A Framework for Control Updating Period Monitoring In Real-Time NMPC Schemes

Mazen Alamir

University of Grenoble
CNRS/Gipsa-lab, Control Systems Department,
BP 46, Domaine Universitaire, 38400 Saint Martin d'Hères, France.
mazen.alamir@gipsa-lab.inpg.fr

Summary. In this contribution, a general scheme is proposed that enables the control updating period used in Nonlinear Model Predictive Control (NMPC) scheme to be dynamically optimized. Such a scheme can be of great interest when applying NMPC to systems with fast dynamics. The updating scheme is based on the on-line identification of generic models for both solver efficiency and disturbance effect on the optimal cost behavior. The efficiency of the proposed approach is illustrated on several examples.

1 Problem Statement

In order to properly state the problem addressed in this paper, the basic framework of parameterized NMPC is first recalled, then the particular implementation that is adopted for fast systems is precisely described.

1.1 Recall on Parameterized NMPC

Let us consider nonlinear systems that admit the following implicit model:

$$x(t) = X(t, x_0, \mathbf{u}) \quad ; \quad t \leq T \tag{1}$$

where $x(\cdot) \in \mathbb{R}^n$ is the state trajectory that starts at the initial value $x(0) = x_0$ under the control profile $\mathbf{u} \in \mathbb{U}^{[0,T]}$ for some subset $\mathbb{U} \subset \mathbb{R}^m$ of admissible control inputs. The implicit map X is obtained by any suitable modeling framework. T stands for some prediction horizon over which the model (1) is meaningful.

In what follows, the evolution of the real system is clearly distinguished from that of the model (1) by adopting the following notation:

$$x(t) = X^r(t, x_0, \mathbf{u}, \mathbf{w}) \quad ; \quad t \leq T \tag{2}$$

where $\mathbf{w}$ stands for the unknown uncertainties/disturbance profile that may affect the system during the time interval.

When using (1) in a parameterized NMPC scheme [1, 2], it is quite usual to use some fixed sampling period $\tau > 0$ that reflects the characteristic time of the system together with some piecewise-constant open-loop control parametrization:

$$\mathcal{U}_{pwc}(p) := \big(u^{(1)}(p), \dots, u^{(N)}(p)\big) \in \mathbb{U}^N \quad ; \quad N\tau = T \tag{3}$$

where $u^{(k)}(p)$ defines the open-loop control value that holds over the future interval $[(k-1)\tau, k\tau]$. As soon as such a parametrization is fixed, the implicit model can be written in terms of the parameter vector p with an obvious overloaded notations:

$$x(t) = X(t, x_0, p) \quad ; \quad t \leq T \tag{4}$$

By doing so, NMPC related cost functions can be viewed as functions of the initial state and the parameter vector p, namely $J(p, x_0)$. This enables to define a state dependent optimal parameter $\hat{p}(x)$ by:

$$\hat{p}(x) := \arg \min_{p \in \mathbb{P}(x) \subset \mathbb{P}} J(p, x) \tag{5}$$

where $\mathbb{P}(x) \subset \mathbb{P} \subset \mathbb{R}^{n_p}$ is some set of admissible parameter values that may depend on the state. In what follows, the above optimisation problem is denoted by $\mathcal{P}(x)$.

according to (3), the optimal value $\hat{p}(x)$ defines a sequence of control inputs by

$$\mathcal{U}_{pwc}(\hat{p}(x)) := \big(u^{(1)}(\hat{p}(x)), \dots, u^{(N)}(\hat{p}(x))\big) \in \mathbb{U}^N \quad ; \quad N\tau = T \tag{6}$$

Classically, NMPC schemes use τ as updating period with the following time sampled state feedback law:

$$K = u^{(1)} \circ \hat{p} \tag{7}$$

that is, only the first input in the optimal sequence $\mathcal{U}_{pwc}(\hat{p}(x(t_k))$ is applied during the time interval $[t_k, t_k + \tau = t_{k+1}]$ before a new optimal sequence is computed by solving the optimization problem associated to the next state $x(t_{k+1})$ and so on.

