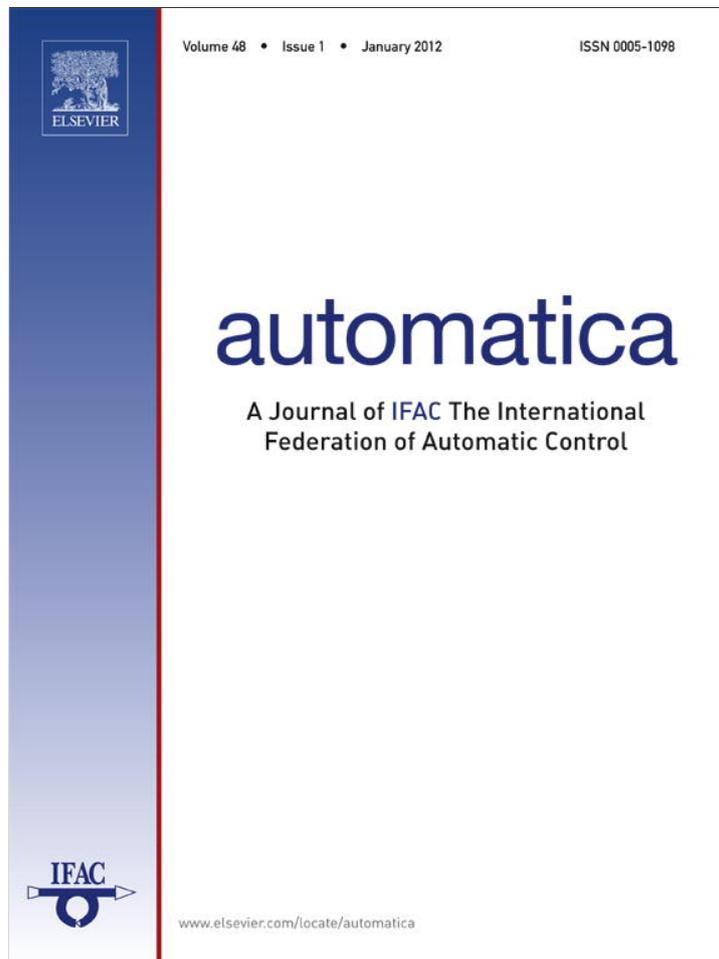


Provided for non-commercial research and education use.  
Not for reproduction, distribution or commercial use.



This article appeared in a journal published by Elsevier. The attached copy is furnished to the author for internal non-commercial research and education use, including for instruction at the authors institution and sharing with colleagues.

Other uses, including reproduction and distribution, or selling or licensing copies, or posting to personal, institutional or third party websites are prohibited.

In most cases authors are permitted to post their version of the article (e.g. in Word or Tex form) to their personal website or institutional repository. Authors requiring further information regarding Elsevier's archiving and manuscript policies are encouraged to visit:

<http://www.elsevier.com/copyright>



Contents lists available at SciVerse ScienceDirect

Automatica

journal homepage: [www.elsevier.com/locate/automatica](http://www.elsevier.com/locate/automatica)

Brief paper

# A framework for real-time implementation of low-dimensional parameterized NMPC<sup>☆</sup>

Mazen Almir<sup>1</sup>

GIPSA-LAB/CNRS, University of Grenoble, Rue des Mathématiques, 38400 Saint Martin d'Hères, France

## ARTICLE INFO

## Article history:

Received 28 May 2010

Received in revised form

9 May 2011

Accepted 1 July 2011

Available online 6 October 2011

## Keywords:

NMPC

Fast implementation

Parameterized approach

Real-time

PVTOL stabilization

Convergence analysis

## ABSTRACT

In this paper, a novel approach is proposed to implement low-dimensional parameterized Nonlinear Model Predictive Control (NMPC) schemes for systems showing fast dynamics. The proposed scheme is based on distributing the reconstruction of the cost function over the real lifetime of the controlled system. The framework is particularly suitable for NMPC formulations that use low dimensional control parametrization. The concrete example of a Planar Vertical Take-Off and Landing (PVTOL) aircraft stabilization problem is used to illustrate the efficiency of the proposed formulation.

© 2011 Elsevier Ltd. All rights reserved.

## 1. Introduction

One of the major challenging issues in modern nonlinear control is the one related to the application of Nonlinear Model Predictive Control (NMPC) schemes (Mayne, Rawlings, Rao, & Scokaert, 2000) to nonlinear systems showing fast dynamics. Indeed, for such systems, the number of iterations that may be needed to reach a sufficiently good solution to the underlying open-loop optimal control problem may be beyond what is reasonably possible to achieve in the available computation time. Since the early work (Almir, 2001) that first suggested the idea of distributing the computation during the real system lifetime, this idea is now widely shared in the NMPC community and many works emerged that are related to the conceptual issues (Almir, 2006; DeHaan & Guay, 2007; Diehl, Bock, & Schloöder, 2005; Ohtsuka, 2004) as well as to successful applications (Almir, 2006; Almir et al., 2009; Ortner, Bergmann, Ferreau, & del Re, 2009; Shimizu, Ohtsuka, & Diehl, 2006) to cite a few works.

This paper proposes a novel solution that addresses the above problem for a class of NMPC formulations. The basic idea is to update a moving horizon quadratic approximation of the cost function as a function of both the state and the control parameters. A particularly appealing feature is that a high number of realizations of the cost function are computed using a relatively low number of

<sup>☆</sup> This paper was not presented at any IFAC meeting. This paper was recommended for publication in revised form by Associate Editor Martin Guay under the direction of Editor Frank Allgöwer.

E-mail address: [mazen.almir@grenoble-inp.fr](mailto:mazen.almir@grenoble-inp.fr).

<sup>1</sup> Tel.: +33 476826326; fax: +33 476826388.

system simulations. In many aspects, the proposed scheme follows the lines suggested in a series of works of Conn and co-workers (see for instance Conn, Gould, and Toint (2006)) who addressed the problem of free-derivative optimization in the particular case where the computation of the cost function is particularly expensive.

