On Adaptive Measurement Inclusion Rate
In Real-Time Moving-Horizon Observers

Mazen Alamir *

* *University of Grenoble, Gipsa-lab, BP 46, Domaine Universitaire,*
38400 Saint Martin d'Hères, France. (mazen.alamir@grenoble-inp.fr)

Abstract: This paper investigates a self adaptation mechanism regarding the rate with which new measurements have to be incorporated in Moving-Horizon state estimation algorithms. This investigation can be viewed as the dual of the one proposed by the author in the context of real-time model predictive control. An illustrative example is provided in order to assess the relevance of the proposed updating rule.

Keywords: Moving-Horizon Observers; Nonlinear Systems; On-line Optimization.

1. INTRODUCTION

Moving-Horizon Observers (MHO) are algorithms that involve repeated on-line optimization in order to update the estimated value of the state [Michalska and Mayne, 1995, Alamir, 1999, Rao et al., 2003, Kuhl et al., 2011]. More precisely, the estimated state is the optimal solution of an optimization problem in which the cost function accounts for the interpretation of the past measurement (over some observation horizon) and the compatibility with the presumed model of the system. Periodically, the observation horizon is shifted in order to take into account new measurements while discarding older ones. Typically, the shifting period is taken equal to the measurement acquisition period which, in the early stage of MHO development was compatible with the assumption of instantaneous solvability of the optimization problem.

Amazingly enough, this feature has never been revisited despite the recent advances in real-time implementation framework where the optimization is truncated before an optimal solution is reached. In such frameworks, the time that is available for the optimization is tightly related to the horizon shifting period. More precisely, the question is the following:

> *How many iterations of the optimization algorithm should be executed before the cost function is updated by including new available measurements?*

The same question can be reformulated equivalently:

> *How many new measurement instances should be acquired before the cost function is updated? (During how many periods should the optimizer work on the same unchanged problem before the latter is updated to account for the newly available measurements?)*

To assess the relevance of the questions above, one can simply argue that it has never been proved that the commonly used answer is optimal in all circumstances. Yet, there is another, probably more convincing argument, which is that recent investigations (see [Alamir, 2008, 2013]) on the dual problem of Model Predictive Control (MPC) context clearly showed the existence of time varying, context dependent, optimal control updating period (which is the period during which the optimizer works on the same problem before the control defined by the updated parameter is applied to the system).

In this paper, it is shown that a similar formulation, similar techniques and, under certain circumstances, similar conclusions can be obtained regarding the optimal measurement inclusion rate problem.

The paper is organized as follows: First, the problem is stated in section 2 by showing that monitoring the measurement inclusion rate can be viewed as a discrete-time output regulation problem in which the control input is the number of iterations to be performed before new set of measurements is accounted for. In section 3, a gradient-based heuristic is proposed to solve this problem leading to an updating law for the measurement inclusion rate. An illustrative example is given in section 4 in order to assess the efficiency of the proposed updating scheme and its ability to handle varying circumstances during the system's lifetime. Finally, section 5 concludes the paper and gives hints for further investigation.

2. THE MEASUREMENT INCLUSION RATE AS AN OUTPUT REGULATION PROBLEM

Let us consider dynamical systems that are governed by the following evolution law:

$$x(k) = X(M, x(k - M), \mathbf{u}(k)) \qquad (1)$$
$$y(k) = h(x(k), u(k)) \qquad (2)$$

where $x \in \mathbb{R}^n$ is the state vector, M is some integer and where the notation $x(k)$ refers to the state at instant $k\tau$ for some sampling period τ that is supposed here (without loss of generality) equal to the measurement acquisition period. $\mathbf{u}(k)$ represents the sequence of some measured exogenous inputs such that:

$$\mathbf{u}(k) = \{u(k-M), \dots, u(k)\} \quad (3)$$

Note that equation (1) represents a multi-step state transition map that gives the value of the state vector starting from the initial value $x(k-M)$ and under the input sequence defined over the time interval $[k-M, k]$ by $\mathbf{u}(k)$. In the sequel, the length of the sequence of inputs $\mathbf{u}(k)$ is defined from the context so that the same notation $\mathbf{u}(k)$ can be used to designate sequences of different lengths provided that the last input is $u(k)$.
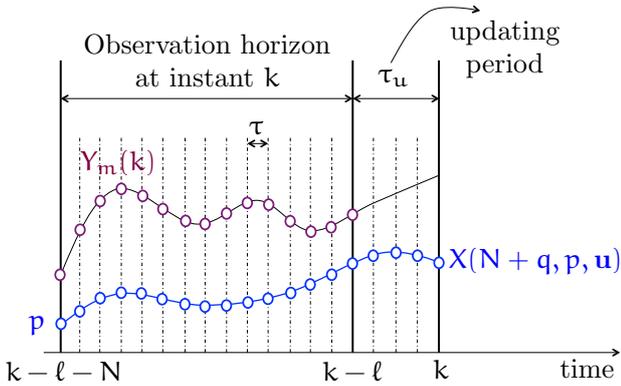
Fig. 1. Schematic view showing the key notation used in the definition of MHO.

Ideal Moving-Horizon Observers (**MHO**) are algorithms which compute a rational estimation $\hat{x}(k)$ of the current state $x(k)$ based on the last estimation $p^{opt}(k-\ell)$ of some decision variable p and the sequence of past measured outputs $Y_m(k)$ defined by (see Figure 1):

$$Y_m(k) := \begin{pmatrix} y_m(k) \\ \vdots \\ y_m(k-N) \end{pmatrix} \in \mathbb{R}^{(N+1) \cdot n_y} \quad (4)$$

More precisely, the general form of MHO-based state estimation is given by:

$$\hat{x}(k) = F(p^{opt}(k), \mathbf{u}(k)) \quad (5)$$

$$p^{opt}(k) = \arg\min_{p \in \mathbb{P}} J\big(p \mid Y_m(k-\ell), \mathbf{u}(k-\ell), p^{opt}(k-\ell)\big) \quad (6)$$

where

✓ $p(k-\ell)$ is a vector of decision variables that is generally taken to be the presumed value of the state at the beginning of the observation horizon (at instant $k-\ell-N$) or it can be the whole state trajectory as in the multiple-shooting version of the MHO implementation [Kuhl et al., 2011].

✓ $\mathbb{P} \subset \mathbb{R}^n$ is the set of admissible parameter values.

✓ $J(\cdot)$ is the cost function that is generally decomposed into an output prediction error-related term and a system's dynamic related term with the standard trade-off weighting coefficients that depend on noise covariance matrices.

Remark 1. At this stage, it is very important to underline that the definition of the cost function is out of the scope of the present contribution in which the cost function is supposed to be given by the designer. The aim of the forthcoming developments is to suggest a way to distribute the minimization of this cost function over the system lifetime. Whether this enhances a better estimation of the state or not depends on the relevance of the weighting matrices that are used to define this cost function. ♠
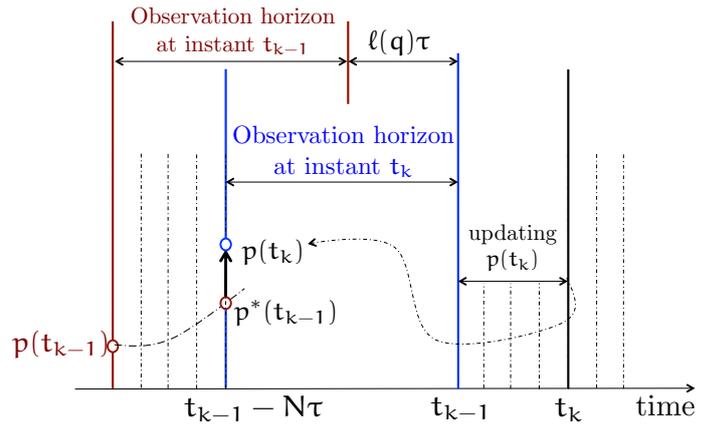
Figure 1 shows a schematic view of the situation when the presumed value of the state at instant $k-\ell-N$ is taken as decision variable, denoted by p. Note that in this specific case, the map F involved in (5) is given by:

$$\hat{x}(k) = X(N+\ell, p^{opt}(k), \mathbf{u}(k)) =: F(p^{opt}(k), \mathbf{u}(k)) \quad (7)$$

In this framework, the time period that lasts between the availability of the measurement data $Y_m(k-\ell)$ (namely instant $k-\ell$) and the delivery of the optimal value $p^{opt}(k)$ (instant k) is equal to the so called updating period τ_u given by:

$$\tau_u = \ell \cdot \tau \quad (8)$$

During this amount of time, the optimization problem defined by (6) has to be solved.