Such scheme assumes that the optimization problem can be solved, or at least a sufficient number of iterations towards its solution can be done in less than τ time units. For fast systems where τ needs to be very small, this condition may become hard to satisfy and some dedicated scheme has to be adopted as shown in the following section. Note that throughout the paper, *iterations* refer to the number of functions evaluations as it is the key operation in NMPC schemes.

1.2 Implementation Scheme for Fast Systems

Since the basic sampling period τ may be too small for control updating. The control updating period can be taken equal to some multiple $\tau_u = N_u\tau$ of the basic sampling period τ. More precisely, for fast systems, one needs to use the following control scheme:

- Denote by $t_i^u = i\tau_u$ the so-called updating instants (see later for a precise definition of the updating process)

- Denote by $t_k = k\tau$ the so-called sampling instants
- At instant $t = 0$, some initial parameter vector $p(t_0^u = 0) \in \mathbb{P}(x_0)$ is chosen (arbitrarily or according to some ad-hoc initialization rule). The corresponding sequence of control inputs

$$\mathcal{U}_{pwc}(p(t_0^u)) = \big(u^{(1)}(p(t_0^u)) \ \dots \ u^{(N_u)}(p(t_0^u)) \ u^{(N_u+1)}(p(t_0^u)) \ \dots \ u^{(N)}(p(t_0^u))\big)$$

is computed.
- During the time interval $[t_0^u, t_1^u]$, the first N_u control inputs

$$\big(u^{(1)}(p(t_0^u)) \ \dots \ u^{(N_u)}(p(t_0^u))\big)$$

are applied. In parallel, the computation unit performs successively the following two tasks:
  1. Compute the model based prediction of the state at the future instant t_1^u according to:
  $$\hat{x}(t_1^u) = X(\tau_u, x(t_0^u), p(t_0^u))$$
  This prediction involves only the first N_u elements of $\mathcal{U}_{pwc}(p(t_0^u))$.
  2. Try to solve the optimization problem $\mathcal{P}(\hat{x}(t_1^u))$ by performing q steps of some optimization process $\mathcal{S}$ with an initial guess $p^+(t_0^u)$. The later may be obtained from $p(t_0^u)$ by some appropriate transformation. For instance, the translatability property invoked in [1, 2] can be used. However, the precise choice of the transformation $p^+(t_1^u)$ is meaningless for the remainder of the sequel. This is shortly denoted as follows:
  $$p(t_1^u) = \mathcal{S}^q\big(p^+(t_0^u), \hat{x}(t_1^u)\big) \tag{8}$$

  Note that $p(t_1^u)$ is generally different from the optimal value $p(\hat{x}(t_1^u))$ that may need much more iterations to be obtained.
- During the time interval $[t_1^u, t_2^u]$, the N_u control inputs

$$\big(u^{(1)}(p(t_1^u)) \ \dots \ u^{(N_u)}(p(t_1^u))\big)$$

are applied and so on.

To summarize

Under the above implementation rules, the state of the real system at the updating instants $\{t_i^u\}_{i\geq 0}$ is governed by the following coupled dynamic equations:

$$x(t_i^u) = X^r\big(\tau_u, x(t_{i-1}^u), p(t_{i-1}^u), \mathbf{w}\big) \tag{9}$$

$$p(t_i^u) = \mathcal{S}^q\Big(p^+(t_{i-1}^u), \underbrace{X(\tau_u, x(t_{i-1}^u), p(t_{i-1}^u))}_{\hat{x}(t_i^u)}\Big) \tag{10}$$

that clearly involve an extended dynamic state

$$z = \big(p^T \ x^T\big)^T \in \mathbb{R}^{n+n_p}$$

and that heavily depend on the design parameters τ_u, $\mathcal{S}$ and q.

