

INTERVAL MODEL PREDICTIVE CONTROL

J.M. Bravo¹, C.G. Varet¹ and E.F. Camacho²

¹ *Departamento de Ingeniería Electrónica, Sistemas Informáticos y Automática
Universidad de Huelva (Spain)*

² *Departamento de Ingeniería de Sistemas y Automática
Universidad de Sevilla (Spain)*

Abstract: Model Predictive Control is one of the most popular control strategy in the process industry. One of the reason for this success can be attributed to the fact that constraints and uncertainties can be handled. There are many techniques based on interval mathematics that are used in a wide range of applications. These interval techniques can mean an important contribution to Model Predictive Control giving algorithms to achieve global optimization and constraint satisfaction. *Copyright © 2000 IFAC*

Keywords: Predictive Control, Intervals, Constraints, Nonlinear Control, Optimization,

1. INTRODUCTION

Model Predictive Control (MPC) is a very ample range of control methods developed around certain common ideas. A model is used to predict the future plant outputs. The elaboration of mathematical models of processes in real life requires simplifications to be adopted. In practice, no model capable of exactly describing a process exists. Therefore, no model can be considered to be complete without taking into account possible modeling errors or uncertainties (Camacho and Berenguel, 1997).

In practice, all processes are subjected to constraints. The control system normally operates close to the limits and constraints violations may occur. The control system has to anticipate constraints violations and correct them in a appropriate way.

If no constraints are present, the model process is linear and the cost function is quadratic, the resulting control law is easy to implement and requires little computation. However, its derivation is more complex than that of the classical PIDs controllers. If the process dynamic does not change, the derivation of the controller can be done

beforehand, but in the adaptive control case all the computation has to be carried out at every sampling time.

When constraints are considered, the control signal has to be computed using a numerical optimization algorithm and the amount of computation required is even higher. The design algorithm is based on a prior knowledge of the model and it is independent of it, but it is obvious that the benefits obtained will depend on the discrepancies existing between the real process and the model used.

This work proposes control algorithms that can be used with linear and nonlinear models, quadratic or non quadratic objective function, linear and nonlinear constraints and bounded uncertainties.

The plant to be controlled can be described by the following non linear state-space model:

$$\begin{aligned}x(k) &= f(x(k-1), u(k), \theta(k)) \\ y(k) &= g(x(k))\end{aligned}\tag{1}$$

Where $u(k)$ is a vector of inputs or manipulated variables, $x(k)$ is a vector of state variables, $\theta(k)$ is a

vector of uncertainties and $y(k)$ is a vector of controlled variables or outputs.

Section 2 shows Model Predictive Control formulation. In sections 3 and 4, interval techniques for modeling and nonlinear constraints solving are described. Section 5 proposes a new Model Predictive Control strategy: Interval Model Predictive Control (IMPC). In order to illustrate the algorithm an application to a nonlinear model of an evaporator is presented in section 6.

2. MODEL PREDICTIVE CONTROL

Model Predictive Control (MPC) does not designate a specific control strategy but a very ample range of control methods which make an explicit use of a model of the process to obtain the control signal by minimizing an objective function. The ideas appearing in greater or lesser degree in all the predictive control family are basically (Camacho and Bordons, 1999): Explicit use of a model to predict the process output at future time instants (horizon). Calculation of a control sequence minimizing an objective function. Receding strategy, so that at each instant the horizon is displaced towards the future, which involves the application of the first control signal of the sequence calculated at each step.

No model capable of exactly describing a process exists. Parametric uncertainties are uncertainties in the parameters of the model without changing its order. That is, the structure of the process is the same as the model's but with parameters differing from the real ones.

It is difficult to handle uncertainties in the process model because its formulation is complex in many cases. Interval mathematics can be used to model bounds uncertainties, it can reduce the complexity of this formulation.

In practice all processes are subject to constraints. Actuators have a limited range of action and a limited slew rate. The operating points of plants are determined to satisfy economic goals and lie at the intersection of certain constraints. There are many aspects that cause bounds in process variables.