For systems that need fast updating periods, the updating time $\tau_u = \ell\tau$ may not be sufficient to reach the optimal solution (regardless of local minima). In such situations, the definition (6) is no more relevant. Indeed, one can only refer to an updating rule that involves a limited number q of iterations of some optimization subroutine $\mathcal{S}$. This implicitly assumes that q iterations of the subroutine $\mathcal{S}$ can be performed during the updating period $\ell\tau$. Denoting by τ_c the time needed to perform a single iteration, it comes that:

$$\ell = \ell(q) = int\left\lfloor \frac{q\tau_c}{\tau} \right\rfloor + 1 \quad (9)$$

where for any positive real s, $int(s)$ stands for the integer part of s. Based on the above discussion, in the sequel, ℓ can be viewed as a function of q for a given pair of measurement acquisition period τ and an optimization subroutine that defines the time τ_c on a specific hardware facility.

Fig. 2. The time structure of the updating scheme defined by (11)-(12).

Consequently, the decision variable p can be updated only at updating instants (see Figure 2):

$$t_k = t_{k-1} + \left[\ell(q(t_{k-1})) \right] \cdot \tau \tag{10}$$

This leads to the following updating rules:

$$p(t_k) = \mathcal{S}^{(q)}\left(p^*(t_k) \,|Y_m(t_{k-1}), \mathbf{u}(t_{k-1}) \right) \tag{11}$$

$$p^*(t_k) = X(\ell(t_{k-1}), p(t_{k-1}), \mathbf{u}(t_{k-1} - N\tau)) \tag{12}$$

Note that the definition of $p^*(t_k)$ involves an ℓ-step prediction map. Note that $p^*(t_k)$ is used as an initial guess for the iterations invoked in (11).

Figure 2 illustrates the temporal structure of the updating laws given by (11)-(12). It shows the following features:

- ✓ The shift of the observation horizon corresponds at instant t_k to $\ell(q(t_k))$ basic sampling periods leading to a temporal shift of $\ell(q(t_k))\tau$.

- ✓ The computation of $p(t_k)$ is done during the time interval $[t_{k-1}, t_k]$ based on the initial guess $p^*(t_k)$.

- ✓ Note that the updating of the state estimate can still be done at each basic sampling period using the last updated value of p using the following expressions:

$$\forall i \in \{1, \dots, \ell\}, \tag{13}$$

$$\hat{x}(t_{k-1} + i\tau) = X(\ell + N + i, p(t_{k-1}), \mathbf{u}(t_{k-1} + i\tau))$$

where $\ell = \ell(q(t_{k-1}))$.

Now given that $p^*(t_k)$ is defined in terms of $p(t_{k-1})$ and $\mathbf{u}(t_{k-1} - N\tau)$ [see (12)] and that the latter is contained in $\mathbf{u}(t_{k-1})$, one can write the evolution equation for the parameter vector p by combining (11)-(12) :

$$p(t_k) = G\left(p(t_{k-1}), q(t_{k-1}), \mathbf{w}(t_{k-1}) \right) \tag{14}$$

for a straightforward definition of G where $\mathbf{w}(t_{k-1})$ stands for the past measurement data, namely:

$$\mathbf{w}(t_{k-1}) := \left\{ Y_m(t_{k-1}), \mathbf{u}(t_{k-1}) \right\} \tag{15}$$

Note also that using the same notation, the value of the cost function at instant t_k can be written in the following form [according to (6)]:

$$J\left(p(t_k)|p(t_{k-1}), \mathbf{w}(t_{k-1}) \right) \tag{16}$$

and again, since $p(t_k)$ is given by (14), the cost function can be written in a more compact form:

$$J = h\left(p(t_{k-1}), q(t_{k-1}), \mathbf{w}(t_{k-1}) \right) \tag{17}$$

The discussion above enables to view the situation as the one in which there is a discrete-time dynamic system defined by (14) in which, the state is p, the control is given by q and the past measurement (including the control and the measured output) are exogenous non modeled signals while the control objective is to steer the output J given by (17) to its minimum value.