1.3 The Scope of the Present Contribution

In the present contribution, the interest is focused on the following issue:

> *Given some optimization process $\mathcal{S}$, propose a concrete updating rules for both the control updating period τ_u and the number of iterations q in order to* improve *the closed-loop behavior under the real-time NMPC implementation framework proposed in section 1.2.*

In particular, it is worth emphasizing here that the choice of the optimization process $\mathcal{S}$ that may be used in order to come closer to the optimal value $\hat{p}(x(t_i^u))$ is not the issue of this paper. The updating rule proposed hereafter can be used as an additional *layer* to any specific choice of the optimization process $\mathcal{S}$.

2 Theoretical Framework

Although we seek a general framework, we still need some assumptions on the optimization process $\mathcal{S}$ being used. In particular, it is assumed that the process is *passive* in the following trivial sense

Assumption 1 [The optimizer is passive]
For all $q \in \mathbb{N}$ and all $z := (p, x) \in \mathbb{R}^{n_p} \times \mathbb{R}^n$, the following inequality holds:

$$J(\mathcal{S}^q(p, x), x) \leq J(p, x) \tag{11}$$

which simply means that at worst, the optimize returns the initial guess p.

A key property in the success of the real-time NMPC scheme is the efficiency of the optimizer, namely, its ability to lead to a best value $\mathcal{S}^q(p, x)$ of the parameter vector starting from the initial guess p. This efficiency may heavily depends on the pair (p, x) and can be quantified through the following definition:

Definition 1 [Efficiency of the optimizer]
For all $(p, x) \in \mathbb{R}^{n_p} \times \mathbb{R}^n$, the map defined by:

$$E^f_{(p,x)}(q) := \frac{J\big(\mathcal{S}^q(p, x), x\big)}{J(p, x)} \tag{12}$$

is called the efficiency map at (p, x).

Note that by virtue of assumption 1, the efficiency map satisfies the inequality $E^f_{(p,x)}(q) \leq 1$ for all q and all (p, x).

The last element that plays a crucial role in determining what would be an optimal choice is the level of model discrepancy. More precisely, how this discrepancy degrades the value of the resulting cost function at updating instants. This may be handled by the following definition:

Definition 2 [Model Mismatch Indicator]
For all pair (p,x), the map defined by:

$$D_{(p,x)}(\tau_u) := \sup_{(\bar{p},\mathbf{w})\in\mathbb{P}\times\mathbb{W}} \Big[\frac{J(\bar{p},X^r(\tau_u,x,p,\mathbf{w}))}{J(\bar{p},X(\tau_u,x,p))}\Big] \tag{13}$$

is called the model mismatch indicator at (p,x).

Note that by definition, $D_{(p,x)}(0)=1$ for all (p,x) since one clearly has $X(0,x,p) = X^r(0,x,p,\mathbf{w}) = x$. As a matter of fact, $D_{(p,x)}(\tau_u)$ represents a worst case degradation of the cost function that is due to the bad prediction $\hat{x}(t^u_{i+1})$ of the future state $x(t^u_{i+1})$ at the next decision instant.

Based on the above definitions, the following result is straightforward:

Lemma 1 [A Small Gain Result]
Under the real-time implementation scheme given by (9)-(10), the dynamic of the cost function satisfies the following inequality:

$$J(x(t^u_{i+1}),p(t^u_{i+1})) \leq \big[E^f_{(i)}(q)\big]\big[D_{(i)}(\tau_u)\big]\cdot J(x(t^u_i),p(t^u_i)) \tag{14}$$

where the following short notations are used:

$$E^f_{(i)}(q) := E^f_{(p(t^u_i),x(t^u_i))}(q) \quad ; \quad D_{(i)}(\tau_u) := D_{(p(t^u_i),x(t^u_i))}(\tau_u)$$