The constraints acting on a process can originate from amplitude limits in the control signal, slew rate limits of the actuator, and limits on the output signals. However, other types of constraints may exist: band constraints, overshoot constraints, monotonic behavior, actuator non linearities and others (Camacho and Bordons, 1999).

It may occur that process or constraints are not linear, or cost function are not quadratic. In these

cases global optimization algorithms using interval analysis can solve this problem.

3. INTERVAL TOOLS

Interval Mathematics has been used since the early 60s giving rise to the development of algorithms to achieve global optimization and constraints satisfaction. Interval Mathematics can be said to have begun with the appearance of R.E. Moore's book *Interval Analysis* (1966). Moore's work transformed this simple idea into a viable tool for error analysis. Instead of merely treating rounding errors, Moore extended the use of Interval Analysis to bound the effect of errors from all sources, including approximation errors and errors in data.

Since the appearance of Moore's book, over 1000 publications on interval analysis has appeared as journal articles and reports. Over two dozen books are devoted entirely or in part to the subject (Moore, 1966; Neumaier, 1990; Hansen, 1992; Kearfott, 1996).

The main idea used in Interval Analysis is to consider that a variable X does not take a real value but is instead define by and interval. That is:

$$X=[a,b]=\{x:a\leq x\leq b\} \quad x,a,b\in\mathfrak{R} \quad (2)$$

All the variables and constants used to model or control systems will take interval values. In this case is important to have a set of tools which allow us to work with the previous variables and constants. This set of tools is formed by Interval Arithmetic, Extended Relation Operators and Interval Functions.

3.1 Interval Arithmetic

Interval Arithmetic is the set of arithmetic operations defined over intervals. Definition of addition, subtraction, multiplication and division are showed below.

$$\begin{aligned} [a,b]+[c,d]&=[a+c,b+d] \\ [a,b]-[c,d]&=[a-d,b-c] \\ [a,b]\cdot [c,d]&=[\min(ac,ad,bc,bd),\max(ac,ad,bc,bd)] \\ [a,b]/[c,d]&=[1/d,1/c]\cdot [a,b] \end{aligned} \quad (3)$$

Extended and generalized operations has been developed (Hansen, 1992).

3.2 Extended Relation Operators

To provide operators to compare two intervals are needed. The semantic of these operators over real numbers is clear, but a definition is necessary over intervals:

$$\begin{aligned}
 [a, b] = [c, d] &\Leftrightarrow [a, b] \cap [c, d] \neq \emptyset \\
 [a, b] > [c, d] &\Leftrightarrow b > c \\
 [a, b] < [c, d] &\Leftrightarrow a < d \\
 [a, b] \geq [c, d] &\Leftrightarrow b \geq c \\
 [a, b] \leq [c, d] &\Leftrightarrow a \leq d
 \end{aligned}
 \tag{4}$$

This definition favors that all the solutions of the problem stay in the space of study, however there may be definitions more appropriate for others applications. To get this robustness, the definition, has been carried out following this criterion:

$$XopY \Leftrightarrow \exists x \in X, \exists y \in Y | xopy
 \tag{5}$$

3.3 Interval Functions

An Interval Function is an interval extension of a real function. There are several ways of extending (Hansen, 1992). Natural Interval Extension has been used in this work. Other kind of extensions and their influence in the control will be studied in later works.

The Natural Interval Extension F of a real function f is formed as the following: Each constant k of f is substituted by the interval $[k, k]$. Each real variable x of f is substituted by an interval variable X . Each real operator op is substituted by an interval operator OP

4. BRANCH & BOUND ALGORITHMS

The Branch and Bound algorithms are tools used to solve a lot of problems. Its methodology consists on dividing a certain search space into smaller spaces where the search is easier. The solution of the global problem will be the best solution of the smaller problems. The best advantage of these algorithms is the guarantee of offering a global solution against local methods.

4.1 Traditional definition

Traditional works (Mitten, 1970) define the important points that characterize the Branch and Bound algorithms:

Problem to solve and search space.

Division Rule; define how to divide the search space.

