A FAST COMPUTATIONAL FRAMEWORK FOR LARGE-SCALE MOVING HORIZON ESTIMATION

Victor M. Zavala, Carl D. Laird and Lorenz T. Biegler

Chemical Engineering Department
Carnegie Mellon University
Pittsburgh, PA USA 15213
{*vzavala,claird,lb01*}*@andrew.cmu.edu*

Abstract: Moving Horizon Estimation (MHE) is an efficient optimization-based strategy for state estimation. Despite the attractiveness of this method, its application in industrial settings has been rather limited. This has been mainly due to the difficulty to solve, in real-time, the associated dynamic optimization problems. In this work, a fast MHE algorithm able to overcome this bottleneck is proposed. The framework exploits the advantages of simultaneous collocation-based formulations and makes use of large-scale nonlinear programming algorithms and sensitivity concepts. The approach is demonstrated on a full-scale polymer process, where accurate state estimates are obtained and on-line calculation times are reduced dramatically. *Copyright* © *2007 IFAC*

Keywords: estimation algorithms, real-time, discretization, finite elements, nonlinear programming, sensitivity analysis

1. INTRODUCTION

Moving horizon estimation has been identified as an efficient method for state estimation for constrained, linear and nonlinear systems. From a theoretical point of view, a deeper understanding of the estimator filtering and stability properties have led to efficient formulations (Robertson *et al.*, 1996; Rao *et al.*, 2003; Rawlings and Bakshi, 2006). In addition to this, increased interest has resulted from its proven superiority over traditional estimation approaches such as the extended Kalman filter (Haseltine and Rawlings, 2005). From a computational point of view, on-line implementations of MHE in industrial settings still represent a challenge. In NMPC strategies, the estimated state of the system is required for the solution of the NMPC or *regulator* problem from which feedback is obtained. An important observation is that, in order to retain the stabilizing properties of the controller, both the MHE and NMPC problems should be solved in real-time (Findeisen and Allgöwer, 2004; Diehl *et al.*, 2005*b*). However, this is not currently possible in most practical applications. Because of this, several alternatives have been explored in trying to overcome this computational burden. A key conceptual breakthrough has been the idea of separating the computational tasks into a preparation and a feedback phase (Diehl *et al.*, 2005). Some variants of this idea have been proposed for NMPC (Diehl *et al.*, 2005; Kadam and Marquardt, 2004; Dehaan and Guay, 2006; Zavala *et al.*, 2006) and, more recently, for MHE (Kraus *et al.*, 2006). These approaches mainly differ on the numerical methods used for the solution of the dynamic optimization problem. In this work, the fast computational framework for NMPC presented in (Zavala *et al.*, 2006) is extended to consider MHE problems. The framework exploits the advantages of simultaneous collocation-based (or direct transcription) formulations. In addition, it makes use of efficient large-scale NLP algorithms and builds on well-established NLP sensitivity concepts. The next section presents the MHE problem under consideration. The proposed fast MHE algorithm is described in detail in Section 3. The algorithmic framework is then applied to an industrial polymer process and the results are presented in Section 4 while the last section concludes the paper.

2. MHE PROBLEM

Consider the scenario in which a given dynamic system is located at time t_ℓ. A general MHE problem consists of obtaining an estimate of the current state of the system z_ℓ and of the parameters π given a set of past measurements $\{\bar{y}_{\ell-N}, \bar{y}_{\ell-N+1}, ..., \bar{y}_\ell\}$. Here, the measurements are distributed along a horizon comprised of N sampling times. For the sake of simplicity and without loss of generality, we consider only measurement noise and that z_ℓ and π can be obtained from the solution of an optimization problem of the form:

$$\begin{aligned}\overset{\min}{z_0, \pi} \quad & \frac{1}{2}(\|z_0 - \bar{z}_0\|^2_{\mathbf{W}_z} + \|\pi - \bar{\pi}\|^2_{\mathbf{W}_\pi}) \\ & \qquad + \sum_{k=0}^{N} \frac{1}{2}\|y_k - \bar{y}_{k+\ell-N}\|^2_{\mathbf{W}_y} \\ \text{s.t.} \quad & z_{k+1} = z_k + \mathbf{B}w_k, \ {}_{k=0,\dots,N-1} \\ & h(z_k, w_k, u_k, \pi) = 0, \ {}_{k=0,\dots,N-1} \\ & y_k - g(z_k) = 0, \ {}_{k=0,\dots,N} \qquad (1)\end{aligned}$$

In the following, this problem is denoted as $\mathbf{M}(\ell)$ and the current state of the system is given by $z_\ell := z_N^*$. Here, $z_k \in \Re^{n_z}$ is the vector of state variables at the k-th sampling time, $u_k \in \Re^{n_u}$ is the vector of past control variables, $y_k \in \Re^{n_y}$ is the vector of estimated output variables and $w_k \in \Re^{n_w}$ is a vector of algebraic variables. Matrix $\mathbf{B}$ projects a subset of the algebraic variables into the evolution of the states. Accordingly, this formulation allows general Runge-Kutta discretizations, including multiple shooting or collocation on finite elements. The first term of the objective function is an *arrival cost* summarizing the effect of past information before $k = 0$. The weighting matrices $\mathbf{W}_z$, $\mathbf{W}_\pi$ represent the inverse of the covariance matrices of the *a priori* state and parameter estimates $\bar{z}_0$ and $\bar{\pi}$, respectively. Finally, it is assumed that all bounds and inequalities can be incorporated in the objective function using appropriate barrier terms.