Consequently, if the following small gain condition is satisfied for all i:

$$K_{(i)}(q,\tau_u) := \big[E^f_{(i)}(q)\big]\big[D_{(i)}(\tau_u)\big] < 1 \tag{15}$$

then the closed loop evolution of (p,x) is such that asymptotically, $p(t^u_i)$ is a minimum for $J(\cdot,x(t^u_i))$. ♡

Recall that our aim is to define a rationale in order to choose the parameters of the real-time scheme, namely q and τ_u. The *small gain* condition (15) of lemma 1 provides a constraint guiding this choice (provided that estimations of the maps $E^f_{(i)}$ and $D_{(i)}$ are available). Another trivial constraint expresses the fact that the time needed to perform q iterations is lower than the control updating period τ_u, this constraint can be written as follows:

$$[\tau_c]\cdot q - \tau_u \leq 0 \tag{16}$$

where τ_c is the time needed to perform a single iteration (function evaluation).

Note that inequalities (15)-(16) are constraints that take account for convergence and feasibility issues respectively. The quality of the closed loop may be expressed in term of the settling time, that is the predicted time necessary to decrease the cost function by 95% of its initial value, this can be expressed by the following cost function:

$$t_r(q,\tau_u) := \frac{\tau_u}{|\log(K_{(i)}(q,\tau_u))|} \tag{17}$$

Therefore, provided that an approximation of the map $K_{(i)}(q,\tau)$ is obtained (via on-line identification), the following constrained optimization problem may be used to compute at each updating instant t_i^u the values of the *optimal* control updating period $\tau_u(t_i^u)$ and the number of iterations $q(t_i^u)$:

$$(q(t_i^u),\tau_u(t_i^u)) := arg \min_{(q,\tau_u)} \frac{\tau_u}{|\log(K_{(i)}(q,\tau_u))|} \quad \text{under the constraints} \tag{18}$$

$$K_{(i)}(q,\tau_u) < 1 \quad ; \quad [\tau_c]\cdot q - \tau_u \leq 0 \quad ; \quad \tau_u \in \mathbb{N}\cdot\tau \tag{19}$$

where the last constraints in (19) expresses the fact that the updating period τ_u must be a multiple of the basic sampling period τ. Note however that for a given updating period τ_u there can be no benefit from a number of iterations that is lower than the maximum allowable one. Consequently, the solution of the constrained optimization problem can be clearly obtained by:

$$q(t_i^u) = \frac{\tau_u(t_i^u)}{\tau_c} \quad \text{where } \tau_u(t_i^u) \text{ is given by} \tag{20}$$

$$\tau_u(t_i^u) = arg \min_{\tau_u\in\mathbb{N}\tau} \frac{\tau_u}{|\log(K_{(i)}(int(\frac{\tau_u}{\tau_c}),\tau_u))|} \quad \text{under } K_{(i)}(int(\tfrac{\tau_u}{\tau_c}),\tau_u) < 1 \tag{21}$$

where for all $\rho\in\mathbb{R}$, $int(\rho)$ stands for the integer value of ρ.

To summarize: In the dynamic equations (9)-(10) of the closed-loop system, the updating period τ_u and the number of iterations q have to be made dynamic by rewriting the equations as follows:

$$t_{i+1}^u = t_i^u + \tau_u(t_i^u) \quad ; \quad t_0^u = 0 \tag{22}$$

$$\hat{x}(t_{i+1}^u) = X(\tau_u(t_i^u), x(t_i^u), p(t_i^u)) \tag{23}$$

$$x(t_{i+1}^u) = X^r(\tau_u(t_i^u), x(t_i^u), p(t_i^u), \mathbf{w}) \tag{24}$$

$$p(t_{i+1}^u) = \mathcal{S}^{q(t_i^u)}\big(p^+(t_i^u), \hat{x}(t_{i+1}^u)\big) \tag{25}$$

In the remainder of this paper, a scheme is proposed in order to identify on line the key function map $K_{(i)}(\cdot,\cdot)$ based on-line available measurements.

