Proceedings of the
44th IEEE Conference on Decision and Control, and
the European Control Conference 2005
Seville, Spain, December 12-15, 2005

Nonlinear Model Predictive Control via Interval Analysis

F. Lydoire and P. Poignet

Abstract— **This paper deals with the application of interval arithmetic to non linear model predictive control (NMPC) problem. The NMPC problem is expressed as a constraints satisfaction problem (CSP) which can be solved by interval analysis techniques. We present the classical interval techniques to build a nonlinear model predictive control law and propose some improvements in order to adapt interval tools to the context of control. Moreover, in order to reduce the pessimism introduced by interval state estimation, we propose a spatial discretisation of the input. These methods are illustrated on a inverted pendulum model.**

I. INTRODUCTION

The ability to handle nonlinear multi-variable systems that are constrained in the state and/or in the control variables motivates the use of Nonlinear Model Predictive Control (NMPC). This approach has proved its efficiency in a large variety of industrial processes, especially on chemical processes [1]. The NMPC problem is usually stated as an optimization one subject to physical coherent constraints, and is solved with classical optimization algorithms. Most of the constraints involved in the NMPC are easily expressed using interval, therefore we investigate in this paper the use of interval analysis techniques [2] in order to find an NMPC constraints satisfying solution.

The use of interval analysis techniques for synthesizing a predictive control scheme brings up some problems related to the pessimism in interval computations which leads to an ineffective state estimation. Therefore, we propose some improvements of classical interval techniques and a new approach based on a spatial discretisation of the considered domains to improve the interval state estimation. The proposed strategy is numerically validated on an inverted pendulum model.

The paper is organized as follows : section II presents the classical nonlinear model predictive control technique, section III introduces interval analysis: constraints propagation, interval state estimation and spatial discretisation to reduce pessimism in state estimation. Finally section IV exhibits numerical simulation results with NMPC and interval analysis techniques on an inverted pendulum model.

F. Lydoire and P. Poignet are with the Laboratoire d'Informatique de Robotique et de Microélectronique de Montpellier, UMR CNRS UM2 5506, 161 rue Ada, 34392 Montpellier, Cédex 5, France; phone: +33 4 67 41 85 61; fax: +33 4 67 41 85 00; email: {lydoire,poignet}@lirmm.fr

II. NONLINEAR MODEL PREDICTIVE CONTROL

The NMPC problem [1] is usually formulated as a constrained optimization problem

$$\min_{\boldsymbol{u}_k^{N_p}} J(x_k, \boldsymbol{u}_k^{N_p}) \tag{1}$$

subject to

$$x_{i+1|k} = f(x_{i|k}, u_{i|k}) \qquad x_{0|k} = x_k \tag{2}$$

$$u_{i|k} \in \mathbb{U}, \quad i \in [0, N_p - 1] \tag{3}$$

$$x_{i|k} \in \mathbb{X}, \quad i \in [0, N_p] \tag{4}$$

where

$$\begin{array}{l} \mathbb{U} := \{u_k \in \mathbb{R}^m | u_{\min} \leq u_k \leq u_{\max}\} \\ \mathbb{X} := \{x_k \in \mathbb{R}^m | x_{\min} \leq x_k \leq x_{\max}\} \end{array} \tag{5}$$

Internal controller variables predicted from time instance k are denoted by a double index separated by a vertical line where the second argument denotes the time instance from which the prediction is computed. $x_k = x_{0|k}$ is the initial state of the system to be controlled at time instance k and $\boldsymbol{u}_k^{N_p} = [u_{0|k}, u_{1|k}, \ldots, u_{N_p-1|k}]$ an input vector.

Predictive control (Fig. 1) consists in computing the vector $\boldsymbol{u}_k^{N_p}$ of consecutive inputs $u_{i|k}$ over the prediction horizon N_p by optimizing the objective function J subject to constraints (2),(3),(4) and applying only the solution input $u_{0|k}$. These computations are updated at each sampling time.

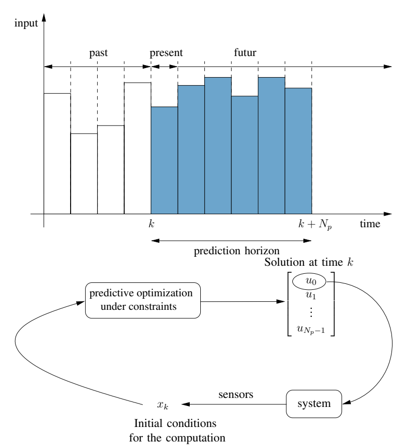

Fig. 1. Principles of the predictive constrained optimal control approach

The nonlinear equality constraint on the state (2) represents the dynamic model of the system. Bounding constraints over the inputs $u_{i|k}$ and the state variables $x_{i|k}$ over the prediction horizon N_p are defined through the sets $\mathbb{U}$ and $\mathbb{X}$.