Once the current state z_ℓ has been estimated from problem (1), it is used for the solution of the NMPC problem. The system then evolves to the next sampling time. At this point, having a new measurement $\bar{y}_{\ell+1}$, the oldest measurement $\bar{y}_{\ell-N}$ is dropped and the following MHE problem $\mathbf{M}(\ell+1)$ is solved over a shifted horizon. Notice that, in practical applications, the solution of $\mathbf{M}(\ell+1)$ cannot be obtained instantaneously. Therefore, if the solution process is started until the new measurement becomes available, a delay would be introduced and propagated over the solution of the NMPC problem, leading to a long feedback delay. To address this issue, a fast MHE algorithm based on a real-time iteration approach is presented; at time t_ℓ, a *nominal* problem $\mathbf{M}(\ell)$ is solved during a preparation or background phase without considering future measurements at $t_{\ell+1}$. Once the measurements become available, a first-order correction is performed around the nominal problem to obtain a fast approximate solution to the neighboring problem $\mathbf{M}(\ell+1)$.

3. FAST MHE ALGORITHM

In order to develop the fast computational framework, we first consider methods for the solution of the MHE problem resulting from simultaneous collocation-based formulations. This gives rise to a large-scale but *sparse* representation of NLP (1) that can be solved using full-space NLP algorithms with exact derivative information. A careful analysis in (Zavala *et al.*, 2006) has shown that the approach enjoys favorable computational complexity when compared against competing approaches (scales better with problem size and number of degrees of freedom). This makes it attractive for large-scale parameter and state estimation problems. On the other hand, it heavily relies on the use of efficient and reliable NLP algorithms.

3.1 NLP Algorithm and Sensitivity

MHE problem (1) can be posed in the general form:

$$\begin{aligned}\overset{\min}{x} \quad & f(x, p) \\ \text{s.t.}\ & c(x, p) = 0 \\ & x \geq 0 \qquad (2)\end{aligned}$$

where x includes all the variables in problem (1) and symbol p denotes a *general* parameter vector. In the context of MHE, problem (2) is usually a large-scale, nonconvex NLP problem with many inequalities and degrees of freedom. For the solution of problems with these characteristics, interior-point methods have been recently shown to be robust and reliable (Forsgren *et al.*, 2002). In this work, the IPOPT algorithm (Wächter and Biegler, 2006) is used for the solution of NLP (2). The algorithm follows a barrier approach, where the bound constraints are replaced by logarithmic barrier terms and added to the objective function to give:

$$\begin{aligned} \min_{x} \quad & f(x,p) - \mu \sum_{i=1}^{n} \ln(x^{(i)}) \\ \text{s.t.} \quad & c(x,p) = 0 \end{aligned} \qquad (3)$$

with a barrier parameter $\mu > 0$. Here, $x^{(i)}$ denotes the i-th component of the vector x. The degree of influence of the barrier is determined by the size of μ, and under mild conditions $x^*(\mu)$ converges to a local solution x^* of the original problem (2) as $\mu \to 0$. Consequently, a strategy for solving the original NLP is to solve a sequence of barrier problems (3), with index l, for decreasing values of μ_l.

IPOPT follows a full-space, primal-dual barrier approach and applies a Newton method to the KKT conditions derived from (3), leading to the solution of the following sparse KKT system at iteration j:

$$\begin{bmatrix} W_j & A_j & -I \\ A_j^T & 0 & 0 \\ V_j & 0 & X_j \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta\lambda \\ \Delta\nu \end{bmatrix} = - \begin{bmatrix} \nabla f(x_j,p) + A_j\lambda_j - \nu_j \\ c(x_j,p) \\ X_j V_j e - \mu_l e \end{bmatrix} \qquad (4)$$

where $X = diag(x)$, $V = diag(\nu)$, W_j is the Hessian of the Lagrangian function $\nabla_{xx}(f(x_j,p) + c(x_j,p)^T\lambda_j)$, and $A_j = \nabla_x c(x_j,p)$. The factorization of the KKT matrix in the above linear system represents the most expensive step in the algorithm. Exact first and second derivatives for this method can be evaluated in a number of ways, including automatically through modeling platforms. As a result, local convergence of Newton's method is fast and global convergence is promoted by a novel filter line search strategy.

The KKT system (4) can be represented in compact form as $\mathbf{K}\Delta v = \varphi(v,\mu_l,p)$ where $v^T = [x^T\ \lambda^T\ \nu^T]$. Assume that problem (2) has been solved for a nominal parameter vector p_0. Therefore, at the solution $\varphi(v^*,\mu_l,p_0) = 0$. If v^* is a KKT point satisfying strong second order conditions it can be shown that, for a sufficiently small μ_l, the solution of $\mathbf{K}\Delta v = \varphi(v^*,\mu_l,p)$ provides a first-order approximation to the solution of (2) with $p = p_0 + \Delta p$ (Fiacco, 1983). Since exact derivative information is used in this case, the approximation can be shown to be $O(||p - p_0||^2) + O(||\mu_l||)$ which allows to establish a rigorous bound on the loss of optimality. Additionally, $\mathbf{K}$ is already factorized at the solution of the nominal problem. Therefore, these sensitivity calculations are very cheap and thus represent an essential component of the fast computational framework.