Bound Rule; define a limit to solve the problem in a certain space.

Selection Rule; allow to select the following space to work with.

Suppression Rule; define what spaces can be suppressed because of its lack of solution.

4.2 Predictive Control formulation

In order to solve a problem of Model Predictive Control, it should be taken into account the following points:

Problem to solve and search space; in MPC the problem is to find the set of control signals that minimize a cost function, so the search space is composed by the control signals domains.

Division Rule; a simple bisection will be used to divide the present search space.

Bound Rule; interval Arithmetic allow to evaluate the cost function in intervals obtaining an upper and lower value.

Selection Rule; select the space, heuristically, which has the higher probability of containing an optimal signal control.

Suppression Rule; two different criteria will be considered: Verify if the present search space may have a better solution that the one already obtained. If it is not possible, it will be dismissed. And verify if the present search space may satisfy the constraints imposed to control and in negative case, to dismiss this space.

5. INTERVAL MODEL PREDICTIVE CONTROL

The present work studies the viability to make an interval extension of the predictive control which permits to solve problems that are difficult for traditional algorithms. For example, non linear models and constraints and modeling uncertainties through interval values.

5.1 Problem Formulation

Consider the plant to be controlled described by (1).The problem to be solve by the interval model predictive controller at each sampling time may be stated as follows:

$$\min_u J(u(k), y(k), \omega(k), \theta(k))$$

Subject to

$$C_1(u(k)), C_2(y(k)), C_3(\theta(k)) \quad (6)$$

J defines an objective function over a finite control horizon, $w(k)$ defines the set point sequence and $C_i(k)$ are sets of non linear constraints.

The key idea of the algorithm is that all variables and constants are given an interval value instead of a real value. That is, uncertainties both in model parameters and errors are given the interval value defined by the minimum and maximum value that the uncertainty can take. In general, the uncertainties, when bounded, can be described by a set of non linear constraints.

The problem is to determine $u(k)$, in an interval mathematics form such that it minimizes J while satisfying the sets of constraints $C_i(k)$. Notice that the decision variables $u(k)$, $\theta(k)$ and $w(k)$ are exogenous variables and $y(k)$ is predicted from the model equations and the values taken by the other variables. But this prediction is made using interval arithmetic.

Three steps are needed to define the problem of the predictive control in the interval domain field: First make the interval extension of the cost function and the model constraints, according to the steps described above. Second design an algorithm to resolve the sets of constraints. Third design an algorithm to minimize the cost function.

5.2 Algorithm to resolve the sets of constraints

There are a lot of works (Kearfott, 1996) about the constraints resolution based on Branch and Bound algorithms and interval techniques. In this work two main algorithms have been developed: First an algorithm of constraint satisfaction allows to know if a certain solution space satisfies the constraints imposed to the control. Those algorithms consist on replacing the interval domains of the solution space in the variables and verifying through relational operators, whether these constraints are satisfied or not. Second an algorithm to bound the solution space in charge of eliminating those parts of the space that do not satisfy constraints, returning a surrounding bound of the minimum space that satisfies them. In control, it is interesting to found a bound of the space adjusted enough to satisfy those constraints. The algorithm adjust made in that bound will depend on the required precision, the available time to make the calculations, and the overestimation that interval arithmetic introduces.

Algorithm to bound the space that satisfies constraints.

```

X = InitialSpace
C = Constraints
if NotSatisfyConstraints(C,X) ToReturn ListEmpty
List=Insert (X)
While Precision reached
    X = FirstElement(List)
    List = List -X
    (X1,X2) = SplitSpace(X)
    if SatisfyConstraints(C,X1) List=List + X1
    if SatisfyConstraints(C,X2) List=List + X2
EndWhile
Return UnionSpace(List)

```

5.3 Algorithm to minimize functions

The development of global optimization algorithms has been a great success in interval analysis (Hansen, 1992; Kearfott 1996). Those algorithms allow to find the global minimum of functions subject to constraints. These functions and constraints can be linear or non linear and differentiable and non differentiable. The more information the functions and the constraints have, the higher the convergence speed will be. For this reason, the minimization of differentiable functions will permit methods with a higher degree of convergence.