0-7803-9568-9/05/$20.00 ©2005 IEEE

The objective function J is usually defined as

$$J(x_k, \boldsymbol{u}_k^{N_p}) = \phi(x_{N_p|k}) + \sum_{i=0}^{N_p-1} L(x_{i|k}, u_{i|k}) \qquad (6)$$

where ϕ is a constraint on the state at the end of the prediction horizon, called state terminal constraint, and L a quadratic function of the state and inputs.

The computation of the solution $\boldsymbol{u}_k^{N_p}$ can be divided in two steps : firstly, computation of a solution satisfying the constraints (including the state terminal constraint), and secondly optimization. The first step involves bounding constraints (3,4), which can be directly expressed as interval constraints and nonlinear constraints expressing the dynamic model of the system (2). These constraints defined by the NMPC formulation may be expressed as intervals and interval techniques may be used to compute a satisfying input.

Therefore we will use interval techniques to compute an input satisfying the constraints defined by the NMPC formulation.

III. INTERVAL ANALYSIS

Initially dedicated to finite precision arithmetic for computer [3] and after used in a context for guaranteed global optimization [4], the interval analysis is based on the idea of enclosing real numbers in intervals and real vectors in boxes. Interval computation which is a special case of set theory, is now used for state estimation or robust control [5], [6]. To well understand the basics of interval computation, the reader is invited to read [2].

A. Interval constraint propagation

The problem considered during constraint propagation is a *Constraint Satisfaction Problem* (CSP) $\mathcal{H}$ written as a vector form $f(\boldsymbol{x}) = 0$.

$$\mathcal{H} : (f(\boldsymbol{x}) = 0,\ \boldsymbol{x} \in [\boldsymbol{x}]) \qquad (7)$$

The solution set of $\mathcal{H}$ is defined as

$$\mathbb{S} = \{\boldsymbol{x} \in [\boldsymbol{x}] \mid f(\boldsymbol{x}) = 0\} \qquad (8)$$

Contracting $\mathcal{H}$ means replacing $[\boldsymbol{x}]$ by a smaller domain $[\boldsymbol{x}']$ such that the solution set remains unchanged.

Different methods to contract a CSP are available [2], and we introduce a particular one called the forward-backward contractor [7], based on constraint propagation. This contractor makes it possible to contract the domains of the CSP by taking into account any one of the constraints in isolation.

Construction of the contractor is done as follows. Firstly, each constraint of the CSP is transformed into a set of equations obtained by isolating each variable of this constraint. Secondly, these new equations are transformed into subsolvers. Finally, these subsolvers are rewritten into interval equations, no more considering the real variables but their interval counterpart. Extending a punctual function into an interval function is done using inclusion function. Let's consider a function f from $\mathbb{R}^n$ to $\mathbb{R}^m$. The interval function $[f]$ from $\mathbb{IR}^n$ to $\mathbb{IR}^m$ (where $\mathbb{IR}$ is the set of all interval real numbers) is an inclusion function for f if

$$\forall [\boldsymbol{x}] \in \mathbb{IR}^n,\ f([\boldsymbol{x}]) \subset [f]([\boldsymbol{x}]) \qquad (9)$$

Contracting a CSP is contracting the equations of the CSP one by one and iterating the procedure while contraction is significant.

Contraction reduces the intervals within which the solutions of the CSP are included, but it does not guarantee that all the values in these intervals are solutions. In addition to the contraction, a validation procedure is then required.

Moreover, the contraction procedure may lead to a fixed point. In that case, the fixed point is *broken* by bisecting domains involved in the CSP. Contraction will be illustrated in section IV.

B. State estimation

During the constraint propagation, the state of the system is estimated over the horizon. The interval values are computed using the following equation

$$[f]([x_k], [u_k]) = [x_{k+1}] \qquad (10)$$

where f represents the dynamic of the system.

We must notice that the domain $[x_{k+1}]$ is different from the domain

$$\mathbb{X}_{k+1} = \{f(x_k, u_k), \forall x_k \in [x_k] \text{ and } \forall u_k \in [u_k]\} \qquad (11)$$

representing the set of possible punctual states for x_{k+1}.