3 On-line identification of the key maps

The aim of this section is to propose some on-line identification scheme for the key map:

$$K_{(i)}(q,\tau) = E_{(i)}^f(q)\cdot D_{(i)}(\tau_u) \tag{26}$$

by identifying the efficiency map $E_{(i)}^f$ and the model uncertainty related map $D_{(i)}$ using the definitions (12) and (13) respectively. In order to do this, some notations have to be introduced in order to simplify the expressions. For instance, the following short expressions related to the cost function are used:

$J_{(i)}^+ = J(p^+(t_{i-1}^u), \hat{x}(t_i^u))$ The value when the iterations start

$\hat{J}_{(i)} = J(p(t_i^u), \hat{x}(t_i^u))$ The value when the iterations stop

$J_{(i)} = J(p(t_i^u), x(t_i^u))$ The effective value given the true state $x(t_i^u)$

Figure 1 shows how these quantities evolve during the closed-loop evolution of the system under the real-time NMPC scheme given by (22)-(25).

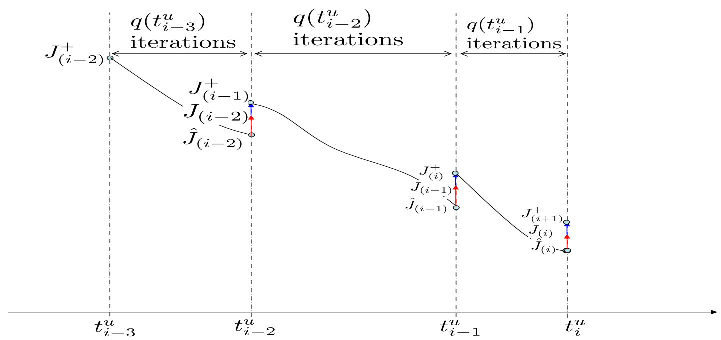


Fig. 1. Typical behavior of the different cost functions during closed-loop system's lifetime. Note that $\hat{J}_{(s)}$ is lower that $J^+_{(s)}$ thanks to the solver's efficiency. $J_{(s)}$ is generally different from $\hat{J}_{(s)}$ because of model mismatches while $J^+_{(s)}$ is different from $J_{(s-1)}$ because of the horizon movement that slightly modifies the cost function.

More precisely, at instant t^u_{i-3}, the iterations start with the initial guess $J^+_{(i-2)} = J(p^+(t^u_{i-3}), \hat{x}(t^u_{i-2}))$. During the interval $[t^u_{i-3}, t^u_{i-2}]$, $q(t^u_{i-3})$ iterations are performed and the predicted sub-optimal value

$$\hat{J}_{(i-2)} = J(p(t^u_{i-2}), \hat{x}(t^u_{i-2})) \quad ; \quad p(t^u_{i-2}) := \mathcal{S}^{q(t^u_{i-3})}\Big(p^+(t^u_{i-2}), \hat{x}(t^u_{i-3})\Big) \qquad (27)$$

is achieved. However, the true value of the cost function is given by

$$J_{(i-2)} = J(p(t^u_{i-2}), x(t^u_{i-2}))$$

which is generally different from the predicted value $\hat{J}_{(i-2)}$ because of model mismatches. Now, the next optimization process (to be performed over $[t^u_{i-2}, t^u_{i-1}]$) is initialized using the value $p^+(t^u_{i-2})$ and the predicted future state $\hat{x}(t^u_{i-1})$ leading to the initial guess $J^+_{(i-1)}$ that may be slightly different from the present value $J_{(i-2)}$ and so on.

