijuana, Calz. del Tecnológico s/n, Tomas Aquino, 22414, Mexico

ARTICLE INFO

Article history:
Received 9 October 2014
Received in revised form 21 January 2015
Accepted 23 February 2015
Available online 17 March 2015

Keywords:
Predictive control
Feasibility
Performance
Computational cost
NSGA-II

ABSTRACT

In the design of predictive controllers (MPC), parameterisation of degrees of freedom by Laguerre functions, has shown to improve the controller performance and feasible region. However, an open question remains: how to select the optimal tuning parameters? Moreover, optimality will depend on the size of the feasible region of the controller, the system's closed-loop performance and the online computational cost of the algorithm. This paper develops a method for a systematic selection of tuning parameters for a parameterised predictive control algorithm. In order to do this, a multiobjective problem is posed and then solved using a multiobjective evolutionary algorithm (MOEA) given that the objectives are in conflict. Numerical simulations show that the MOEA is a useful tool to obtain a suitable balance between feasibility, performance and computational cost.

1. Introduction

Model based predictive control (MPC) is the general name for different computer control algorithms that use past information of the inputs and outputs and a mathematical model of the plant to optimise the predicted future behavior [27,5,33]. MPC is well established and widely used, but there are still some theoretical and practical problems without a satisfactory answer. For instance, one key conflict is between feasibility, performance and online computational cost. A controller that is well tuned to give high performance will often have a relatively small feasible region unless a large number of decision variables (or degrees of freedom, d.o.f.) are used, which produces an increase of the online computational load of the algorithm. Conversely, with a strategy that focuses on producing a large feasibility region, the result will be a detuned controller with relatively poor performance [37].

This issue becomes particularly important when one tries to implement MPC on special purpose hardware, such as FPGA's [21,28], PLC's [32,19,45] or PAC's [18]. For these devices, linear MPC is demanding, as they have limited memory space and very low processing power. In this cases, even a small improvement could be the difference for a successful implementation; for example, Huyck et al. [18] analyses the maximum number of flops and memory space for a particular PLC and the conclusion is that (linear) MPC implementation is not always possible due to the computational cost and memory space needed. In contrast, in Rauová et al. [32] a very limited linear MPC is successfully embedded on a PLC with just 1024 bytes of memory space. Therefore, idea here is to find a suitable balance on the different performance indexes in order to contribute to the solution of this problem.

Several authors have proposed different strategies to implement the MPC algorithm and solve this problem; such as multiparametric solutions [3], time-varying control laws [26], fast optimisations [47], interpolations of different control laws [35,34], move-blocking [4,15], among others. Nevertheless, this paper will focus on algorithms with parameterised d.o.f. where the main idea is to form the degrees of freedom in the predictions as a combination of either Laguerre/Kautz polynomials or through generalised orthonormal functions [37,23,22], since they have proven to be very effective at improving the volume of the feasible region with a limited number of d.o.f. with almost no performance loss. The effectiveness of these approaches, is such that, there are successful experimental results on industrial hardware with limited memory and very low processing power [43].

Traditionally, conventional MPC controllers have been tuned by trial and error simulations or using thumb rules for the selection of some parameters, see for example the reviews by Rani and Unbehauen [31] and Garriga and Soroush [12]. In the case of automatic tuning, it is common to use some type of simplification [40,1,42] or excluding some particular aspects of the problem; for instance,

* Corresponding author. Tel.: +52 662 260 6500x136.
E-mail addresses: ro_gutierrez@ith.mx (R.C. Gutiérrez-Urquídez), gvalencia@ith.mx (G. Valencia-Palomo), omrodriguez@ith.mx (O.M. Rodríguez-Elias), leonardo.trujillo@tectijuana.edu.mx (L. Trujillo).

http://dx.doi.org/10.1016/j.asoc.2015.02.033
1568-4946/

not considering the horizons [25]. Other authors have been tuning MPC by solving different optimisation problems for specific types of MPC; interesting examples include: using particle swarm optimisation [16,41], fuzzy decision making [46], minimising dynamical indexes as performance measures [11], among others.

The original papers, proposing parameterisation of the d.o.f. for MPC algorithms, introduce one or more tuning parameters without guidelines for their selection [37,23,22]. The addition of these parameters make the tuning even more challenging. The tuning task can be particularly difficult since the whole set represents a large array of possible combinations and because many of these parameters have overlapping and/or contrary effects on the closed-loop performance and stability. In this case, the advantage of using an offline multiobjective optimisation tuning method is clear.

On the other hand, the use of evolutionary algorithms (EA) is becoming more accepted in the control community, since they offer a flexible representation of the decision variables, which facilitates the evaluation of controller performance [9,20]. Some applications include the solution of constrained combinatorial problems [30] and fuzzy multiobjective bi-level programming problems [8], performance optimisation of electrical systems [48], and the parameter selection of intelligent controllers [29], among others. Therefore, the main contribution of this paper is to propose a procedure to systematise the selection of controller tuning parameters of a MPC algorithm whose d.o.f.s have been parameterised using Laguerre functions. To select the controller parameters a multiobjective optimisation problem is formulated and solved using the Non-dominated Sorting Genetic Algorithm II (NSGA-II) as this EA offers a simple strategy for handling constraints in addition to being easy to adapt into design. Similar experiments of those presented in Khan et al. [23], Valencia-Palomo et al. [44], Rossiter et al. [37] are revised. The results demonstrate that EA is a useful tool to obtain a suitable balance between feasibility, performance and computational cost.