This work has developed a valid algorithm for non linear and non differentiable functions and constraints. The algorithm has as an input an initial search space, a cost function and a set of constraints. Specifying the predictive control problem, we found:

Initial search space; it is the set domain of the present and future inputs that compound the predictive control. The dimension of that space will depend on the number of input variables and horizon control and prediction used.

Cost function; its minimization permits to calculate the optimal control signals.

Input constraints; constraints imposed to control.

A first step in the algorithm consists on bounding the search space to the constraints as much as the interval techniques allow. This very first part of the algorithm returns a subspace that bound externally the defined space by the constraints. Next, the subspace is inserted in an ordered list used by the algorithm to keep the different subspaces created. The following step will be to execute a loop that will be repeating while a certain precision is reached. This precision will be calculated by the subspaces width. The loop starts taking the first element of the list. The mid point is calculated, and

the cost function is evaluated with these values. A minimal bound is obtained. If that bound is better than the ones obtained previously, the bound is taken. Following, the space is split into two subspaces. The following operations are made with these two subspaces: Verify that constraints are satisfying, and in a negative case, to dismiss them; evaluate the cost function obtaining an upper and lower bound for these subspaces and verify that the lower bound does not exceed the present minimum (if it does, it will be dismissed) and to insert it in an ordered way in the list. The list can be ordered according to the best upper or best lower bound.

Once the precision is obtained by achieving the subspaces width wished, the algorithm ends returning the first of the list.

Algorithm to minimize cost functions

```

X = InitialSpace
C = Constraints
J = FunctionToMinimize
List = Insert(BoundSpaceSatisfyConstraintns(C,X))
While PrecisionNonReached
    X = FirstElement(List)
    Minimum = MidPointTest(X)
    List = List - X
    (X1,X2) = SplitSpace(X)
    if SatisfyConstraint(C,X1)
        Evaluate J with X1
        if LowBound(J) > Minimo
            List = InsertOrdered(X1)
        Endif
    Endif
    if SatisfyConstraint(C,X2)
        Evaluate J with X2
        if LowBound(J) > Minimo
            List = InsertOrdered(X1)
        Endif
    Endif
EndWhile
Return FirstElement(List)

```

More efficient algorithms have been developed, however it is needed the use of the gradient (Van Hentenryck, 1997).

6. SIMULATION

An evaporator has been chosen as a testing bed for interval model predictive control. The results presented in this section have been obtained by simulation on a non linear model of the process (Newell and Lee, 1989).

The system dynamics is mainly dictated by the differential equations modeling the mass balance. In the solute the mass balance can be expressed by the differential equation:

$$M \frac{dX_2}{dt} = F_1 X_1 - F_2 X_2 \quad (7)$$

Where M is a constant that defines the total quantity of liquid in the evaporator, F_1 is the feed flowrate, X_1 is the feed composition, F_2 is the product flowrate and X_2 is the product composition. F_2 is the manipulated variable. X_2 is the process variable. F_1, X_1 are disturbances.

Taking a sampling time of one minute, the non linear discrete model used by the interval controller is:

$$X_2(k) = X_2(k-1) + \frac{(F_1(k) \cdot X_1(k) - F_2(k) \cdot X_2(k))}{20} \quad (8)$$

Four results of applying the interval controller are presented. Figure 1 shows the system response and the control signal without disturbances and without constraints. Figure 2 shows the system response and the control signal with a constraint over X_2 . Figure 3 shows the system response and the control signal with a disturbance of $\pm 10\%$ in X_1 . Figure 4 shows the system response and the control signal with the same disturbances but using an interval model where these disturbances are considered.