The paper is organized as follows: First, the principle of real-time parameterized NMPC is recalled in Section 2 which clearly states the class of NMPC formulations that are covered by the proposed approach. The proposed solution is presented in Section 3 together with some convergence analysis results. The efficiency of the proposed solution is investigated in Section 4 through an illustrative example. Finally Section 5 concludes the paper and gives some hints for future work.

## 2. Recalls on parameterized NMPC

Let us consider discrete-time nonlinear systems given by:

$$x(k+1) = f(x(k), u(k)) \quad (1)$$

where  $x \in \mathbb{R}^n$ ,  $u \in \mathbb{R}^m$  denote the state and the control vectors respectively. Classical NMPC is defined by computing at each decision instant  $k$ , an optimal sequence of future control actions:

$$\mathbf{u}(k) = (u^T(k), \dots, u^T(k+N_p-1))^T \in \mathbb{U}(x(k)) \quad (2)$$

that minimizes some cost function depending on the current state  $x(k)$ . The first control in the optimal sequence  $\hat{\mathbf{u}}(k) = (\hat{\mathbf{u}}^{(0)}(k), \dots, \hat{\mathbf{u}}^{(N_p-1)}(k))^T$ , namely  $\hat{\mathbf{u}}^{(0)}(k)$  is applied during the time interval  $[k, k+1]$  at the end of which the whole process is

repeated leading to an implicit state feedback  $u(k) = K(x(k)) = \hat{\mathbf{u}}^{(0)}(k) \in \mathbb{R}^n$ . The parameterized NMPC formulations on which this paper is focused are those in which the admissible set  $\mathbb{U}(x) = \mathbb{U}$  is state independent and defined by a low dimensional parameter vector  $p \in \mathbb{R}^{n_p}$ , namely:

$$\mathbb{U} := \{\mathbf{u} \mid \mathbf{u}^{(i)} = \mathcal{U}(i, p) \text{ where } p \in \mathbb{P}\} \quad (3)$$

for some predefined parametrization map  $\mathcal{U}$  that structurally incorporates control related constraints. Consequently, the cost function can be expressed as a function  $J(x, p)$  of the initial state and the parameter vector. At each  $k$ , the optimal parameter vector  $p^*(x(k)) := \arg \min_{p \in \mathbb{P}} J(x(k), p)$  is computed and the corresponding control  $\mathcal{U}(0, p^*(x(k)))$  is applied on  $[k, k + 1]$  and so on.

**Remark 1.** The fact that the admissible set  $\mathbb{U}$  is state independent implicitly means that all potential state related constraints are taken into account (through appropriate barrier-like weighting terms) in the definition of the cost function. It is assumed that the resulting formulation (when computed exactly regardless real-time constraints) leads to closed-loop stability that is based on the following inequality expressing the decrease of the cost function:

$$J^*(x(k^+)) - J^*(x(k)) \leq -W(x(k)) \quad (4)$$

where  $J^*(x) := J(x, p^*(x))$  while  $W$  is some positive definite function. This contribution focuses on real-time implementation of the resulting state constraints-free formulation. Consequently, there is no underlying feasibility issue to be addressed.

When systems with fast dynamics are considered, short sampling time is needed and it is no longer possible to consider that the optimal solution  $p^*(x(k))$  can be obtained in a small fraction of the sampling period. Instead, the parameter vector  $p$  becomes an internal state of the controller that has to be updated conveniently so as to hopefully recover the above ideal situation asymptotically. An extended dynamical system is then obtained that can be formally written as follows:

$$x(k + 1) = \mathcal{F}(x(k), p(k)) := f(x(k), \mathcal{U}(0, p(k))) \quad (5)$$

$$p(k + 1) = \mathcal{S}(p(k), x(k), \Theta(k)) \quad (6)$$

$$\Theta(k + 1) = \mathcal{D}(p(k), x(k), \Theta(k)) \quad (7)$$

where  $\Theta$  gathers the internal state of the updating algorithm while  $\mathcal{S}$  and  $\mathcal{D}$  are updating functions that have to be defined which is precisely the aim of this paper.

In the sequel, the notation  $\mathbf{x}^{(i)}(x, p)$  denotes the solution at instant  $i$  of (1) under the control sequence defined by  $p$  and starting from the initial state  $x(0) = x$ .

### 3. The proposed real-time framework

First of all, it is assumed that the following property is satisfied (Alamir, 2006):

**Assumption 1.** The parametrization map  $\mathcal{U}$  is such that for all  $p \in \mathbb{P}$ , there is a parameter value  $p^+ \in \mathbb{P}$  such that:

$$\forall i \in \mathbb{N} \quad \mathcal{U}(i + 1, p) = \mathcal{U}(i, p^+). \quad (8)$$

This is in particular satisfied as soon as exponential parametrization is involved (see Alamir (2006) and the illustrative example of Section 4). Moreover, we assume throughout the paper that the admissible parameter set  $\mathbb{P}$  is a hypercube in  $\mathbb{R}^{n_p}$ . In the sequel, the following notation is used to designate successive applications of the operator (+) invoked in Assumption 1:

$$p^{0+} = p; \quad p^{i+} = [p^{(i-1)+}]^+ \quad (9)$$

Similarly, given a pair  $z = (p, x)$ , the compact notation  $z^+ = (p^+, x^+)$  where  $x^+ = \mathcal{F}(x, p)$  will be used together with the following multi-step notation:

$$z^{0+} = z; \quad z^{i+} = [z^{(i-1)+}]^+ \quad (10)$$

Note that by using this notation, the cost function becomes a function of the variable  $z$ . Based on the above, the following fact that lies in the heart of the proposed approach can be emphasized:

**Fact 1.** Assume that a cost function  $J$  is defined using a prediction horizon  $N_p$ . Given a pair  $z = (p, x)$ , the computation of the corresponding system's trajectory over a horizon of length  $(N_p + r)$  provides  $(r + 1)$  realizations of the function  $J(z)$ , namely  $\{J(z^{i+})\}_{i=0}^r$  where  $z^{i+}$  is the pair defined by (10).  $\square$