$\mathbb{X}_{k+1} \subseteq [x_{k+1}]$ is guaranteed by interval computations, but the situation $\mathbb{X}_{k+1} = [x_{k+1}]$ is very rare. The function f regroups the dynamic equation of the system and a double integration. Many occurrences of the same variables appear in f and it is therefore very pessimistic in terms of estimation of the set $\mathbb{X}_{k+1}$.

The computation of smallest domains in ordinary differential equations is an open problem in mathematics [8] and also specifically for state estimation [9].

The computation of the state with the inclusion function $[f]$ leads to an outer estimation of the states which propagates over the horizon. Therefore, all the domains considered in the CSP are outer estimations.

This usually leads to the selection of a possible very small domain for the input satisfying the constraints. Due to the pessimism of the state estimation, the constraints propagation is very inefficient. Furthermore, the domain of the input has to be small in order to respect the constraints in spite of the pessimism. The selection of the input domain is therefore done using bisection which involves many computations and is very ineffective.

C. Inner state estimation

The computation of the exact domain of the state (11) is not possible. Usual techniques based on Taylor series lead to guaranteed but outer estimation of the state which is not effective in the case of predictive control. As the exact estimation of the state being impossible, we will compute an inner estimation. We propose to compute this estimation by using a spatial discretisation of the domains.

On each iteration, the set of inputs $u_k^1, \ldots, u_k^n$ which define a spatial distribution of the input constraint $[u_{k_{\min}}, u_{k_{\max}}]$, is applied on each punctual state values $x_k^1, x_k^2, \ldots, x_k^m$ defining a spatial discretisation of $[x_k]$. This gives a new set of punctual values defining a spatial discretisation of $[x_{k+1}]$ (Fig. 2).

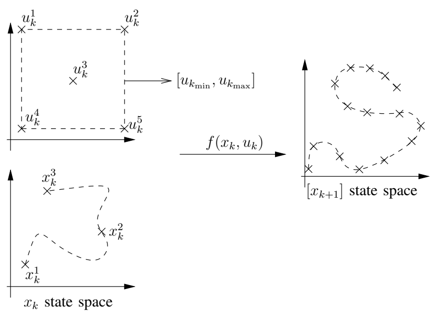

Fig. 2. Spatial discretisation

Assuming that f is continuous, the spatial discretisation of $[x_{k+1}]$ computed by the algorithm provides an inner approximation of $f([x_k], [u_{k_{\min}}, u_{k_{\max}}])$. Indeed, for any punctual value x_k^p in $[x_k]$, $p \in [1, m]$, and any inputs u_k^l and u_k^{l+1}, $l \in [1, n-1]$ continuity of f leads to

$$[\min(f(x_k^p, u_k^l), f(x_k^p, u_k^{l+1})), \max(f(x_k^p, u_k^l), f(x_k^p, u_k^{l+1}))] \subseteq f(x_k^p, [u_k^l, u_k^{l+1}]) \quad (12)$$

therefore the set of input variables $\mathbb{S}'$ considering the inner approximation of the state

$$\begin{aligned} \mathbb{S}' = \{u_k \in [u_k^l, u_k^{l+1}] \mid & \\ & [\min(f(x_k, u_k^l), f(x_k, u_k^{l+1})), \\ & \max(f(x_k, u_k^l), f(x_k, u_k^{l+1}))] \\ & \subseteq [x_{k+1_{\min}}, x_{k+1_{\max}}]\} \end{aligned} \quad (13)$$

is an inner approximation of the set of input variables $\mathbb{S}$ in case of perfect state estimator over intervals.

$$\mathbb{S} = \{u_k \in [u_k^l, u_k^{l+1}] \mid f(x_k, [u_k^l, u_k^{l+1}]) \subseteq [x_{k+1_{\min}}, x_{k+1_{\max}}]\} \quad (14)$$

The inner approximation of the state of the system allows the efficient use of the contraction algorithm. The efficiency of the solution depends on the sampled values u_k^l of the initial input interval $[u_{k_{\min}}, u_{k_{\max}}]$, and on the accuracy threshold ϵ defining the minimum width for an interval allowed to be bisected during the set inversion procedure.