Notice that in the context of parameter and state estimation, second order conditions may not hold at the solution. This will occur if the parameters or the state of the system are not observable given the available data. In this case, the KKT matrix will be said to have wrong inertia (number of positive, negative and zero eigenvalues) at the solution (Zavala and Biegler, 2006). On the other hand, if the inertia is correct, this gives the important result that the parameters and/or the state of the system are observable. Since most of the modern direct linear solvers provide the inertia of the KKT matrix as an outcome of the factorization procedure, checking for the observability condition comes at no additional expense even for large-scale estimation problems.

3.2 Shifting Strategy

At time t_ℓ we obtain the current measurements $\{\bar{y}_{\ell-N}, \bar{y}_{\ell-N+1}, \ldots, \bar{y}_\ell\}$ to formulate the MHE problem $\mathbf{M}(\ell)$. Having the solution of $\mathbf{M}(\ell)$ it is desired to obtain a fast approximation to the solution of $\mathbf{M}(\ell{+}m)$, where m denotes the number of sampling times required to obtain the exact solution of $\mathbf{M}(\ell)$ (background or preparation tasks). In order to account for the yet unknown future measurements $\{\bar{y}_{\ell+1}, ..., \bar{y}_{\ell+m}\}$, the MHE problem (1) is modified as,

$$\begin{aligned} \min_{z_0,\pi,\hat{y}_k} \quad & \frac{1}{2}(\|z_0 - \bar{z}_0\|^2_{\mathbf{W}_z} + \|\pi - \bar{\pi}\|^2_{\mathbf{W}_\pi}) \\ & + \sum_{k=0}^{N} \frac{1}{2}\|y_k - \bar{y}_{k+\ell-N}\|^2_{\mathbf{W}_y} \\ & + \sum_{k=N+1}^{N+m} \frac{1}{2}\|y_k - \hat{y}_k\|^2_{\mathbf{W}_y} \\ \text{s.t.} \quad & z_{k+1} = z_k + \mathbf{B}w_k, \ {\scriptstyle k=0,\ldots,N+m-1} \\ & h(z_k,w_k,u_k,\pi) = 0, \ {\scriptstyle k=0,\ldots,N+m-1} \\ & y_k - g(z_k) = 0, \ {\scriptstyle k=0,\ldots,N+m} \end{aligned} \qquad (5)$$

where the estimation horizon has been extended using m additional sampling times. Notice the introduction of dummy variables $\hat{y}_k \in \Re^{n_y}$, $k = N+1, \ldots, N+m$ and that, since these dummy measurements are set free, they will exactly match the model predictions over the future sampling times ($\hat{y}_k^* = y_k^*$ holds at the solution of this problem). Therefore, the proposed modifications do not alter the solution of the problem. Also notice that, in order to formulate $\mathbf{M}(\ell)$ at t_ℓ, we assume that the future control sequence $\{u_N, \ldots, u_{N+m}\}$ is known as an outcome of the NMPC problem and given by $\{u_\ell, \ldots, u_{\ell+m}\}$. Figure 2 presents a

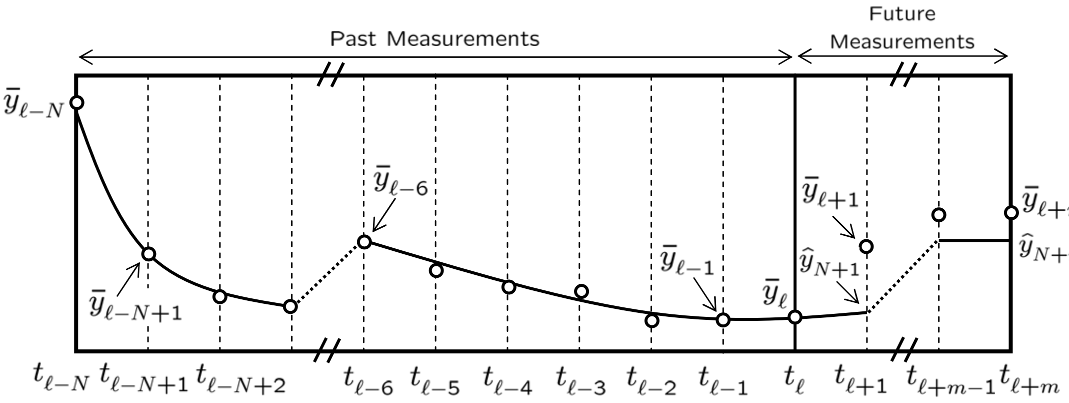

Fig. 1. Schematic representation of estimation horizon for nominal problem $\mathbf{M}(\ell)$.

schematic representation of the prediction horizon of the nominal problem.