**Fact 2.** Consider the system at instant  $k - 1$  with the state  $x(k - 1)$  and the parameter  $p(k - 1)$ . For any given discrete set  $\mathbb{A}(k)$  containing  $n_{\max}$  elements of  $\mathbb{P}$ , one can compute the  $((r + 1) \cdot n_{\max})$  realizations of the cost function  $\{J(z)\}_{z \in \mathbb{Z}(k)}$  where  $\mathbb{Z}(k)$  is the set defined by:

$$\mathbb{Z}(k) := \{(\hat{x}(k), p)^{i+} \mid (p, i) \in \mathbb{A}(k) \times \{0, \dots, r\}\} \quad (11)$$

in which  $\hat{x}(k) := \mathcal{F}(x(k - 1), p(k - 1))$  is the predicted state at the beginning of the next sampling period. Moreover, the computation of these  $((r + 1) \cdot n_{\max})$  realizations needs only  $n_{\max}$  system simulations over a prediction horizon of length  $N_p + r$ .  $\square$

The idea is to compute a moving-horizon quadratic approximation of  $J(z)$  based on the data:

$$\Theta(k) := \{D(k), \dots, D(k - N_m + 1)\} \quad (12)$$

in which

$$D(k) := \{Z(k), \{J(z)\}_{z \in Z(k)}\} \quad (13)$$

Note that  $\Theta$  is precisely the internal state that is involved in (6)–(7). Note also that the data contained in  $\Theta(k)$  involves those pairs of  $z$ 's including  $(\hat{x}(k - j), p)^{i+}$  where:

$$(i, j, p) \in \{0, \dots, r\} \times \{0, \dots, N_m - 1\} \times \mathbb{A}(k - j). \quad (14)$$

This represents  $n_\theta := (r + 1) \cdot N_m \cdot n_{\max}$  realizations of the cost function in the variable  $z$ . Remember that these realizations can be used during each sampling period while only  $n_{\max}$  new system simulations are performed.

Based on (12), the updating law  $\mathcal{D}$  for  $\Theta$  is straightforward: at instant  $k + 1$ , drop the oldest data  $D(k - N_m + 1)$  and use the newest data  $D(k + 1)$ . The fact that according to (7),  $D(k + 1)$  depends on  $p(k)$  is due to the definition of the set  $\mathbb{A}(k + 1)$  (see (18) hereafter).

The data contained in  $\Theta(k)$  can be used to compute a quadratic approximation of the cost function:

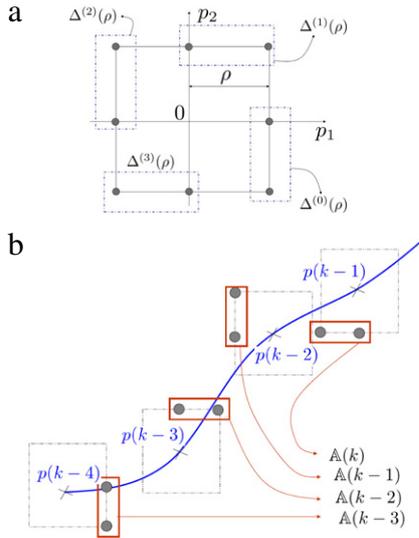
$$\hat{J}_k(z) = J_k^0 + L_k^T z + z^T Q_k z =: \Phi(z) \cdot q \quad (15)$$

where  $Q_k$  is nonnegative symmetric. Note that  $q$  is the vector of the coefficients gathering  $J_k^0$ , the coefficients of  $L_k$  and the upper-diagonal coefficients of  $Q_k$  involved in (15). Consequently, the quadratic approximation involves  $n_q = (1 + n_z)(1 + n_z/2)$  unknowns that are gathered in a single vector  $q(k)$ . The value of  $q$  is computed by minimizing the following least squares criterion:

$$q(k) := \arg \min_q [\| [A(\Theta(k))]q - B(\Theta(k)) \|^2 + \varepsilon \cdot \|q - q(k - 1)\|^2] \quad (16)$$

where  $\varepsilon > 0$  is a regularization coefficient. Note that  $A(\Theta)$  and  $B(\Theta)$  are obtained by concatenating the set of linear equations representing (15) for all  $z$ 's contained in the data set  $\Theta(k)$ . Note that because of the structure of  $\Theta(k)$  that incorporates newly acquired data at each sampling period, it is straightforward that (16) can be solved using recursive least squares in which a forgetting factor  $\mu \in ]0, 1[$  is used. This leads to the following fact:

**Fact 3.** The computational burden that is related to the identification of the cost function corresponds to the solution, at each sampling period, of an unconstrained linear least squares of dimension  $n_{ls} := (r + 1) \cdot n_{\max} + n_q$  unknowns.  $\square$



**Fig. 1.** Definition of the sets  $\mathbb{A}(k)$ : (a) Example of subsets  $\Delta^{(j)}(\rho)$  in the particular case:  $N_m = 4$ ,  $n_p = 2$  and  $n_{\max} = 2$ . (b) The corresponding subset  $\mathbb{A}(k)$ .

### 3.1. Definition of parameter set $\mathbb{A}(k)$

Let us define for each  $\rho > 0$  a discrete subset  $\Delta(\rho) \subset \mathbb{R}^{n_p}$  such that the following two conditions hold;

- (1)  $\Delta(\rho) = \{\delta_1(\rho), \dots, \delta_{n_\Delta}(\rho)\}$  where  $\delta_i(\rho) \in \mathbb{R}^{n_p}$ .
- (2) the matrix  $[\delta_1(\rho), \dots, \delta_{n_\Delta}(\rho)]$  has full rank  $n_p$ .

Moreover, the  $\delta_i(\rho)$  are chosen on the edge of the hypercube centered at 0 and of length  $\rho$ .