One of the drawback of the inner approximation of the state is that state values outside the inner approximation are not considered and therefore could violate the constraints (Fig. 3). This leads to the validation of an incorrect input interval. However, the punctual values defining the spatial discretisation of the state are guaranteed to belong to the constrained space. These values have been computed from punctual input values defining the spatial discretisation of the input interval. Therefore, these punctual input values are guaranteed to respect the constrained state space. However, tacking any punctual value in the computed input interval may lead to constraint violation. This constraint violation has not been characterized yet and will be the object of future work.

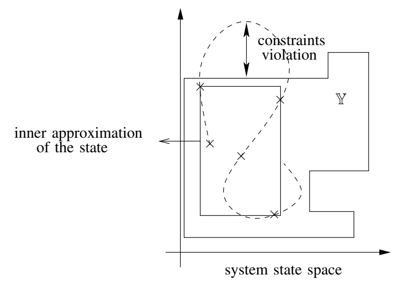

Fig. 3. Constraints violation due to the inner approximation of the state

IV. APPLICATION TO THE CONTROL OF AN INVERTED PENDULUM

To illustrate the efficiency of the proposed approach, we compute the strategy for controlling an inverted pendulum.

A. The model

Let's consider the inverted pendulum which is a classical benchmark for non linear control techniques [10], [11]. Its dynamic equation is:

$$\ddot{q}_{k+1} - K_{\sin}\, \sin(q_k) + K_{\cos}\, \ddot{x}_k\, \cos(q_k) = 0 \quad (15)$$

where $K_{\sin}$ and $K_{\cos}$ regroups parameters of a pendulum available at the laboratory and $\ddot{x}$ the acceleration of the carrier. Friction has been neglected and it has been assumed that the pendulum is a rigid body.

The state of the pendulum $[q_{k+1}, \dot{q}_{k+1}]^T$ at time instance $k+1$ from the state at time instance k and the input u_k (the acceleration $\ddot{x}_k$) is estimated through the dynamic equation (15). The acceleration of the pendulum $\ddot{q}_{k+1}$ is then integrated twice to get the state of the pendulum at time instance $k+1$.

In the following, the integration is computed using first order Taylor series in the predictive controller

$$\dot{q}_{k+1} = \dot{q}_k + \delta_t\, \ddot{q}_{k+1} \quad (16)$$

$$q_{k+1} = q_k + \delta_t\, \dot{q}_{k+1} \quad (17)$$

and using Runge-Kutta formula in the simulator.

B. NMPC control

To compare an interval approach with the classical approach, we first apply a standard predictive controller on the pendulum. The problem is formulated as an NMPC problem (section II) but with no reference trajectory. Moreover, in order to be able to compare this controller with interval controller, we consider a single input value applied over the horizon.

The objective function is defined as a quadratic product of the input with a stability constraint over the state at the end of the horizon.

$$\min_{\boldsymbol{u}_k^{N_p}} \frac{1}{2}\,(\boldsymbol{u}_k^{N_p})^T\,\boldsymbol{u}_k^{N_p} + X_{N_p}^T\,P\,X_{N_p} \tag{18}$$

The penalty matrix P is computed following [12], [13].

The constraints on the state are constant over the horizon except for the final position constraint. They can be described by bounds $q_{\min}, q_{\max}, \dot{q}_{\min}, \dot{q}_{\max}$ and $q_{N_{p_{\min}}}, q_{N_{p_{\max}}}$ defining admissible intervals.

In the following, the simulations are completed using the following parameters:

K_{sin}	K_{cos}	N_p	q_{ini} (rad)	$\dot{q}_{\text{ini}}$ (rad.s^{-1})	δ_t (s)
109	11.11	10	$-\pi$	0	0.010

The constraints are:

$[q_{\text{admissible}}]$ (rad)	$[\dot{q}_{\text{admissible}}]$ (rad.s^{-1})
$[-\pi - \frac{3\pi}{2}; -\pi + \frac{3\pi}{2}]$	[-50;50]

$[u_{\text{admissible}}]$ (m.s^{-2})	q_{N_p} (rad)
[-100;100]	[-0.2;0.2]

Simulations are computed using MATLAB on a PENTIUM IV 2GHz computer.

Fig. 4 displays the simulations results using NMPC scheme without reference trajectory. The pendulum starts vertically down ($-\pi$ rad) and is stabilized around the stable position (0 rad) reaching the final position domain $[q_{N_p}]$.