This is obviously a control problem (see Figure 3) in which the control variable is defined by q which is the number
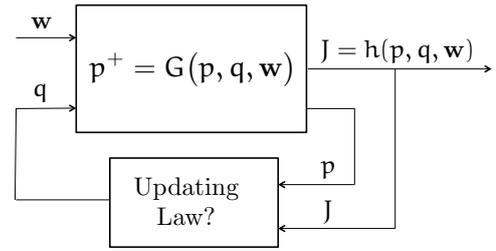
Fig. 3. The measurement inclusion rate problem viewed as an discrete-time output regulation problem with dynamic state p, control input q and a regulated output J.

of measurements that enter (and leave) the buffer of the MHO at the next shift of the observation interval.

Solving this control problem leads to an adaptive behavior of q that takes into account both the measurement buffer and the measured behavior of the cost function during the system lifetime. The decision variable q is called hereafter the *Measurement Inclusion Rate* as it defines the updating time $\ell(q)\tau$ during which no new measurements are accounted for and iterations are applied to the same cost function.

Remark 2. Note that the computations involved in the updating rule for q (the feedback law) must correspond to a negligible burden since the whole framework is supposed to compensate for the lack of computation time (see section 3.2).

3. UPDATING LAW FOR THE MEASUREMENT INCLUSION RATE

In order to simplify the expressions, the following short notation is used:

- ✓ $J(t_k)$ denotes the best obtained value of the cost function defined on the observation horizon $[t_{k-1} - N\tau, t_{k-1}]$ after q iterations.

- ✓ $J^*(t_k)$ denotes the value of the same cost function at the initial guess $p^*(t_k)$ that is computed according to (12).

Obviously, one key step toward understanding the convergence issue is to examine the ratio between the best obtained values of the cost function at two successive updating instants t_{k-1} and t_k, namely (see Figure 2):

$$\frac{J(t_k)}{J(t_{k-1})} = \underbrace{\frac{J(t_k)}{J^*(t_k)}}_{E(q, t_k)} \times \underbrace{\frac{J^*(t_k)}{J(t_{k-1})}}_{D(q, t_k)} \tag{18}$$

$$= E(q, t_k) \times D(q, t_k) =: K(q, t_k) \tag{19}$$

Note however that in order for the ratios involved in (18) to be well defined, the following easy-to-meet assumption is needed:

Assumption 1. There is a positive $c > 0$ such that for all t and all p, one has: $J(p, t) \geq c$. ♠

Note that this can be fulfilled by adding c to any original nonnegative cost function's definition.

The terms $E(q, t_k)$ and $D(q, t_k)$ are similar to the terms invoked in [Alamir, 2008, 2013] where the dual MPC problem is studied. More precisely:

- ✓ $E(q, t_k)$ is linked to the *local* efficiency of the optimizer's iterations as it represents the contraction of the cost function due to the execution of q successive iterations. This ratio is obviously lower or equal to 1.

- ✓ $D(q, t_k)$ is the ratio between the value of the cost function after horizon shift and using the model-based predicted value $p^*(t_k)$ that is compatible with the last achieved value $p(t_{k-1})$ (hot start). In the case where $p(t_{k-1})$ matches the true value of the state $x(t_{k-1} - N\tau)$ then in the ideal case (no model discrepancies and no measurement noise), the ratio $D(q, t_k)$ is equal to 1. Consequently, when this is not the case, $D(q, t_k)$ can be viewed as a *disturbance indicator* which gathers all the unavoidable above mentioned imperfections. Note also that since the length of the updating period $\tau_u = \ell\tau = int(q\tau_c/\tau)+1$ depends on q, the disturbances induced ratio $D(q, t_k)$ depends also on q.

Based on the discussion above, the following model of the disturbance ratio is used in the sequel:

$$D(q, t_k) = 1 + \alpha(t_k) \cdot q \qquad (20)$$

where $\alpha(t_k)$ is a parameter to be identified on-line as shown later.