Once  $\Delta(\rho)$  is chosen, let us define  $N_m$  discrete sets  $\Delta^{(j)}(\rho)$ ,  $j \in \{0, \dots, N_m - 1\}$  containing each  $n_{\max}$  elements of  $\Delta(\rho)$  and such that:

$$\Delta(\rho) = \bigcup_{j=0}^{N_m-1} \Delta^{(j)}(\rho) \quad (17)$$

with the trivial condition  $n_p \leq n_\Delta \leq N_m \cdot n_{\max}$ . Fig. 1 shows an example of such subsets when  $n_p = 2$ ,  $n_\Delta = 8$ ,  $n_{\max} = 2$  and  $N_m = 4$ . Now, given the subsets  $\Delta^{(j)}(\cdot)$  and the current value  $\rho(k-1)$  of  $\rho$ , the definition of the subsets  $\mathbb{A}(k)$  is given by:

$$\mathbb{A}(k) := Pr(p^+(k-1) + \Delta^{(j(k-1))}(\rho(k-1)), \mathbb{P}) \quad (18)$$

$$j(k) := (j(k-1) + 1) \bmod (N_m) \quad (19)$$

where for a discrete subset  $\mathbb{D} \subset \mathbb{R}^{n_p}$ , the notation  $Pr(\mathbb{D}, \mathbb{P})$  denotes the discrete set obtained by projecting all the elements of  $\mathbb{D}$  on the hypercube  $\mathbb{P}$ . In the next section, it is shown how  $\rho$  is updated by implementing a trust-region like iteration.

### 3.2. Trust region updating law of $\rho$

Remember that  $\rho$  controls the size of the neighborhood of the current value of  $p$  over which the search for a better value is done based on the current quadratic approximation of the cost function. Note that on one hand,  $\rho$  must be sufficiently high in order to enhance rapid decrease of the distributed-in-time algorithm. But on the other hand, small values of  $\rho$  might be required in order for the quadratic approximation to be relevant. Based on the above discussion,  $\rho$  is increased if the quadratic approximation is efficient at decreasing the cost function while it is decreased in case of failure.

More precisely, the following Quadratic Programming (QP) problem is first solved to obtain the candidate value  $p_c(k)$ :

$$p_c(k) := \arg \min_{p \in \mathcal{P}(k)} [\hat{J}_k(\hat{x}(k), p)] \quad (20)$$

where  $\mathcal{P}(k)$  is the hypercube given by:

**Table 1**

Computation times needed to achieve the tasks involved in the proposed scheme. These times are taken to be the worst among 1000 random trials as long as the least squares and the QP times are concerned. For the 4th order Runge–Kutta simulation times, the results are obtained for the example system studied in Section 4 hereafter. Evaluation has been done for Matlab 2011a software running on a Mac OS X platform with 8GB-RAM and a 2.3 GHz Intel Core i7 processor. (Remember that  $n_{\max} + 1$  system simulations are necessary.)

Computation times (ms)	
Least squares	15
QP	20
RK4 ( $N_p = 150$ )	18
Total ( $n_{\max} = 1 \rightarrow$ ) 2 simulations	71

$$\mathcal{P}(k) := \text{Conv}\{Pr(p^+(k-1) + \Delta(\rho(k-1)), \mathbb{P})\} \quad (21)$$

while  $\hat{J}(\hat{x}(k), p)$  is the value of the identified quadratic approximation of  $J$  at  $(\hat{x}(k), p)$ . Note that (20) is a quadratic programming problem of dimension  $n_p$  since  $x$  is fixed to  $\hat{x}(k)$ . Once the candidate value  $p_c(k)$  is obtained, the true corresponding cost is computed by simulating the system using the pair  $(\hat{x}(k), p_c(k))$  to obtain  $J_c := J(\hat{x}(k), p_c(k))$ . The quadratic approximation is declared relevant if the following inequality holds:

$$J_c < \min\{J(\hat{x}(k), p) \mid p \in \mathbb{A}(k)\}. \quad (22)$$

This leads to the following updating law for  $\rho$ :

$$\rho(k) := \begin{cases} \beta^+ \cdot \rho(k-1) & \text{if (22) is satisfied} \\ \beta^- \cdot \rho(k-1) & \text{otherwise} \end{cases} \quad (23)$$

where  $\beta^+ \geq 1$  and  $\beta^- \in [0, 1]$  are respectively the expansion and the contraction factors. Finally, the updating law  $\mathcal{J}$  for  $p$  that has been invoked in (6) is given by:

$$p(k) := \begin{cases} p_c(k) & \text{if (22) is satisfied} \\ p^+(k-1) & \text{otherwise} \end{cases} \quad (24)$$

where  $p^+(k-1)$  is the translated value of  $p(k-1)$  that is invoked in Assumption 1. This enables consistency to be enhanced. Note that the updating law (24) is of the form (6) since condition (22) depends on  $\hat{x}(k) = \mathcal{F}(x(k-1), p(k-1))$ .

### 3.3. Computational burden

The computational effort during a single sampling period is mainly due to the following operations:

- Solving the resulting linear system of equations of dimension  $n_{ls} := n_{\max} \cdot (r+1) + n_q$  in  $n_q$  unknowns.
- Solving the QP problem (20) with  $(n_p)$  decision variables and  $(2n_p)$  constraints
- Simulating the system model  $n_{\max} + 1$  times to update  $\Theta(k)$

Table 1 shows the computation times needed to perform the main tasks involved in the proposed algorithm with Matlab 2011a running on a Mac OS X platform with 8GB-RAM and a 2.3 GHz Intel Core i7 processor. This clearly shows that the proposed algorithm would be compatible with current real-time platforms as far as the proposed example is concerned.

### 3.4. Convergence analysis

In this section, some results concerning the convergence of the proposed framework are given. First of all, the following definition is needed:

**Definition 1.** A function  $F$  defined on  $\mathbb{R}^n \times \mathbb{R}^{n_p}$  is said to be strictly convex in  $p$  uniformly in  $x$  iff the following conditions hold:

- $F(x, \cdot)$  is strictly convex for all  $x$
- There is a constant  $\kappa$  such that for all  $x$ , one has:

$$\left\| \frac{\partial F}{\partial p}(x, p) \right\| \geq \kappa \cdot \|p - p^*(x)\| \quad (25)$$

where  $p^*(x)$  is the global minimum of  $F(x, \cdot)$ .