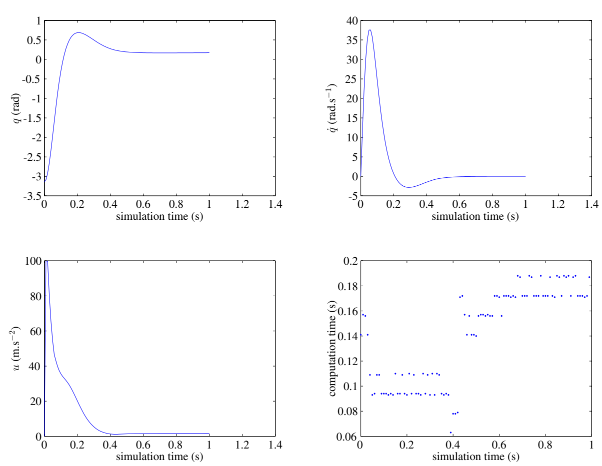

Fig. 4. Simulation results using NMPC

C. Interval analysis control

The standard forward-backward contractor applied for the pendulum is presented in algorithm 1. The considered CSP is defined by the dynamic equation and integration through first order Taylor series. At the end of the contraction procedure (lines 3 to 12), additional computation determine if the input is feasible *i.e* leads the system into admissible state (lines 13 to 23). We therefore estimate the state over the horizon and if one of the encountered states is not admissible, then the input is not feasible. In the other hand, if all the states are admissible, then the input is feasible. If the contraction procedure leads to a fixed point this one is *broken* by bisecting the domain of the input. As it is impossible to infinitely bisect the domains, we define the minimum width $\epsilon_{\text{bisection}}$ under which we will not bisect anymore.

$[u]$:= **contraction**($[u]$)

1 state_admissible ← false
2 **while** no fixed point AND no admissible_state **do**
  // Forward propagation over the horizon
3 **for** $k = 0 \dots (N_p - 1)$ **do**
4 $[\ddot{q}_{k+1}] = (-K_{\cos}\,[u]\,\cos([q_k]) + K_{\sin}\,\sin([q_k])) \cap [\ddot{q}_{k+1}]$
5 $[\dot{q}_{k+1}] = ([\dot{q}_k] + \delta_t\,[\ddot{q}_{k+1}]) \cap [\dot{q}_{k+1}]$
6 $[q_{k+1}] = ([q_k] + \delta_t\,[\dot{q}_{k+1}]) \cap [q_{k+1}]$
  // Backward propagation over the horizon
7 **for** $k = (N_p - 1) \dots 1$ **do**
8 $[q_k] = ([q_{k+1}] - \delta_t\,[\dot{q}_{k+1}]) \cap [q_k]$
9 $[\dot{q}_{k+1}] = \left(\dfrac{[q_{k+1}] - [q_k]}{\delta_t}\right) \cap [\dot{q}_{k+1}]$
10 $[\dot{q}_k] = ([\dot{q}_{k+1}] - \delta_t\,[\ddot{q}_{k+1}]) \cap [\dot{q}_k]$
11 $[\ddot{q}_{k+1}] = \left(\dfrac{[\dot{q}_{k+1}] - [\dot{q}_k]}{\delta_t}\right) \cap [\ddot{q}_{k+1}]$
12 **if** $\{0\} \notin \cos([q_i])$ **then**
  $[u] = \left(\dfrac{[\ddot{q}_{k+1}] - K_{\sin}\,\sin([q_k])}{-K_{\cos}\,\cos([q_k])}\right) \cap [u]$
  // Feasibility of the input
13 state_admissible ← true
14 $k \leftarrow 1$
15 **while** $k < N_p$ AND admissible_state **do**
16 $\ddot{\theta}_{k+1} = -K_{\cos}\,[u]\,\cos([\theta_k]) + K_{\sin}\,\sin([\theta_k])$
17 $\dot{\theta}_{k+1} = \dot{\theta}_k + \delta_t\,\ddot{\theta}_{k+1}$
18 **if** $\dot{\theta}_{k+1} \nsubseteq [\dot{q}_{\text{admissible}}]$ **then**
19 state_admissible ← false
20 $\theta_{k+1} = \theta_k + \delta_t\,\dot{\theta}_{k+1}$
21 **if** $\theta_{k+1} \nsubseteq [q_{\text{admissible}}]$ **then**
22 state_admissible ← false
23 $k \leftarrow k + 1$

Algorithm 1: Contraction for the pendulum control

The bisection introduces many other domains which must be contracted. The ordering of these candidates boxes is crucial and we define a criteria defining an ordering relation for the candidate boxes into a sorted list $\mathcal{L}$. Indeed, the aim of the algorithm is to compute a single punctual feasible input and not the biggest set of feasible inputs. Therefore computations are stopped as we find a single feasible domain in which we will select a single punctual feasible input.