Now, given the evolution equation:

$$J(t_k) = K(q, t_k) \cdot J(t_{k-1}) \qquad (21)$$

it becomes obvious that one rationale that can be used in the derivation of the updating law for q is to force the multiplicative gain K to be lower than one as this enhances the convergence of the cost function J and when this goal is achievable, q must be monitored so that the response time of the *closed-loop* continuous-time system (in the sense of Figure 3) is minimized.

These considerations lead to the following ideal updating law for q:

$$q(t_{k+1}) =$$
$$\underset{q \in \{q_{min}, \dots, q_{max}\}}{\arg\min} \begin{cases} \dfrac{q}{|\log(K(q, t_k))|} & \text{if } K(q(t_k), t_k) < 1 \\[2ex] K(q, t_k) & \text{otherwise} \end{cases} \qquad (22)$$

This is because the ratio:

$$\frac{q}{|\log(K(q, t_k))|}$$

is almost proportional (up to the *int* function discontinuity) to the response time of a discrete-time dynamics (21) that is characterized by the discrete pole K and the sampling time $\ell(q)\tau$ where $\ell(q)$ is given by (9). Note that $q_{min} \geq 2$ is systematically considered in order to be always capable of estimating the gradient of the cost function involved in (22) w.r.t q as it is explained in the sequel.

The exact solution of the optimization problem (22) would needs too many computations that would make the updating rule inappropriate (see Remark 2). Instead, an approximated gradient approach is used following the ideas proposed in [Alamir, 2013] in the case of MPC framework.

To do this, the sensitivity of $K(\cdot, t_k)$ w.r.t q is computed by computing those of $E(\cdot, t_k)$ and $D(\cdot, t_k)$ using the available algorithm data at the past updating instant. Then a quantized gradient step is applied to update the value of $q(t_{k+1})$. This is detailed in the following section.

3.1 Updating Algorithm

Note first of all that since the past value $q(t_k) \geq 2$, it is possible to compute the following approximation of the gradient of the efficiency map:

$$\frac{\Delta E}{\Delta q}(t_k) \approx \frac{J(p^{(q(t_k))}, t_{k-1}) - J(p^{(q(t_k)-1)}, t_{k-1})}{J(p^*(t_k), t_{k-1})} \qquad (23)$$

On the other hand, using the presumed structure (20) of D, one can use the available algorithm's data to compute an estimation of $\alpha(t_k)$ which is nothing but the gradient of D w.r.t q, that is:

$$\frac{\Delta D}{\Delta q}(t_k) \approx \frac{1}{q(t_k)} \left[\frac{J^*(t_k)}{J(t_{k-1})} - 1 \right] \approx \alpha(t_k) \qquad (24)$$

Now, using equations (23)-(24), the gradient of the multiplicative gain K w.r.t q can be computed according to:

$$\frac{\Delta K}{\Delta q}(t_k) \approx E(t_k)\frac{\Delta D}{\Delta q}(t_k) + D(t_k)\frac{\Delta E}{\Delta q}(t_k) \qquad (25)$$

and having this estimation of the gradient, the gradient of the cost function involved in the ideal updating rule (22) can be computed:

$$\frac{\Delta\big(q/|\log(K(t_k))|\big)}{\Delta q} \approx \frac{-\log(K(t_k)) + \dfrac{q}{K(t_k)} \times \dfrac{\Delta K(t_k)}{\Delta q}}{\big[\log(K(t_k))\big]^2} \qquad (26)$$

Using the above computed quantities, the following algorithm can be used to compute the updated value $q(t_{k+1})$:

Algorithm 1 Updating rule $q(t_{k+1}) = U(q(t_k), t_k)$

1: **If** $(K(t_k) \geq 1)$ **then**
2: $\quad \Gamma \leftarrow \dfrac{\Delta K}{\Delta q}(q(t_k))$ $\quad$ [see (23)-(24) and (25)]
3: **Else**
4: $\quad \Gamma \leftarrow \dfrac{\Delta(q/|\log(K(t_k))|)}{\Delta q}$ $\quad$ [see (26)]
5: **End If**
6: $q(t_{k+1}) \leftarrow \max\Big\{q_{min}, \min\big\{q_{max}, q(t_k) - \delta \cdot sign(\Gamma)\big\}\Big\}$

In this algorithm, Γ represents the gradient of the quantity to be minimized. A quantized step ($\delta \in \mathbb{N}$) in the opposite direction is implemented in Step 6 to update the value of q.