Having the solution of $\mathbf{M}(\ell)$ at time $t_{\ell+m}$, we obtain a fast approximation to the solution of $\mathbf{M}(\ell+m)$ without altering the structure of the nominal problem. Preserving the structure of $\mathbf{M}(\ell)$ is required in order to reuse the already computed factors of the KKT matrix $\mathbf{K}_\ell$ available at the solution of $\mathbf{M}(\ell)$. For this, it is necessary to derive a suitable sensitivity approach. The derivation of the proposed approach starts by recalling that the Lagrange function associated to the new augmented problem $\mathbf{M}(\ell)$ in (5) is given by,

$$\begin{aligned}\mathbf{L} = &\sum_{k=0}^{N}\left(\frac{1}{2}\|y_k-\bar{y}_{k+\ell-N}\|^2_{\mathbf{W}_y}+\gamma_k^T(y_k-g(z_k))\right)\\ &+\sum_{k=N+1}^{N+m}\left(\frac{1}{2}\|y_k-\hat{y}_k\|^2_{\mathbf{W}_y}+\gamma_k^T(y_k-g(z_k))\right)\\ &+\sum_{k=0}^{N+m-1}\left(\lambda_{k+1}^T(z_{k+1}-z_k-\mathbf{B}w_k)+\eta_k^T h(z_k,w_k,u_k,\pi)\right)\\ &+\frac{1}{2}\left(\|z_0-\bar{z}_0\|^2_{\mathbf{W}_z}+\|\pi-\bar{\pi}\|^2_{\mathbf{W}_\pi}\right) \qquad (6)\end{aligned}$$

and we obtain the desired sensitivity equations upon linearization of the first-order optimality conditions around the solution of $\mathbf{M}(\ell)$,

$$\begin{aligned} z_0:\quad & (\mathbf{W}_z+H^0_{zz}-G^0_{zz})\Delta z_0+H^0_{zw}\Delta w_0-\Delta\lambda_1\\ & +\nabla_z h_0\Delta\eta_0-\nabla_z g_0\Delta\gamma_0+H^0_{z\pi}\Delta\pi=0 \qquad (7)\\ z_k:\quad & \Delta\lambda_k-\Delta\lambda_{k+1}+(H^k_{zz}-G^k_{zz})\Delta z_k+H^k_{zw}\Delta w_k\\ & +\nabla_z h_k\Delta\eta_k-\nabla_z g_k\Delta\gamma_k+H^k_{z\pi}\Delta\pi=0 \qquad (8)\\ & k=1,\dots,N+m-1\\ w_k:\quad & H^k_{wz}\Delta z_k+H^k_{ww}\Delta w_k+\nabla_w h_k\Delta\eta_k\\ & +H^k_{w\pi}\Delta\pi-\mathbf{B}^T\Delta\lambda_{k+1}=0 \qquad (9)\\ \lambda_k:\quad & \Delta z_{k+1}-\Delta z_k-\mathbf{B}\Delta w_k=0 \qquad (10)\end{aligned}$$

$$\begin{aligned} \eta_k:\quad & \nabla_z h_k^T\Delta z_k+\nabla_w h_k^T\Delta w_k+\nabla_\pi h_k^T\Delta\pi=0 \qquad (11)\\ \gamma_k:\quad & \Delta y_k-\nabla_z g_k^T\Delta z_k=0 \qquad (12)\\ & k=0,\dots,N+m-1\\ y_k:\quad & \mathbf{W}_y\Delta y_k+\Delta\gamma_k=0 \qquad (13)\\ & k=1,\dots,N\\ z_{N+m}:\quad & \Delta\lambda_{N+m}-\nabla_z g_{N+m}\Delta\gamma_{N+m}\\ & +G_{zz}\Delta z_{N+m}=0 \qquad (14)\\ \gamma_{N+m}:\quad & \Delta y_{N+m}-\nabla_z g_{N+m}\Delta z_{N+m}=0 \qquad (15)\\ \pi:\quad & \mathbf{W}_\pi\Delta\pi+\sum_{k=0}^{N+m-1}(\nabla_\pi h_k\Delta\eta_k+H^k_{\pi z}\Delta z_k)\\ & +\sum_{k=0}^{N+m-1}(H^k_{\pi w}\Delta w_k+H^k_{\pi\pi}\Delta\pi)=0 \qquad (16)\\ y_k:\quad & \mathbf{W}_y(\Delta y_k-\Delta\hat{y}_k)+\Delta\gamma_k=0 \qquad (17)\\ \hat{y}_k:\quad & \mathbf{W}_y(\Delta y_k-\Delta\hat{y}_k)=0 \qquad (18)\\ & k=N+1,\dots,N+m\end{aligned}$$

where $G^k_{zz}=\nabla_{zz}(g_k^T\gamma_k)$ and $H^k_{zz}=\nabla_{zz}(h_k^T\eta_k)$, etc. To estimate a solution for problem $\mathbf{M}(\ell+1)$ at the next sampling time, it is assumed that the term $\|y_0-\bar{y}_{\ell-N}\|^2_{\mathbf{W}_y}$ contributes to the arrival cost in the next horizon. Also, at sampling times $t_{\ell+m}$ the addition of m new measurements is considered. For this, the definition of $\hat{y}_k$ is modified by adding the equations:

$$\begin{aligned}\Delta\hat{y}_k &= \bar{y}_{k+\ell-N}-\hat{y}^*_k\\ k &= N+1,\dots,N+m \qquad (19)\end{aligned}$$

where $\hat{y}^*_k$ is obtained from the solution of $\mathbf{M}(\ell)$. In order to capture the effect of this perturbation, condition (18) needs to be *relaxed* by adding new extra slack variables σ_{k-N}, i.e.:

$$\begin{aligned}\mathbf{W}_y(\Delta y_k-\Delta\hat{y}_k)+\sigma_{k-N} &= 0\\ k &= N+1,\dots,N+m \qquad (20)\end{aligned}$$

where notice that these extra variables play the role of Lagrange multipliers. Adding the new equations (19) and variables in (20) leads to an augmented set of NLP sensitivity equations, which, in terms of the general problem (2) can be expressed in the condensed form:

$$\left[\begin{array}{c|c}\mathbf{K}_\ell & E_{s,m} \\ \hline E_m^T & 0\end{array}\right]\left[\begin{array}{c}\Delta v \\ \hline \Delta p\end{array}\right] = \left[\begin{array}{c}0 \\ \hline \mathbf{r}_m\end{array}\right] \tag{21}$$

where Δv represents the deviation from the solution v_ℓ^* of $\mathbf{M}(\ell)$ generated by the perturbation Δp. The perturbed right-hand sides are given by,

$$\mathbf{r}_m^T = [(\bar{y}_{\ell+1} - \hat{y}_{N+1}^*)^T, \ldots, (\bar{y}_{\ell+m} - \hat{y}_{N+m}^*)^T] \tag{22}$$

The extra variables are condensed in vector,

$$\Delta p^T = [\Delta\sigma_1^T, \ldots, \Delta\sigma_m^T]. \tag{23}$$

In other words, Δp represents the perturbation required in problem $\mathbf{M}(\ell)$ to force the dummy variables $\hat{y}_k$ to match the plant measurements. Matrix $E_{s,m}^T = [0 \ \ldots I_{m\times n_y} \ \ldots 0]$ *places* the extra slack variables in the constraints of problem (2) corresponding to (20) and $E_m^T = [0 \ \ldots I_{m\times n_y} \ldots 0]$ *extracts* the dummy variables $\hat{y}_k$ from the primal-dual vector v of problem (2) to generate the additional constraints (19).

The previous factorization of $\mathbf{K}_\ell$ from $\mathbf{M}(\ell)$ is used to shift to problem $\mathbf{M}(\ell + m)$ by solving for the perturbation Δp. To reuse the factorization of $\mathbf{K}_\ell$, this system could be solved with a Schur complement approach, i.e.,

$$\mathbf{S}_\ell \, \Delta p = -(E_m^T \mathbf{K}_\ell^{-1} E_{s,m})\Delta p = \mathbf{r}_m. \tag{24}$$

The construction of the Schur complement $\mathbf{S}_\ell \in \Re^{(m\times n_y)\times(m\times n_y)}$ requires $m \times n_y$ backsolves with $\mathbf{K}_\ell$ in background. Although these backsolves are easily parallelized, this calculation may be expensive for systems with many outputs. On the other hand, since the Schur complement is a relatively small dense matrix, it can be factorized efficiently using standard dense linear solvers (i.e., from the LAPACK library).

Once Δp has been found, an approximation to the solution $v_{\ell+m}^*$ of problem $\mathbf{M}(\ell + m)$ is obtained from the perturbed KKT system,

$$\mathbf{K}_\ell \Delta v = -E_{s,m}^T \Delta p \tag{25}$$

where $v_{\ell+m}^* \approx v_\ell^* + \Delta v$ contains the current state estimate $z_{\ell+m} = z_{N+m}^*$. Figure 2 presents a schematic representation of the fast MHE algorithm for the case $m = 1$. Notice how the background computational tasks are delayed by $m = 1$ sampling times so that the state $z_{\ell+1}$ is estimated instantaneously on-line from the sensitivity approach around the previous solution v_ℓ^*. With this, we obtain the fast MHE algorithm summarized below.

Algorithm 1 - Fast MHE Algorithm

```
Start clock variable $\theta = 0$.
while $\theta > 0$ do
  for $j = 1, ..., m$ do
    if $\theta = t_\ell$ then
      On-line: Obtain $\bar{y}_\ell$ from plant.
      if $j = m$ then
        On-line: Retrieve solution $v_{\ell-m}^*$, factorization of $\mathbf{K}_{\ell-m}$ and $\mathbf{S}_{\ell-m}$ from the solution of $\mathbf{M}(\ell - m)$ in background.
        On-line: Obtain approximate solution $v_\ell^*$ from shifting approach.
        1) Define
           $\mathbf{r}_m^T = [(\bar{y}_{\ell-m+1} - \hat{y}_{N+1}^*)^T, \ldots, (\bar{y}_\ell - \hat{y}_{N+m}^*)^T]$
        2) Solve
           $\mathbf{S}_{\ell-m}\Delta p = -\mathbf{r}_m$
        3) Compute
           $\mathbf{K}_{\ell-m}\Delta v = -E_{s,m}^T \Delta p$
        4) Set $v_\ell^* \approx v_{\ell-m}^* + \Delta v$ and extract current estimate $z_\ell$ for NMPC problem.

        Background: Start solution of $\mathbf{M}(\ell)$ using the approximate solution $v_\ell^*$ as warm-start. Upon convergence,
        1) Store factors of $\mathbf{K}_\ell$
        2) Construct $\mathbf{S}_\ell = -(E_m^T \mathbf{K}_\ell^{-1} E_{s,m})$ using the factorization of $\mathbf{K}_\ell$.
        3) Factorize $\mathbf{S}_\ell$
      end if
    end if
    $\ell \leftarrow \ell + 1$
  end for
end while
```