Let us first consider the following lemma:

**Lemma 3.1.** *In the case where the dynamic system (1) is obtained by discretizing a continuous ODE with a sampling period  $\tau > 0$ . For all  $n_{\max} \geq 1$  and provided that  $\varepsilon = 0$  is used in (16), if the following condition holds over the closed loop trajectory:*

$$\text{span} \left\{ \bigcup_{j=1}^{N_m} \mathbb{A}(k-j) \right\} = \mathbb{R}^{n_p} \quad (26)$$

the following inequality holds over the closed-loop system trajectory:

$$J(x(k), p(k)) = \hat{J}_k(x(k), p(k)) + O(\rho^3(k), \tau) \quad (27)$$

where  $O(\cdot, \cdot)$  vanishes when its two arguments vanish.

**Proof.** Note that when  $\tau \rightarrow 0$ , all the instantiations of  $J$  contained in the date set  $\Theta(k)$  are obtained for roughly the same value of  $x$  since the differences between the  $N_m$  successive states tend to 0. Moreover, thanks to (26), the identification is done using a regular set of values of  $p$ . Since any function is locally quadratic, the result follows for  $\tau \rightarrow 0$ . This with the definition of the term  $O(\rho^3(k), \tau)$  ends the proof.  $\square$

Based on Lemma 3.1, the following result can be stated:

**Lemma 3.2.** *Assume that*

- (1) *the dynamic system (1) is obtained by discretizing a continuous ODE with a sampling period  $\tau > 0$*
- (2) *the original system shows no finite escape time*
- (3) *the cost function  $J(x, \cdot)$  is strictly convex in  $p$  uniformly in  $x$  in the sense of Definition 1*
- (4)  *$p$  and  $p^+$  invoked in Assumption 1 are such that:  $\|p^+ - p\| = O(\tau)$*
- (5)  $\mathbb{P} = \mathbb{R}^{n_p}$ .

Then the following inequality holds over the closed-loop system trajectory:

$$\hat{J}_{k^+}(x(k^+), p(k^+)) \leq \hat{J}_k(x(k), p(k)) - K_1(\|\Delta p(k^+)\|) \cdot \rho(k^+) + O(\rho^2(k^+), \tau) \quad (28)$$

where  $k^+ = k + 1$  and

- $K_1, K_2$  are continuous positive definite.
- $\Delta p(k^+) := p^+(k) - p^*(x(k^+))$
- $p^*(x)$  is the minimizer of  $J(x, \cdot)$  over  $\mathbb{P}$

**Proof.** Since  $J$  is supposed to be strictly convex in  $p$  uniformly in  $x$ , for sufficiently small  $\rho(k)$ , it happens that the resulting local quadratic approximation  $\hat{J}_k$  inherits this property. Now since  $p(k^+)$  is obtained by minimizing  $J(x(k^+), \cdot)$  over a hypercube  $\mathcal{P}(k^+)$  of size  $\rho(k^+)$  centered at  $p^+(k)$  [see (20)–(21)], it can be inferred from the strict uniform convexity of  $\hat{J}_{k^+}$  that the following inequality holds:

$$\hat{J}_{k^+}(x(k^+), p(k^+)) \leq \hat{J}_k(x(k^+), p^+(k)) - K_1(\|p^+(k) - p^*(x(k^+))\|) \cdot \rho(k^+) + O(\rho^2(k^+), \tau) \quad (29)$$

for some continuous positive definite function  $K_1$ . This is because far from  $p^*(x(k^+))$ , the hypercube  $\mathcal{P}(k^+)$  necessarily contains a descent direction for the strictly uniformly convex function  $\hat{J}_{k^+}$ , moreover this descent direction is admissible for the optimization problem (20)–(21) since  $\mathbb{P} = \mathbb{R}^{n_p}$ . This leads to a decrease with respect to the initial guess  $p^+(k)$  (the center of the hypercube of

search) that is a positive definite function of the hypercube size  $\rho(k^+)$ .

Now using the two following facts (which are implied by the assumptions (1) and (4) of the current lemma):

$$\|p^+ - p\| = O(\tau); \quad \|x(k) - x(k^+)\| = O(\tau) \quad (30)$$

together with (29) clearly gives (28).  $\square$

The above two lemmas enables the main result to be stated:

**Proposition 3.1.** *Under the conditions of Lemmas 3.1 and 3.2, the use of a constant trust region radius  $\rho(k) = \rho$  [namely  $\rho(0) = \rho, \beta^+ = \beta^- = 1$ ] leads to a closed-loop trajectory satisfying the following asymptotic property:*

$$\lim_{k \rightarrow \infty} \|p(k) - p^*(x(k))\| = O(\rho, \tau). \quad (31)$$

Consequently, if the ideal parametrized formulation (without distributing the computation over time) asymptotically stabilizes the origin  $x = 0$  then the proposed formulation leads to a practical stability with a final attractive region with a radius that is of the form  $O(\rho, \tau)$ .

**Proof.** First of all, note that (31) is a direct consequence of (29) in which (27) is used for  $k$  and  $k^+$ . Indeed, merging the above equations leads to:

$$J(x(k^+), p(k^+)) \leq J(x(k), p(k)) - K_1(\|p^+(k) - p^*(x(k^+))\|) \cdot \rho(k^+) + O(\rho^2(k^+), \tau) \quad (32)$$

and since  $\rho(k^+) = \rho$ , one obtains:

$$J(x(k^+), p(k^+)) \leq J(x(k), p(k)) - K_1(\|p^+(k) - p^*(x(k^+))\|) \cdot \rho + O(\rho^2, \tau) \quad (33)$$

therefore  $p^+(k)$  converges to a neighborhood of  $p^*(x(k^+))$  of size  $O(\rho, \tau)$  and since  $\|p(k^+) - p^+(k)\| \leq \rho$ . This clearly gives (31).