3.2 Complexity Analysis

Based on the expressions (23)-(26), one can construct Table 1 that shows the additional number of elementary operations that are needed to compute the updating law involved in Algorithm 1. By *additional*, it is meant that the computation of the cost function $J(t_k)$, $J^*(t_{k-1})$ and $J(t_{k-1})$) are excluded because they are by-products of the optimization process and are therefore computed regardless of whether an updating process is used or not.

Expression	Equation	$(\pm)$	$(\times)$	$(\div)$	log
K	(19)		2	1	
$(\Delta E)/(\Delta q)$	(23)	1		1	
$(\Delta D)/(\Delta q)$	(24)	1		2	
$(\Delta K)/(\Delta q)$	(25)	3	2	3	
Γ	(26)	4	5	6	1
Algorithm 1		**5**	**5**	**6**	**1**

Table 1. Elementary computations involved in the updating law $q(t_{k+1}) = U(q(t_k), t_k)$.

Table 1 clearly shows that the computations involved in the updating rule involves only few arithmetic operations and a single logarithm computation).

4. ILLUSTRATIVE EXAMPLE

Let us consider the following modified van-der-pol system:

$$\dot{x}_1 = x_2$$
$$\dot{x}_2 = -ax_1 + (1 - ux_3x_1^2)x_2$$
$$\dot{x}_3 = 0$$

The measured output is defined by:

$$y = x_1 + \nu$$

where ν is a white noise. The parameter a is a parameter that can be badly known in order to enhance the uncertainty feature of the model. The basic sampling (measurement acquisition) period is taken equal to $\tau = 2\ ms$.

The optimization subroutine involved in (11) is based on a fast gradient approach [Nesterov, 1983, 2004] with restart mechanism [O'Donoghue and Candes, 2012]. The details of these algorithms can be found also in [Alamir, 2013]. The explicit definition of these algorithms is not mandatory here since the proposed approach is generic and the use of fast gradient approach as an instantiation of $\mathcal{S}$ is only a matter of choice that enables the main idea to be illustrated. Note that the optimization is performed assuming the following box constraints that are to be interpreted component-wise:

$$\begin{pmatrix} -10 \\ -10 \\ 0.1 \end{pmatrix} \leq x \leq \begin{pmatrix} 10 \\ 10 \\ 40 \end{pmatrix} \tag{27}$$

The computation time τ_c needed to perform a single iteration of the subroutine $\mathcal{S}$ [see (9)] is given by $\tau_c = 500\ \mu sec$ The minimum and the maximum number of iterations involved in the updating rule appearing in the step 6. of Algorithm 1 are $q_{min} = 20$ and $q_{max} = 1000$.

The observation horizon is taken equal to $N = 200$ basic sampling period leading to a time window of $T = 0.4\ sec$. The variance of the measurement noise is taken equal to $\nu = 0.03$. The cost function $J(p)$ that has been used in all the simulations takes the following form at instant k:

$$J := \sum_{i=0}^{N} \|\hat{y}(k + i|p) - y(k + i)\|^2 + \rho\|p - \hat{p}\|^2$$

where $\hat{p}$ is the estimate based on the past estimation of p. The coefficient $\rho = 0.01$ is used in the sequel.

In the following sections, the validation scenario is clearly stated and the comparison indicators are defined.

4.1 Validation Scenarios

For each comparison, $N_s := 50$ scenarios are executed using different values of the initial estimated state $\hat{x}_0$. These values are randomly chosen according to:

$$\hat{x}^0 := (0.2 \times \mathbb{I} + 1.8 \begin{pmatrix} r_1 & 0 & 0 \\ 0 & r_2 & 0 \\ 0 & 0 & r_3 \end{pmatrix})x^0 \tag{28}$$

where $x^0 = (3, 1, 1)^T$ is the true initial state. r_i are uniformly distributed random variable belonging to the interval $[0, 1]$. Note that (28) simply means that each component $\hat{x}_i^0$ of the initial state of the observer is randomly chosen in the interval $[0.2x_i^0, 2x_i^0]$. The resulting set of observer initial states is denoted hereafter by $\hat{\mathbb{X}}^0$.