Remark 1: Consider the case where the solution of $\mathbf{M}(\ell)$ in background requires $m > 1$ steps to be completed. Here, the proposed MHE algorithm allows only to provide an instantaneous estimate at time $t_{\ell+m}$ with no action taken at intermediate times $t_{\ell+j}$, $j = 1, ..., m - 1$. In other words, this would be equivalent to have a longer sampling time and set $m = 1$. In order to avoid this and keep short sampling times, it is possible to solve $\mathbf{M}(\ell)$ using $2m - 1$ additional time steps in order to accommodate the future measurements at $t_{\ell+j}$, $j = 1, ..., m, ..., 2m-1$. With this, once the solution is available at $t_{\ell+m}$, it would be possible to provide instantaneous estimates at $t_{\ell+j}$, $j = m, ..., 2m - 1$ while a new problem $\mathbf{M}(\ell + m)$ is initialized and solved over this time frame. This scenario is illustrated in Figure (3) for the case

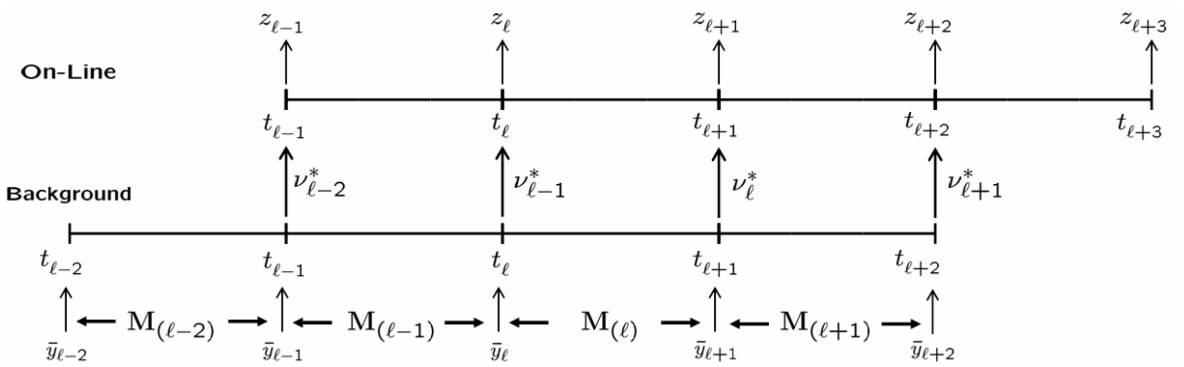

Fig. 2. Schedule of on-line and background computional tasks (Case $m = 1$).

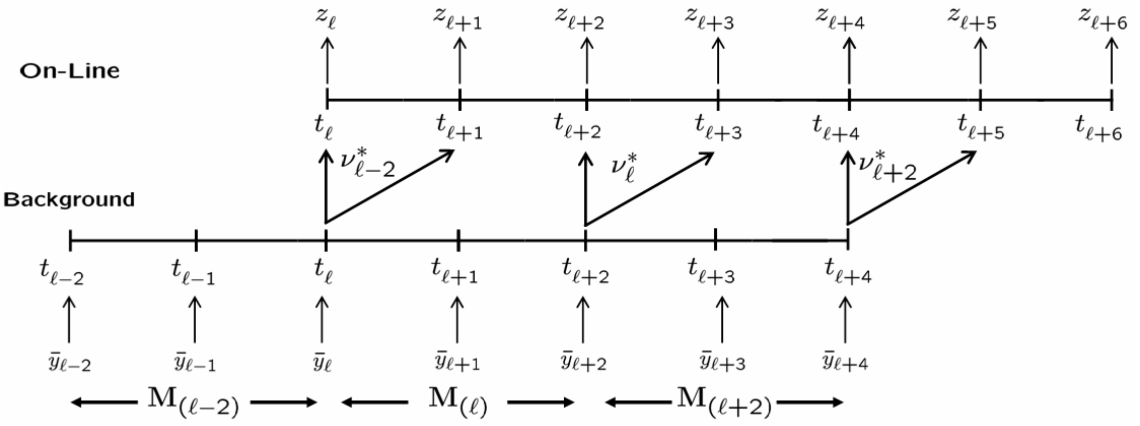

Fig. 3. Alternative schedule of on-line and background computional tasks (Case $m = 2$).

where $m = 2$. Here, notice that the solution of $\mathbf{M}(\ell)$, v^*_ℓ is used to provide instantaneous estimates at $t_{\ell+2}$ and $t_{\ell+3}$.

Finally, if the addition of $2m - 1$ extra time steps makes the background problem expensive to solve, it is also possible to accommodate m parallel background solutions of the set of problems $\mathbf{M}(\ell + j)$, $j = 0, \ldots, m$, where the solution of each $\mathbf{M}(\ell + j)$ is started at $t_{\ell+j}$ using the measurement sequence $\{\bar{y}_{\ell+j-N}, \ldots, \bar{y}_{\ell+j}\}$ to provide the state estimate $z_{\ell+j}$. Notice that by doing so, the most recent information is always used. In addition, this would allow to update the covariance matrices at each sampling time which is crucial to obtain good performance of the estimator.