To prove the remaining part of the proposition, it suffices to note that under the assumption of asymptotic stability in the ideal case, one would have:

$$J^*(x(k^+)) - J^*(x(k)) \leq -W(x(k)) \quad (34)$$

where  $J^*(x) := J(x, p^*(x))$  while  $W$  is some positive definite function. Now using (31), some classical continuity arguments together with the invariance principle leads to:

$$\lim_{k \rightarrow \infty} W(x(k)) = O(\rho, \tau)$$

which clearly ends the proof by the continuity and the positive definiteness of  $W$ .  $\square$

#### 4. Illustrative example

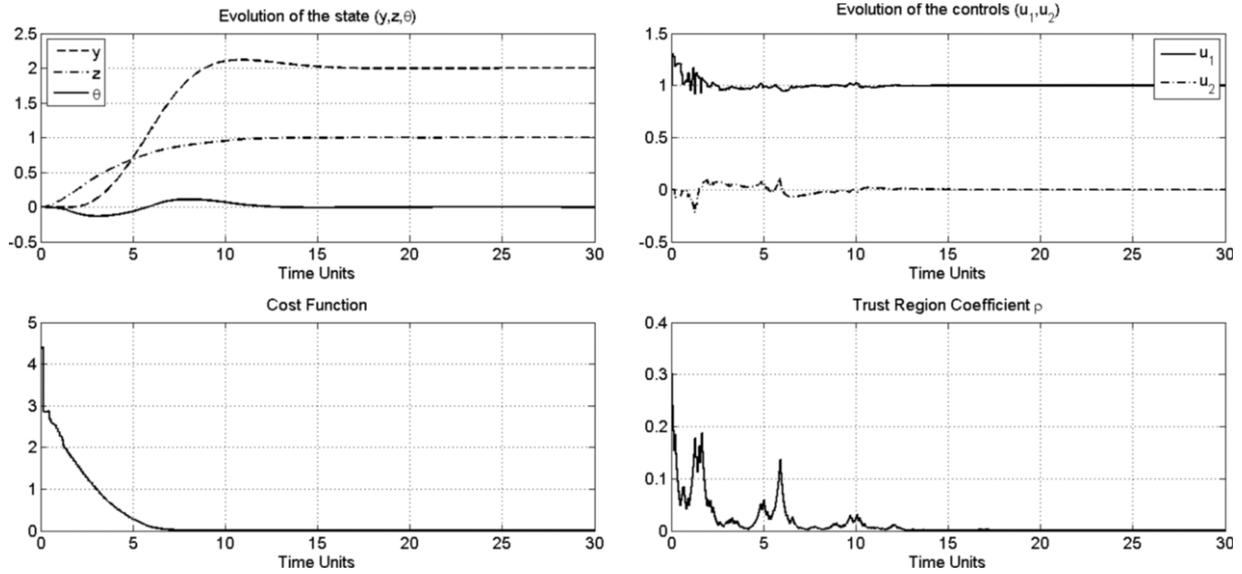
Consider the PVTOL aircraft model (Martin, Devasia, & Paden, 1994) given by:

$$\ddot{y} = -u_1 \sin \theta + \epsilon u_2 \cos \theta \quad (35)$$

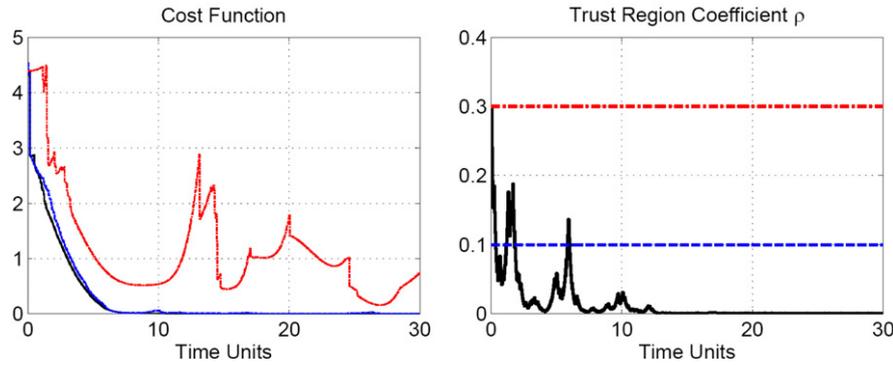
$$\ddot{z} = u_1 \cos \theta + \epsilon u_2 \sin \theta - 1 \quad (36)$$

$$\ddot{\theta} = u_2 \quad (37)$$

where  $\theta$  is the roll angle,  $(y, z)$  is the plane on which the aircraft movement is projected with  $z$  being the vertical coordinate.  $u_1$  and  $u_2$  stand for the lift and the roll angle acceleration respectively. The system is discretized using a sampling period  $\tau = 100$  ms. The control input has to satisfy the following saturation constraints  $u_1 \in [0, u_1^{\max}]$ ;  $u_2 \in [-u_2^{\max}, +u_2^{\max}]$  where  $u_1^{\max} > 0$  and  $u_2^{\max} > 0$  are some given positive upper bounds. Note that for obvious feasibility reasons, the upper bound  $u_1^{\max}$  must be sufficiently high to compensate the normalized gravity, namely  $u_1^{\max} = 1 + \gamma$  with



**Fig. 2.** Evolution of the closed-loop system under the proposed real-time NMPC schemes using the set of parameters  $n_{\max} = 1$ ,  $N_m = 6$  and  $r = 5$ . The trust region algorithm is on with  $\beta^+ = 1.2$  and  $\beta^- = 0.8$ .



**Fig. 3.** Illustration of the efficiency of the trust region adaptation mechanism of  $\rho$ . (Black): the same results as Fig. 2. (Blue-dashed): no trust region adaptation (i.e.  $\beta^+ = \beta^- = 1$ ) with the initial value  $\rho = 0.1$ . (Red-dot-dashed): no trust region adaptation with the initial value  $\rho = 0.3$  for the size of the investigated hypercube. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

$\gamma > 0$ . The control objective is to stabilize the system around  $x^d = (y^d, z^d, 0, 0, 0, 0) \in \mathbb{R}^6$  using admissible controls. The parametrization map is given by

$$\mathcal{U}(i, p) = \text{Sat}_{[0, u_1^{\max}] \times [-u_2^{\max}, +u_2^{\max}]} [\mathcal{U}^{\text{unc}}(i, p)] \quad (38)$$

$$\mathcal{U}^{\text{unc}}(i, p) := \begin{pmatrix} p_1 e^{-\lambda(i\tau)} + p_2 e^{-3\lambda(i\tau)} + p_3 e^{-6\lambda(i\tau)} \\ p_4 e^{-\lambda(i\tau)} + p_5 e^{-3\lambda(i\tau)} + p_6 e^{-6\lambda(i\tau)} \end{pmatrix}.$$