Simulations are performed during $N_{sim} = 2000$ sampling periods (4 *sec*) using the following input profile:

$$u(t) := 1 - \frac{1}{2}cos(2t)$$

Five observer settings are compared which are:

(1) **Setting 1**. $q = q_{min} = 20$. No updating is used.
(2) **Setting 2**. $q = q_{min} = 50$. No updating is used.
(3) **Setting 3**. $q = q_{min} = 100$. No updating is used.
(4) **Setting 4**. $q = q_{min} = 300$. No updating is used.
(5) **Setting 4**. $q(0) = q_{min} = 20$. Updating is used for q using the rule expressed in Algorithm 1 starting from the initial value $q(0) = q_{min} = 20$ and the increment size $\delta = 10$. Note that the first four settings can be obtained using $\delta = 0$ while initializing q to 20, 50, 100 and 300 respectively.

Note that for each of the above mentioned five settings, the same 50 trials leading to the set $\hat{\mathbb{X}}^0$ are used. That is, the trials are first done and then the 50 simulated scenarios are executed for each of the five settings leading to 250 simulations. This avoid biased comparison that may be due to different trials for each setting.

4.2 Performance indicator

As mentioned in Remark 1, we are interested in the behavior of the cost function. That is the reason why

the characteristics (mean and variance) of the following quantities are monitored:

$$\hat{J}^{(s)}(k|\hat{x}^0) \text{ for } k = 1, \ldots, N_{sim} - N \text{ and } \hat{x}^0 \in \hat{\mathbb{X}}^0$$

where $\hat{J}^{(s)}(k|\hat{x}^0)$ is the cost function at instant k for the scenario starting at $\hat{x}^0 \in \hat{\mathbb{X}}^0$ and using the setting number s.

More precisely, for easiness of comparison, the following quantities are considered for all setting index s:

$$m^{(s)} := \text{Mean}_{k,\hat{x}^0} \left[\frac{J^{(s)}(k|\hat{x}^0) - J^{(1)}(k|\hat{x}^0)}{J^{(1)}(k|\hat{x}^0)} \right] \quad (29)$$

$$\sigma^{(s)} := \text{Var}_{k,\hat{x}^0} \left[\frac{J^{(s)}(k|\hat{x}^0) - J^{(1)}(k|\hat{x}^0)}{J^{(1)}(k|\hat{x}^0)} \right] \quad (30)$$

where the mean and the variance invoked in (29) and (30) are taken on the set of values of $(k, \hat{x}^0)$ given by:

$$\left\{ 1, \ldots, N_{sim} - N \right\} \times \hat{\mathbb{X}}^0$$

Two scenarios are used to assess the relevance of the updating rule. In the first the parameter $a = 10$ is perfectly known by the observer while in the second scenario, the observer uses an erroneous value $\hat{a} = 7$. Figure 4 shows the corresponding performance of the different observers using circles that are centered at $m^{(s)}$, $s = 1, \ldots, 5$ and with radius that are equal to twice the corresponding variance $\sigma^{(s)}$. In the first case, an optimal updating period seems to be close to $q = 100$ (see Figure 4.a). However, when model uncertainties increase (as in the second scenario), the optimal updating period becomes the minimum value $q = q_{min}$. Note that in both cases, the performance of the updated version systematically spots close to the optimal performance. Note also that if the parameter a changes during the scenario, then, the observer with the proposed updating scheme would likely be better that any observer with constant updating period.

5. CONCLUSION AND FUTURE WORK

In this paper, a novel updating rule for the measurement inclusion rate in MHO is proposed and validated through a simple example. The proposed rule enables near optimal performance to be achieved in presence of unavoidable, unpredictable model discrepancies and without off-line extended tuning. Current investigation focuses on the validation of the proposed methodology on real-world estimation problems.