Remark 2: Due to noisy operation or inaccuracy of the model, strong perturbations $(\bar{y}_{\ell+j} - \hat{y}^*_{N+j})^T$, $j = 1, \ldots, m$ can be encountered. In this case, a single iteration might not be sufficient to obtain a sufficiently accurate approximate solution. In order to improve the quality of the approximation $v^*_{\ell+m}$, it is possible to perform fast fixed-point iterations on the system,

$$\left[\begin{array}{c|c} \mathbf{K}_\ell & E_{s,m} \\ \hline E_m^T & 0 \end{array}\right] \left[\begin{array}{c} \Delta v^i \\ \hline \Delta p^i \end{array}\right] = \left[\begin{array}{c} \varphi(v^i, \mu_l) \\ \hline \mathbf{r}_m \end{array}\right] \tag{26}$$

where $\varphi(v^i, \mu_l)$ are the *nonlinear* KKT conditions of the nominal problem and i is an iteration counter with $v^0 = v^*_\ell$. In this case, the quality of the approximation can be monitored through the residual $||\varphi(v^i, \mu_l) - E^T_{s,m}\Delta p^i|| \leq \epsilon$ where ϵ is a predefined tolerance. Here, the Schur complement is only formed and factorized at $i = 0$ so that these fixed-point iterations come at a small extra expense.

Remark 3: In order to reduce the number of background backsolves for the shifted variant, an iterative algorithm can also be applied to select Δp directly to match the dummy variables with the corresponding measurements. This can be done with a number of iterative methods, including quasi-Newton (e.g., Broyden) or a preconditioned Krylov method, such as GMRES, where the stored factors of a previous Schur complement can be used as preconditioner; each iteration requires a single backsolve with $\mathbf{K}_\ell$.

4. CASE STUDY

The fast computational framework is demonstrated on a simulated MHE scenario arising on a full-scale low-density polyethylene (LDPE) process. A simplified flowsheet of a typical LDPE plant is depicted in Figure 4. For a more detailed explanation of the process, please refer to (Zavala and Biegler, 2006) and the references therein. In this process, ethylene is polymerized in a long

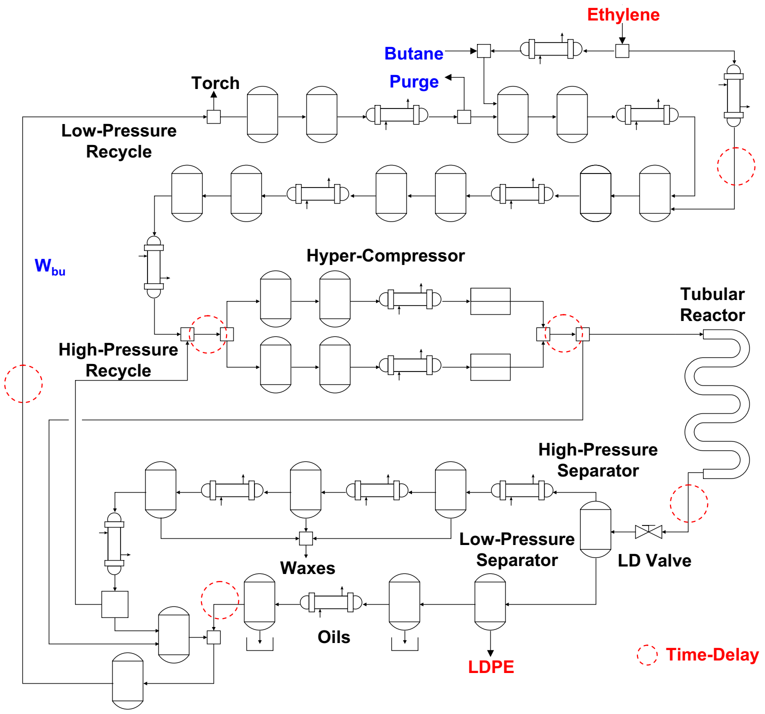

Fig. 4. Simplified flowsheet of a typical high-pressure LDPE tubular reactor process.

tubular reactor at high pressures (2000-3000 atm) and temperatures (150-300 ^{o}C) through a free-radical mechanism. Due to these extreme conditions, a large number of compression stages are required. The final product is recovered by flash separation. The process presents a difficult dynamic system with long time delays due to the recycle loops. In addition, the only available measurements are the reactor temperature profile and the gas concentration leaving the hyper-compressor. This concentration can in turn be related to the concentration in the recycle loops. In this scenario, the reactor is treated as a simple black-box conversion model. Consequently, the objective is to estimate the remaining differential states corresponding to the concentrations of ethylene, butane, methane and impurities throughout the plant units. The dynamic evolution of these states can be described by material balances around each plant unit,

$$\frac{d\left(V_k \cdot \rho_k \cdot w_{k,j}\right)}{dt} = F_k\, w_{k,j}^{in} - F_k w_{k,j}$$
$$w_{k,j}(0) = w_{k,j}^{0}$$
$$k = 1,...,N_U, \quad j = 1,...,N_C \qquad (27)$$