The final computed input is the first feasible solution minimizing the ordering criteria, therefore the control algorithm (algorithm 2) is an optimization one. This algorithm is similar to those in [2], [14]. These algorithm are based on branch (recursive bisection of the variables) and bound strategy (estimation of the function extrema) and they mainly differ in the way they order the list of boxes to be bisected.

$[u_{\text{solution}}]$:= **control**($\mathcal{L}$)

- $[u_{\text{solution}}] \leftarrow \emptyset$
- **while** $\mathcal{L} \neq \emptyset$ AND $[u_{\text{solution}}] = \emptyset$ **do**
  - pop out the first element of $\mathcal{L}$ in $[u]$
  - $[u_{\text{contracted}}]$,input_feasible $\leftarrow$ contraction($[u]$)
  - **if** $[u_{\text{contracted}} \neq \emptyset]$ **then**
    - **if** input_feasible **then**
      - $[u_{\text{solution}}] \leftarrow [u_{\text{contracted}}]$
    - **else**
      - **if** diam($[u_{\text{contracted}}]$)$> \epsilon_{\text{bisection}}$ **then**
        - bisection of $[u_{\text{contracted}}]$ in $[u_1]$ and $[u_2]$
        - insert $[u_1]$ and $[u_2]$ in $\mathcal{L}$
      - **else**
        - // domain too small to be bisected AND input not feasible
        - // no operation
  - **else**
    - // no feasible solution
    - // no operation

Algorithm 2: Control with boxes ordering

D. Modified interval analysis control

The standard interval tools can be improved to be more efficient in the context of predictive control. In the following paragraph we focus on some important points that have been modified from the standard algorithm.

First of all, let's discuss about the fixed point. By definition, a fixed point is reached when there is no variation of any domain involved in the CSP. To avoid numerical problems when implementing this algorithm, we define the $\epsilon_{\text{contraction}}$ threshold. It represents the variation of the bounds of the input domain. If the input domain bounds are not modified by more than $\epsilon_{\text{contraction}}$ during the contraction procedure, then we consider being on a fixed point. This threshold can also be used in order to favor bisection over contraction.

Secondly, we had to make some additional computations after the forward-backward computations in order to check the feasibility of the input (algorithm 1, lines 13 to 23). These computations are the same as the propagation ones (algorithm 1, lines 3 to 6) except that they are combined with intersection of the previous definition of the domains. We propose to separate state estimation and intersection from the propagation computation in order to save computational effort.

Thirdly, as a consequence of our previous remark, the forward propagation is useless as long as the input domain is not modified. Therefore, the backward propagation is repeated as long as the input domain is not modified.

Finally, in order to reduce state estimation pessimism, the computations in the contraction are not done on interval values but on punctual values defining a spatial discretisation (section III-C).

$[u]$:= **modified_contraction**($[u]$)

- state_admissible $\leftarrow$ false;
- **while** no fixed point AND no state_admissible **do**
  - // Forward propagation over the horizon
  - **for** $k = 0 \ldots (N_p - 1)$ **do**
    - $[\ddot{q}_{k+1}] = -K_{\cos}\,[u]\,\cos([q_k]) + K_{\sin}\,\sin([q_k])$;
    - $[\dot{q}_{k+1}] = [\dot{q}_k] + \delta_t\,[\ddot{q}_{k+1}]$;
    - $[q_{k+1}] = [q_k] + \delta_t\,[\dot{q}_{k+1}]$;
  - // Intersection over the horizon
  - $k \leftarrow 1$;
  - **while** $t < N_p$ AND no empty set **do**
    - $[q_{k+1}] = [q_{k+1}] \cap [q_{\text{admissible}}]$;
    - $[\dot{q}_{k+1}] = [\dot{q}_{k+1}] \cap [\dot{q}_{\text{admissible}}]$;
    - $k \leftarrow k + 1$
  - **if** the intersection modified one of the $[q_k]$ or one of the $[\dot{q}_k]$ **then**
    - // the states are not admissible therefore the input is not feasible
    - **while** $[u]$ not modified more than $\epsilon_{\text{contraction}}$ AND no fixed point **do**
      - // Backward propagation
      - $[q_k] = ([q_{k+1}] - \delta_t\,[\dot{q}_{k+1}]) \cap [q_k]$;
      - $[\dot{q}_{k+1}] = \left(\dfrac{[q_{k+1}] - [q_k]}{\delta_t}\right) \cap [\dot{q}_{k+1}]$;
      - $[\dot{q}_k] = ([\dot{q}_{k+1}] - \delta_t\,[\ddot{q}_{k+1}]) \cap [\dot{q}_k]$;
      - $[\ddot{q}_{k+1}] = \left(\dfrac{[\dot{q}_{k+1}] - [\dot{q}_k]}{\delta_t}\right) \cap [\ddot{q}_{k+1}]$;
      - **if** $\{0\} \notin \cos([q_k])$ **then**
        - $[u] = \left(\dfrac{[\ddot{q}_{k+1}] - K_{\sin}\,\sin([q_k])}{-K_{\cos}\,\cos([q_k])}\right) \cap [u]$
  - **else**
    - // the intersection had no effect therefore the states are admissible and the input is feasible
    - state_admissible$\leftarrow$ true