REFERENCES

M. Alamir. Optimization based nonlinear observers revisited. *International Journal of Control*, 72 (13):1204–1217, 1999.

M. Alamir. *A Framework for Monitoring Control Updating Period in Real-Time NMPC*, chapter In Assessement and Future Directions in Nonlinear Model Predictive Control. Lecture Notes in Control and Information Sciences, Springer-Verlag,, 2008.

M. Alamir. Monitoring control updating period in fast gradient-based NMPC. In *Proceedings of the European Control Conference (ECC2013), Zurich, Switzerland*, 2013.
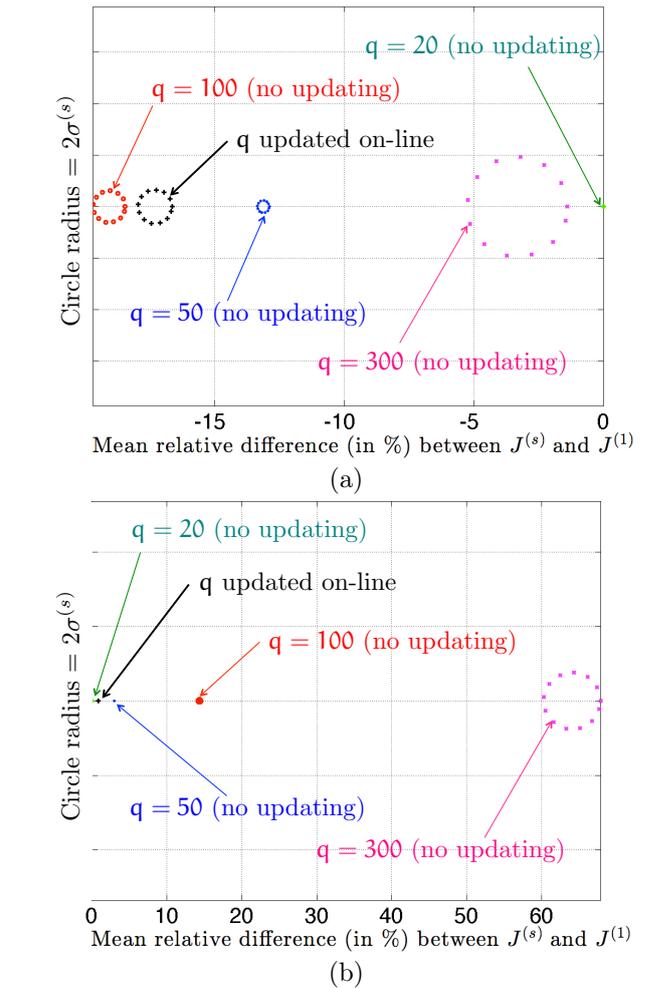
Fig. 4. Comparison of the performance of the observer under different constant number of iterations $q \in \{20, 50, 100, 300\}$ on one side and under the proposed updating scheme that start at $q(0) = 20$. **(a)** Case where the observer knows the exact value of the parameter $a = 10$. **(b)** Case where the observer uses an erroneous value $\hat{a} = 7$ instead of $a = 10$. The radius of each circle is equal to twice the corresponding variance.

P. Kuhl, M. Diehl, T. Kraus, J. P. Schloder, and H. G. Bock. A real-time algorithm for moving horizon state and parameter estimation. *Computers and Chemical Engineering*, 35(1):71–83, 2011.

H. Michalska and D. Q. Mayne. Moving-horizon observers and observer-based control. *IEEE Transactions on Automatic Control*, 40:995–1006, 1995.

Y. Nesterov. A method of solving a convex programming problem with convergence rate o (1/k2). *Soviet Mathematics Doklady*, 27(2):372–376, 1983.

Y. Nesterov. *Introductory lectures in convex optimization: a basic course*. Kluwer Academic Publishers, 2004.

B. O'Donoghue and A. Candes. Adaptive restart for accelerated gradient schemes. arxiv:1204.3982. April 2012.

C. V. Rao, J. B. Rawlings, and D. Q. Mayne. State Constrained Estimation for Nonlinear Discrete-Time Systems: Stability and Moving Horizon Approximations. *IEEE Transactions on Automatic Control*, 48(2), 2003.