Recall that at each instant t^u_i, the computation of the next updating instant $t^u_{i+1} = t^u_i + \tau_u(t^u_i)$ and the corresponding number of iterations $q(t^u_i)$ is based on the solution of the constrained optimization problem (18). This needs the maps $E^f_{(i)}$ and $D_{(i)}$ to be identified based on the past measurements. However, the identification step has to rely on some beforehand given structure of the functions to be identified. The following structures may be used among many others for $E^f_{(i)}$ and $D_{(i)}$:

$$E^f(q) := \frac{1}{\alpha^f \cdot \max\{0, q - q^f\} + 1} \quad ; \quad D(\tau_u) := 1 + \alpha^D \cdot \tau_u \qquad (28)$$

where α^f, q^f and α^D are the parameters to be identified. Note that α^f monitors the speed of convergence of the optimizer while q^f stands for the minimum number of iterations before a decrease in the cost function may be obtained. This *dead-zone* like property is observed in many optimization algorithms. Finally, the coefficient α^D reflects the effect of the model mismatch on the evolution of the cost function.

Consequently, by the very definition of the map $D_{(i)}$, the following straightforward identification rule for α^D can be adopted:

$$\alpha^D_{(i)} := \frac{J_{(i)} - \hat{J}_{(i)}}{\tau_u(t^u_{i-1}) \cdot \hat{J}_{(i)}}$$

The estimation of the efficiency map's parameters $q^f_{(i)}$ and $\alpha^f_{(i)}$ is obtained based on the behavior of the iterations that are performed during the last updating period $[t^u_{i-1}, t^u_i]$. This behavior is described by the following sequence:

$$\left\{ d_{(i,j)} := \frac{\hat{J}_{(i,j)}}{J^+_{(i)}} \right\}_{j=0}^{q(t^u_{i-1})} \quad := \quad \left\{ \frac{\mathcal{S}^j(p^+(t^u_i), \hat{x}(t^u_i))}{\mathcal{S}^0(p^+(t^u_i), \hat{x}(t^u_i))} \right\}_{j=0}^{q(t^u_{i-1})}$$

where the notation $\hat{J}_{(i,j)} := \mathcal{S}^j(p^+(t^u_i), \hat{x}(t^u_i))$ is used to refer to the value of the estimated cost after j iterations.

Indeed, based on the computed sequence (29), the following estimations of $q^f_{(i)}$ and $\alpha^f_{(i)}$ can be obtained:

$$q^f_{(i)} = \max\left\{ j \in \{1, \dots, q(t^u_{i-1})\} \quad | \quad d_{(i,j)} = 1 \right\} \tag{29}$$

$\alpha^f_{(i)}$ is the least squares solution of the following system [see (28)]

$$\left[d_{(i,j)} \cdot \max(0, j - q^f_{(i)}) \right] \cdot \alpha^f_{(i)} = 1 - d_{(i,j)} \quad ; \quad j = 1, \dots, q(t^u_{i-1}) \tag{30}$$

where the least squares problem is obtained by putting together all the linear equations (30) (in the unknown $\alpha^f_{(i)}$) corresponding to the different values of j. Note that the computations in (29) and (30) are straightforward and can therefore be done without significant computational burden when compared to NMPC-related computations.

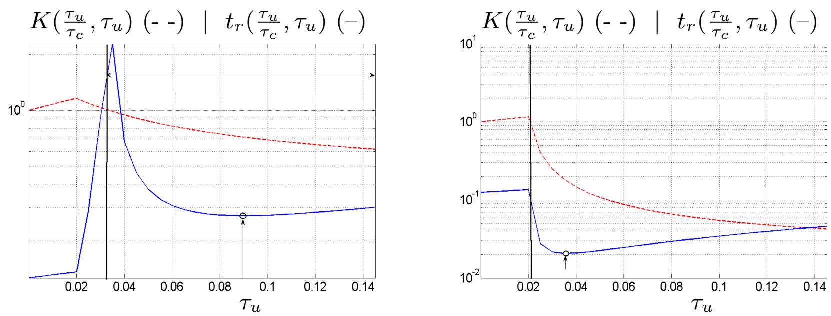
4 Numerical Investigations

In this section, numerical examples are given in order to illustrate the concepts presented in the preceding sections. First, the form of the cost function used in (21) to determine the optimal updating period is first illustrated under several configurations of the problem's parameters q^f, α^f and α^D. Then a concrete and simple example of closed-loop behavior under model mismatches is proposed to illustrate the efficiency of the proposed scheme.