This paper is organised as follows: Section 2 gives the necessary background about predictive control and Laguerre optimal predictive control (LOMPC); Section 3 presents the evolutionary algorithm used to solve the multiobjective optimisation problem; Section 4 formulates the multiobjective optimisation problem; Section 5 shows numerical examples; and finally Section 6 presents the conclusions.

2. Predictive control and Laguerre functions

This section introduces the assumptions used in this paper and background information.

2.1. Model and constraints

Assume a state-space model of the form:

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

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 $\hat{\mathbf{d}}_k = \mathbf{y}_k - \mathbf{w}_k$. Disturbance rejection and offset free tracking will be achieved using the offset form of state feedback that is:

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

where $\mathbf{x}_k$ is the state of the independent model and $\mathbf{x}_{ss}$, are estimated values of the steady-states giving no offset; these depend upon the model parameters and the disturbance estimate.

Associated to the model are constraints of the form

$$\left.\begin{aligned} \mathbf{u}_{\min} \le \mathbf{u}_k \le \mathbf{u}_{\max}; \\ \triangle\mathbf{u}_{\min} \le \mathbf{u}_{k+1} - \mathbf{u}_k \le \triangle\mathbf{u}_{\max}; \\ \mathbf{y}_{\min} \le \mathbf{y}_k \le \mathbf{y}_{\max}. \end{aligned}\right\} \forall k. \tag{3}$$

In the context of predictive control, it is common to take the following quadratic performance index as the objective to be minimised at each sample:

$$J = \sum_{i=0}^{\infty} \left\{ (\mathbf{x}_{k+i} - \mathbf{x}_{ss})^T \mathbf{Q} (\mathbf{x}_{k+i} - \mathbf{x}_{ss}) + (\mathbf{u}_{k+i} - \mathbf{u}_{ss})^T \mathbf{R} (\mathbf{u}_{k+i} - \mathbf{u}_{ss}) \right\}, \tag{4}$$

with $\mathbf{Q} \in \mathbb{R}^{n\times n}$ and $\mathbf{R} \in \mathbb{R}^{m\times m}$ positive definite state and input cost weighting matrices.

2.2. Optimal predictive control (OMPC)

The key idea of optimal MPC (OMPC) [39,36] is to embed into the predictions the unconstrained optimal behaviour and handle constraints by using perturbations about this. Hence, assuming $\mathbf{K}$ is the feedback, the input predictions are defined as follows:

$$\mathbf{u}_{k+i} - \mathbf{u}_{ss} = \begin{cases} -\mathbf{K}(\mathbf{x}_{k+i} - \mathbf{x}_{ss}) + \mathbf{c}_{k+i}; & i \in \{0, \ldots, n_c - 1\} \\ -\mathbf{K}(\mathbf{x}_{k+i} - \mathbf{x}_{ss}); & i \in \{n_c, n_c + 1, \ldots\} \end{cases}, \tag{5}$$

where the perturbations $\mathbf{c}_k$ are the d.o.f. for optimisation; conveniently summarised in vector: $\underset{\rightarrow k}{\mathbf{c}} = [\mathbf{c}_k^T, \mathbf{c}_{k+1}^T, \ldots, \mathbf{c}_{k+n_c-1}^T]^T$.

It is known that for suitable $\mathbf{M}$, $\mathbf{N}$, $\mathbf{d}$ (e.g. [36]), the input predictions (5) and associated state predictions for model (1) satisfy constraints (3) if:

$$\mathbf{M}\mathbf{x}_k + \mathbf{N}\underset{\rightarrow k}{\mathbf{c}} \le \mathbf{d}(k). \tag{6}$$

It is easy to show [36] that optimisation of (4) over input predictions (5) is equivalent to minimising $J = \underset{\rightarrow k}{\mathbf{c}}^T \mathbf{W} \underset{\rightarrow k}{\mathbf{c}}$ ($\mathbf{W} = \mathbf{B}^T\Sigma\mathbf{B} + \mathbf{R}$, $\Sigma - \Phi^T\Sigma\Phi = \mathbf{Q} + \mathbf{K}^T\mathbf{R}\mathbf{K}$, $\Phi = \mathbf{A} - \mathbf{B}\mathbf{K}$) and thus, in the absence of constraints, the optimum is $\underset{\rightarrow k}{\mathbf{c}^*}$. Where the unconstrained predictions would violate constraints, non-zero $\underset{\rightarrow k}{\mathbf{c}^*}$ would be required to ensure constraint satisfaction.

Algorithm 1 *(OMPC)*. The OMPC algorithm is summarised as Scokaert and Rawlings [39,36]:

$$\begin{aligned} \underset{\rightarrow k}{\mathbf{c}^*} &= \arg\min_{\underset{\rightarrow k}{\mathbf{c}}} \ \underset{\rightarrow k}{\mathbf{c}}^T \mathbf{W} \underset{\rightarrow k}{\mathbf{c}} \\ \text{s.t.} \quad & \mathbf{M}\mathbf{x}_k + \mathbf{N}\underset{\rightarrow k}{\mathbf{c}} \le \mathbf{d}(k) \end{aligned} \tag{7}$$

Use the first element of $\underset{\rightarrow k}{\mathbf{c}^*}$ in the control law of (5), with $\mathbf{K}$.

This algorithm will find the global optimal, with respect to (4), whenever that is feasible and has guaranteed convergence/recursive feasibility in the nominal case.

OMPC algorithm has implied linear-quadratic-regulator (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 LQR [39], the feasible region depends only on the class of prediction, and hence also the number of free movements, that is, n_c.

Definition 2.1 *(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. The greatest invariant set possible for use as the terminal state set is referred as maximum admissible set (MAS) [14]. For a linear discrete system, observable, pre-stabilised by a gain $\mathbf{K}$ of state