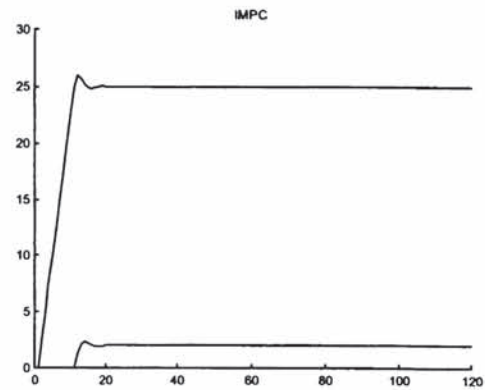


Fig. 1. Interval control without constraints, without disturbances.

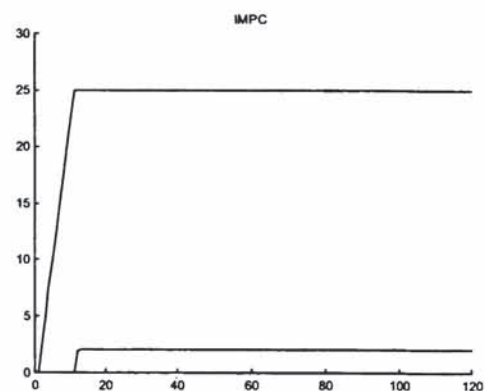


Fig. 2. Interval Control with $X_2 \leq 25$, without disturbances

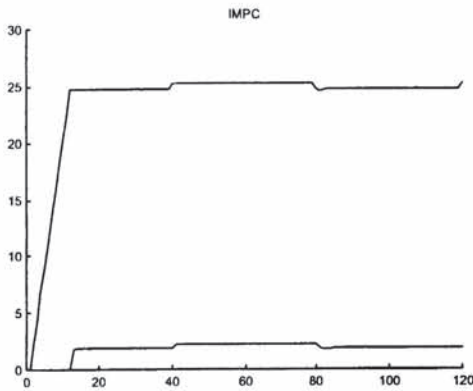


Fig. 3. Interval Control with disturbances not considered by the model.

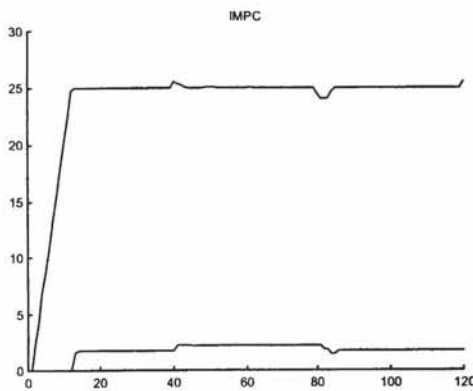


Fig. 4. Interval Control with disturbances considered by the model.

7. CONCLUSIONS

In this work, a set of interval methods for application in a MPC framework is proposed. These methods allow to use nonlinear models, nonlinear constraints and bounds uncertainties. However, there are some limitations: High computational cost, so its application in real time control is difficult if the process is fast; high storage cost (to handle spaces generated by branch & bound algorithms) and function overestimation in Interval mathematics.

Hardware development more and more fast make possible the application of interval methods.

REFERENCES

- Camacho, E.F. and Bordons, C. (1999). *Model Predictive Control*. Springer-Verlag.
- Camacho, E.F. and Berenguel, M. (1997). Robust Adaptive Model Predictive Control of a Solar Power Plant with Bounded Uncertainties, *International Journal of Adaptive Control and Signal Processing*, Vol. 11, pp 311-325.

- Hansen, E. (1992). *Global Optimization Using Interval Analysis*. Marcel Dekker, Inc. New York.
- Kearfott, R.B. (1996) *Rigorous Global Search: Continuous Problems*. Kluwer Academic Publishers.
- Mitten, L.G. (1970). Branch and Bound methods: general formulation and properties. *Operation Research* 18:24,34.
- Moore, R.E. (1966). *Interval Analysis*. Prentice Hall.
- Neumaier, A. (1990) *Interval Methods for Systems of Equations*. Cambridge University Press, Cambridge
- Newell, R.B. and Lee, P.L. (1989). *Applied Process Control. A case Study*. Prentice Hall
- Van Hentenryck, P. (1997) *Numerica: A Modeling Language for Global Optimization*. The MIT Press, Cambridge, Massachusetts.