Note that Assumption 1 is satisfied with the following definition of  $p^+$ :

$$p^+ = \text{diag}(e^{-\lambda\tau}, e^{-3\lambda\tau}, e^{-6\lambda\tau}, e^{-\lambda\tau}, e^{-3\lambda\tau}, e^{-6\lambda\tau}) \cdot p.$$

Note that this expression clearly satisfies assumption (4) of Lemma 3.2. The admissible set of parameters is taken as follows:

$$\mathbb{P} := [0, u_1^{\max}]^2 \times [-u_2^{\max}, u_2^{\max}]^2. \quad (39)$$

The cost function  $J(x, p)$  is defined by:

$$J(x, p) := \sum_{i=0}^{N_p} [\|x^{(i)}(x, p) - x^d\|_{Q_x}^2 + \|\mathcal{U}(i, p, x) - u^d\|_{R_u}^2] \\ \times \|x^{(N_p)}(x, p) - x^d\|_{Q_N}$$

where  $u^d = (0, 1)^T$ . In the forthcoming simulations, the weighting matrices  $Q_x = \text{diag}(1, 1, 1, 10^{-2}, 10^{-2}, 10^{-2})$ ,  $R_u = 10^{-7} \cdot \mathbb{I}_2$

and  $Q_N = 0$  are used ( $\mathbb{I}_2$  denotes the identity matrix in  $\mathbb{R}^{2 \times 2}$ ). The minimum number of elements  $n_\Delta = n_p = 6$  is used with the set  $\Delta(\rho)$  defined by:  $\Delta(\rho) := \{\rho \cdot e_i\}_{i=1}^6$  where  $e_i$  is the  $i$ -element of the orthogonal basis of  $\mathbb{R}^6$ .

Fig. 2 shows the closed-loop behavior when the parameters  $(n_{\max}, N_m, r) = (1, 6, 5)$  are used. Note that this corresponds to 2 system simulations at each sampling period and satisfies the constraints  $N_m \cdot n_{\max} \geq n_p = 6$ . This is the basic scenario that is used in Figs. 3–5 to illustrate the effect of different choices on the resulting performance.

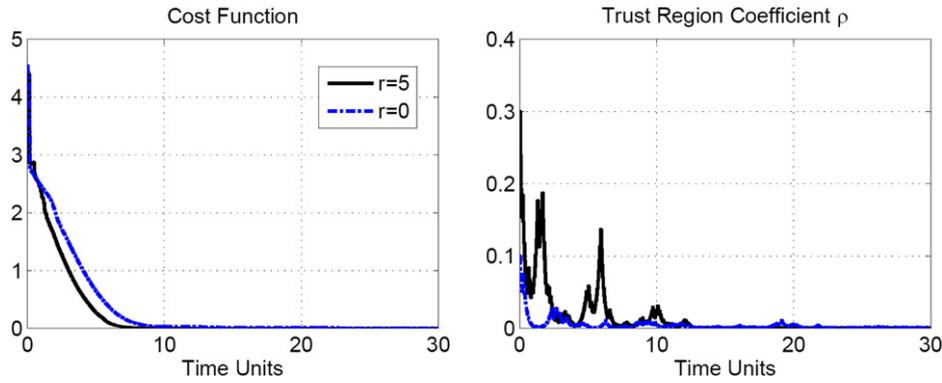
Fig. 3 shows the relevance of the trust region adaptation mechanism since it shows the closed-loop behavior of the cost function value in the absence of this mechanism and for two different values of the parameter  $\rho$ .

The relevance of the use of  $r > 0$  is illustrated through Fig. 4 in which two different values  $r = 0$  and  $r = 5$  are used. A noticeable improvement of the decreasing rate of the cost function can be obtained at less than a 4% increase in the computational burden.

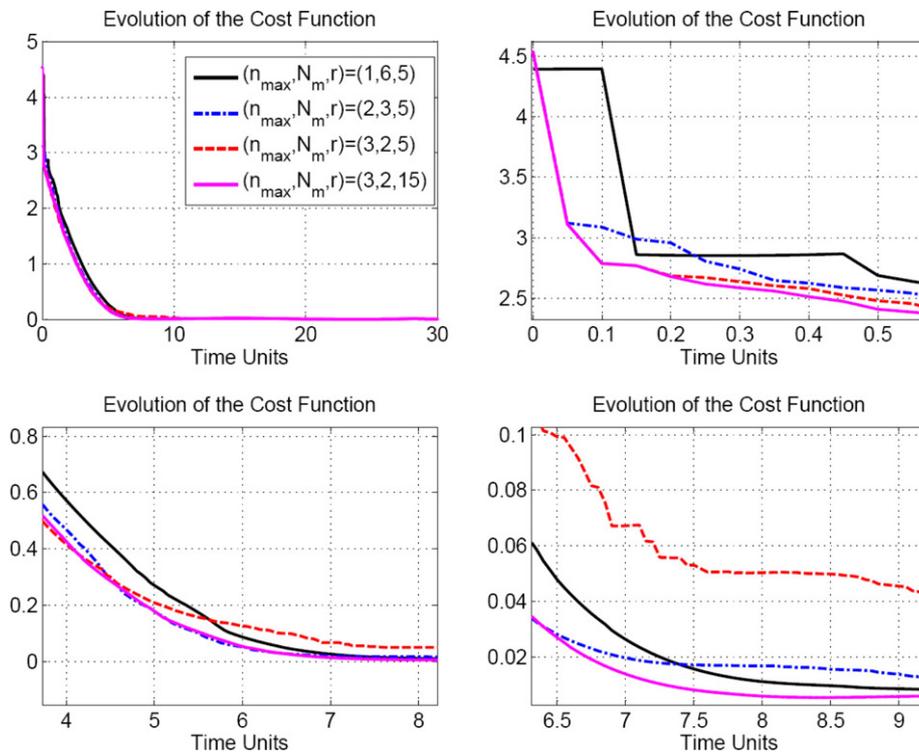
Finally, Fig. 5 goes deeper in the comparison by adding to the above two choices the following ones:

$$(n_{\max}, N_m, r) = (3, 2, 5); \quad (n_{\max}, N_m, r) = (3, 2, 15).$$

Moreover, zooms at different time intervals are proposed to better understand the effect of each parameter. More precisely, the following remarks can be made:



**Fig. 4.** Illustration of the usefulness of taking  $r > 0$ . (Black-solid): the same results as Fig. 2 (i.e with  $r = 5$ ). (Blue-dotted): no additional computation (i.e  $r = 0$ ). Note that since  $N_p = 150$ , the use of  $r = 5$  increases the computational burden related to the system integration by less than 4%. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)



**Fig. 5.** Comparing different pairs  $(n_{\max}, N_m, r)$ . (Black-solid):  $(n_{\max}, N_m, r) = (1, 6, 5)$ /(blue-dash-dotted)  $(n_{\max}, N_m, r) = (2, 3, 5)$ /(red-dotted)  $(n_{\max}, N_m, r) = (3, 2, 5)$ /(magenta, dot-marked)  $(n_{\max}, N_m, r) = (3, 2, 15)$ . Far from the  $J \gg 0$ , higher  $n_{\max}$  induces faster convergence. Close to  $J \sim 0$ , lower  $N_m$  induces bad conditioning of the identification of the  $x$ -related terms of the cost function. This can be compensated for by increasing  $r$ . (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

‡ The use of a higher value of  $n_{\max}$  (3 rather than 2) still improves the decrease of the cost function [sub-plot (1, 2)] as long as the value of the cost function is still sufficiently high.

‡ However, when the state approaches the desired value, increasing  $n_{\max}$  while keeping the product  $n_{\max} \cdot N_m = 6$  leads to a bad identification of how the cost function depends on  $x$  and this deteriorates the decrease of the cost function [sub-plot (2, 1)]. This can be viewed on the red-dotted line going above the (blue-dashed) one. Using a higher ( $r = 15$ ) enables to recover this lack of information [subplot (2, 2) in magenta-solid line] at a slight increase (10%) in the computational burden.

### 5. Conclusion and future work

In this paper, a novel framework is proposed that addresses the NMPC real-time implementation issue when applied to a

system with fast dynamics. The framework is based on the control parametrization approach to enhance a rather low dimensional optimization problem. The latter is solved by a mixed (Sequential Quadratic)/(Trust region) approach based on recursive least squares based iterative identification of a quadratic approximation of the cost function as a function of the state vector and the vector of control parameters. An illustrative example shows the efficiency of the proposed scheme in achieving the control task that consists of stabilizing the PVTOL dynamics around some desired position while using only two system simulations, the solution of a low dimensional linear system (of dimension 90) and the solution of 6-dimensional quadratic program with 12 constraints at each sampling period. It is important to underline that the proposed scheme is intimately linked to the two following particularities of the problem under consideration that are: the low dimensional parametrization of the control profile and the translatability property of the control parametrization map.

Future works will focus on investigating the performance of the proposed schemes on real-life systems including ones that involve real state dependent constraints.

## References

- Alamir, M. (2001). Nonlinear receding horizon sub-optimal guidance law for minimum interception time problem. *Control Engineering Practice*, 9(1), 107–116.
- Alamir, M. (2006). Stabilization of nonlinear system using receding-horizon control schemes: a parametrized approach for fast systems. In *Lecture notes in control and information science*. London: Springer, ISBN: 1-84628-470-8.
- Alamir, M., Murilo, A., Amari, R., Tona, P., Furhapter, R., & Ortner, P. (2009). On the use of parameterized NMPC in real-time automotive control. In *Automotive model predictive control*. Springer-Verlag.
- Conn, A. R., Gould, N. I. M., & Toint, Ph. L. (2006). Trust region methods. In *MPS-SIAM series on optimization*. Philadelphia: SIAM.
- DeHaan, D., & Guay, M. (2007). A real-time framework for model-predictive control of continuous-time nonlinear systems. *IEEE Transactions on Automatic Control*, 52(11), 2047–2057.
- Diehl, M., Bock, H. G., & Schloöder, J. P. (2005). A real-time iteration scheme for nonlinear optimization in optimal feedback control. *Siam Journal on Control and Optimization*, 43, 1714–1736.
- Martin, P., Devasia, S., & Paden, B. A different look at output tracking: control of a vtol aircraft. In *Proceedings of the 33rd IEEE Conference on Decision and Control*. 1994.
- Mayne, D. Q., Rawlings, J. B., Rao, C. V., & Sckaert, P. O. M. (2000). Constrained model predictive control: stability and optimality. *Automatica*, 36(6), 789–814.
- Ohtsuka, T. (2004). A continuation/gmres method for fast computation of nonlinear receding-horizon control. *Automatica*, 40, 563–574.
- Ortner, P., Bergmann, R., Ferreau, H. J., & del Re, L. (2009) Nonlinear model predictive control of a diesel engine airpath. In *IFAC workshop on control applications of optimisation, Jyvaskyla*.
- Shimizu, Y., Ohtsuka, T., & Diehl, M. (2006) Nonlinear receding horizon control of an underactuated hovercraft with a multiple-shooting-based algorithm. In *Proceedings of the 2006 IEEE International Conference on Control Applications, Munich, Germany October 4–6*.



**Mazen Alamir** graduated in Mechanics (Grenoble, 1990) and Aeronautics (Toulouse, 1992). He received his Ph.D. in Nonlinear Model Predictive Control in 1995. Since 1996, he has been a CNRS Research Associate in the Control Systems Department of Gipsa-lab, Grenoble. His main research topics are model predictive control, receding horizon observer, nonlinear hybrid systems, signature based diagnosis and optimal cancer treatment as well as industrial applications. He is member of the IFAC technical committee on Nonlinear Systems and served as head of the “Nonlinear Systems and Complexity” research group of the Control Systems Department of Gipsa-lab, Grenoble.

Home page: [www.mzenalamir.fr](http://www.mzenalamir.fr).