4.1 Qualitative analysis

Let us consider an efficiency map E^f and an uncertainty map D that have the structure given by (28). Note that the ability of these structures to represent realistic situations is strengthened in the following section through a simple illustrative example. It goes without saying that other structures may be adopted for on-line identification. The conjecture here is that the qualitative implications and the efficiency of the resulting scheme remain unchanged.

The aim of this section is to show how different sets of parameters involved in (28), namely q^f, α^f and α^D influence the resulting optimal updating period τ_u. More precisely, Figures 2 and 3 underline the influence of the efficiency parameter α^f (Figure 2) and the uncertainty parameter α^D (Figure 3).



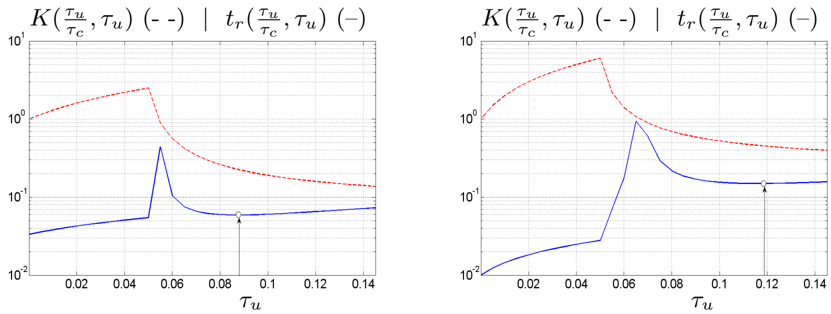
(a) (b)

Fig. 2. Variations of the stability indicator $K(\tau_u/\tau_c, \tau_u)$ [dotted line] and the settling time $t_r(\tau_u/\tau_c, \tau_u)$ [solid line] as functions of the updating period τ_u. Figures (a) and (b) correspond to two different values of the efficiency parameter α^f respectively equal to 0.1 (a) and 2 (b). The remaining parameters are identical [$q^f = 4$, $\tau_c = 0.005$ and $\alpha^D = 8$]. Note the two different corresponding optimal updating periods (minimum of the curve $t_r(\tau_u/\tau_c, \tau_u)$) respectively equal to 90 ms (a) and 35 ms (b).

4.2 Illustrative example

References

1. Alamir M., "Stabilization of Nonlinear Systems Using Receding-Horizon Control Schemes: A Parametrized Approach for Fast Systems", *Lecture Notes in Control and Information Sciences, Springer*. London, ISBN 1-84628-470-8 (2006)
2. Alamir M., "A Low Dimensional Contractive NMPC Scheme for Nonlinear Systems Stabilization: Theoretical Framework and Numerical Investigation on Relatively Fast Systems". In *R. Findeisen, F. Allgöwer, L. Biegler (Eds.). Assessement and future directions in Nonlinear Model Predictive Control. Lecture Notes in Control and Information Sciences, Springer-Verlag* (2006)



(a) (b)

Fig. 3. Variations of the stability indicator $K(\tau_u/\tau_c, \tau_u)$ [dotted line] and the settling time $t_r(\tau_u/\tau_c, \tau_u)$ [solid line] as functions of the updating period τ_u. Figures (a) and (b) correspond to two different values of the uncertainty parameter α^D respectively equal to 8 (a) and 30 (b). The remaining parameters are identical [$q^f = 10$, $\tau_c = 0.005$ and $\alpha^f = 2$]. Note the two different corresponding optimal updating periods (minimum of the curve $t_r(\tau_u/\tau_c, \tau_u)$) respectively equal to 88 ms (a) and 120 ms (b).