Algorithm 3: Modified contraction for the pendulum control

Fig. 5 displays the results of a simulation using the same parameters as in section IV-B but with interval tools. The interval method has been improved following our remarks and without spatial discretisation. The parameters specific to interval techniques are $\epsilon_{\text{bisection}} = 0.01$ (m.s^{-2}) and $\epsilon_{\text{contraction}} = 0.01$ (m.s^{-2}). The vertical lines appearing in the input graphic represents the computed feasible domain. In order to select a punctual input in this domain, we apply the input of minimal norm. The minimal norm is also the ordering criteria for the list $\mathcal{L}$.

Fig. 6 displays the results of a simulation computed with the same parameters, but with spatial discretisation of the input in 10 samples. The computed input and therefore the behavior of the system is similar as without discretisation. The number of operations is dramatically improved, specially in the beginning of the simulation. However, the computation time is still not competitive with the NMPC method. This may be because of the use of non compiled interval routines in MATLAB. However, we can still improve the efficiency

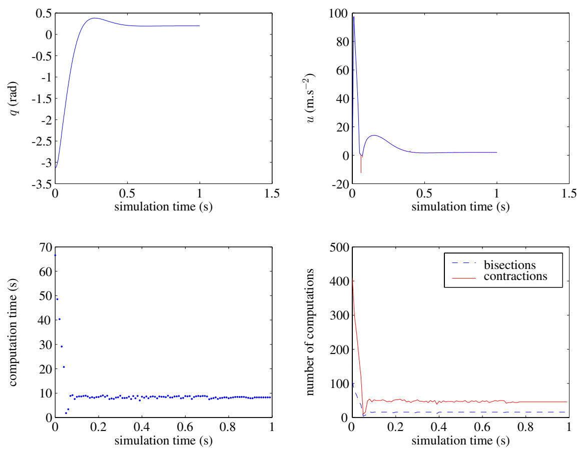

Fig. 5. Simulation results using improved interval techniques

of the interval method: i) we can adjust the threshold parameters. ii) we can look for an input in an area around the former input applied on the system. iii) we can also add at the beginning of the list $\mathcal{L}$ the punctual former input applied on the system. This reduces the optimality of the solution but reduces the number of computations required to find a feasible input. Fig. 7 displays simulation results using these last considerations and with $\epsilon_{\text{bisection}} = 3$ (m.s^{-2}) and $\epsilon_{\text{contraction}} = 0.5$ (m.s^{-2}).

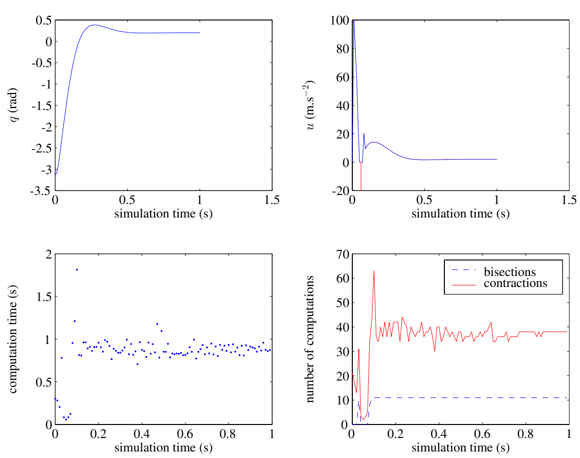

Fig. 6. Simulation results using improved interval techniques and spatial discretisation

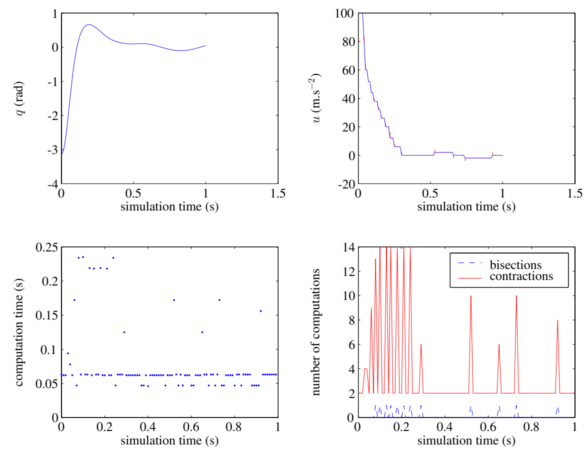

Fig. 7. Simulation results using improved interval techniques, spatial discretisation and previous input consideration

V. CONCLUSION

This paper introduces a non linear control approach associated with interval analysis, using NMPC without reference trajectory. The punctual values involved in the NMPC have been replaced by their interval counterparts, allowing the use of interval analysis methods. We exhibited the problems occurred when using interval analysis techniques to build a predictive control law. We presented the use of standard tools and described the problems appearing when using these tools in control context. We introduced several improvements: the introduction of a threshold defining the fixed point, as well as a modification of the forward-backward contractor to combine forward propagation and state estimation needed in model based control. In order to reduce pessimism of interval state estimation, we proposed a spatial discretisation of the domains. All these modifications lead to an efficient nonlinear model predictive controller via interval analysis.

REFERENCES

[1] F. Allgöwer, T. A. Badgwell, S. J. Qin, J. B. Rawlings, and S. J. Wright, "Nonlinear predictive control and moving horizon estimation - an introductory overview," in *Advances in Control: Highlights of ECC '99*, P. Frank, Ed. Springer-Verlag, 1999, ch. 12, pp. 391–449.

[2] L. Jaulin, M. Kieffer, O. Didrit, and E. Walter, *Applied Interval Analysis*. Springer, 2001.

[3] R. E. Moore, "Methods and applications of interval analysis," *Philadelphia, SIAM*, 1979.

[4] E. R. Hansen, *Global Optimization Using Interval Analysis*. New York, NY: Marcel Dekker, 1992.

[5] L. Jaulin, I. Braems, M. Kieffer, and E. Walter, "Interval methods for nonlinear identification and robust control," in *Proceedings of the IEEE Conference on Decision and Control (CDC)*, Las Vegas, Nevada, 2002.

[6] S. Malan, M. Milanese, and M. Taragna, "Robust analysis and design of control systems using interval arithmetic," *Automatica*, vol. 33, no. 7, pp. 1363–1372, 1997.

[7] F. Benhamou, F. Goualard, L. Granvilliers, and J.-F. Puget, "Revising hull and box consistency," in *Proceedings of the International Conference on Logic Programming*, Las Cruces, 1999, pp. 230–244.

[8] N. S. Nedialkov, "Computing rigorous bounds on the solution of an initial value problem for an ordinary differential equation," Ph.D. dissertation, Computer Science Dept., Univ. of Toronto, 1999.

[9] T. Raïssi, N. Ramdani, and Y. Candau, "Guaranteed stated estimation for nonlinear continuous systems with Taylor models," in *Proceedings of the IFAC Symposium on System Identification (SYSID)*, 2002.

[10] K. J. Åström and K. Furuta, "Swinging up a pendulum by energy control," *Automatica*, vol. 36, pp. 287–295, 2000.

[11] L. Magni, R. Scattolini, and K. J. Åström, "Global stabilization of the inverted pendulum using model predictive control," in *Proceedings of the 15th IFAC World Congress*, Barcelona, 2002.

[12] H. Chen and F. Allgöwer, "A quasi-infinite horizon nonlinear model predictive control scheme with guaranteed stability," *Automatica*, vol. 34, no. 10, pp. 1205–1217, 1998.

[13] P. Scokaert, D. Mayne, and J. Rawlings, *Suboptimal Model Predictive Control (Feasibility Implies Stability)*. IEEE Trans. Automat. Contr., 1999, vol. 44(3).

[14] A. P. Leclerc, "Efficient and reliable global optimization," Ph.D. dissertation, The Ohio State University, 1992.