where N_C, number of gaseous components in the process; N_U, number of plant units; F_k, mass flow rate (kg/h); V_k, equipment volume (m^3); t, time (s); ρ_k, gas density (kg/m^3); $w_{k,j}^{in}$, inlet weight component of j-th component to the k-th unit; $w_{k,j}$, outlet weight component of j-th component from the k-th unit. The gas density at the extreme conditions is calculated through nonlinear thermodynamic relations. Most of the complexity of the dynamic model is caused by the presence of time delays. For simplicity, these delays are lumped into $N_T = 6$ overall locations along the process and are directly incorporated into the model by considering each one as a tube of length L where a plug flow is assumed. The resulting component material balances are given by the following set of partial differential equations:

$$\frac{\partial w_{i,j}}{\partial t} + \frac{1}{\tau_i}\frac{\partial w_{i,j}}{\partial z} = 0$$
$$\left(\frac{\partial w_{i,j}}{\partial z}\right)_L = 0 \qquad w_{i,j}(z,0) = w_{i,j}^{0} \qquad (28)$$

where τ_i represents the i-th time delay in the process with $i = 1,...,NT$. The PDEs are transformed to ordinary differential equations by applying a spatial finite difference scheme with 10 intervals. The resulting large-scale DAE model contains 294 differential and 64 algebraic state variables. The concentration of butane in the recycle loop y_{C_4} is used as the only output variable that is measured. The output measurements were obtained by simulation of the dynamic model using fixed control profiles over a long horizon of 5.6 hours divided into 60 sampling points. The profiles for the feed butane F_{C_4} and purge F_{Pu} flow rates are presented in Figure 5. The predicted output profile is then corrupted using Gaussian noise with a 5% standard deviation (σ). Following this reasoning, the least-squares objective function,

$$\min\ (\mathbf{z}^0 - \bar{\mathbf{z}}^0)^T \mathbf{W}_\mathbf{z}(\mathbf{z}^0 - \bar{\mathbf{z}}^0) + \sum_{k=1}^{N} \frac{1}{\sigma^2}(y_{C_4}(t_k) - \bar{y}^k_{C_4})^2 \quad (29)$$

and the model equations (27)-(28) are used for the formulation of the estimation problem. Here, $\bar{y}^k_{C_4}$ is the measured concentration of butane in the recycle loop at sampling time t_k, vector $\mathbf{z}^0 \in \Re^{294}$ contains the initial conditions for all the states with a given *a priori* estimate $\bar{\mathbf{z}}^0$ obtained from simulation. Finally, $\mathbf{W}_\mathbf{z} \in \Re^{294\times 294}$ is a diagonal matrix with entries set to $\frac{1}{0.1}$ and we impose lower and upper bounds on all the states.

Following the simultaneous collocation-based approach, a total of 15 finite elements and 3 collocation points are used for the time discretization of the dynamic model. The finite elements are placed in order to match the sampling times along the horizon ($N = 15$). The resulting NLP contains 27,121 constraints, 9330 lower bounds, 9330 upper bounds and 295 degrees of freedom corresponding to the initial conditions for the states and an extra dummy variable (m=1). Since the dynamics of the system are slow, a prediction time of 1.4 hours is used with sampling times $(t_{\ell+1} - t_\ell) = 5.6$ min.

Table 1. Average computational times associated to the background solution of the MHE problem (3.0 GHz Pentium IV processor, 1 Gb RAM).

Algorithmic Step	**CPU Time (s)**
Full Solution (6 iterations)	202.64
Single Factorization of KKT Matrix	33.77
Step Computation (single backsolve)	0.9-1.0
Rest of Steps	0.936

Computational results associated to the background solution of the NLP using IPOPT are presented in Table (1). In all our numerical experiments a monotone barrier parameter μ update with an initial value of 1x10^{-6} is used, while the rest of the algorithmic parameters were specified with their default values. It is clear that the vast majority of the total CPU time is devoted for the factorization of the KKT matrix. Note also that the NLP can be solved reliably under the allotted sampling time. In addition, the inertia of the KKT matrix was correct at the solution of these problems. Therefore, it is possible to conclude that the state of the system is observable given the limited measurement data.

Figure 6 presents the measured, estimated and true profiles of the output variable along 60 sampling times. Notice that, despite the large noise, the algorithm is able to estimate accurately the true output variable. The background computational performance of IPOPT is depicted in Figure 7. Here, the shifted approximate solutions are used to warm-start the algorithm for the solution of the nominal problem at every sampling time. By doing so, the algorithm is able to converge the nominal problem in 3-5 iterations. In this case, the noise perturbations do not induce drastic changes between neighboring problems. As a consequence, a single iteration is required to obtain instantaneous (0.9-1.0 seconds) and accurate state estimates. Although the analyzed MHE scenario has been rather simplified, the approach is general and can be applied to more complicated models and MHE formulations including, for example, process, state and input noise.

5. CONCLUSIONS

A fast and efficient moving horizon estimation algorithm is presented in this work. The framework follows a real-time iteration strategy, exploits the advantages of simultaneous collocation-based formulations and makes use of large-scale NLP algorithms and sensitivity concepts. The approach is demonstrated on a full-scale polymer process where the algorithm is able to track accurately the 294 states in the process. Moreover, on-line calculation times are reduced by over two orders of magnitude. The results obtained in this work are encouraging and will be extended to larger scale applications. In addition, future work will focus on a detailed stability analysis of the proposed MHE approach based on conditions established in previous reports (Michalska and Mayne, 1995; Rao *et al.*, 2003).

REFERENCES

Dehaan, D. and M. Guay (2006). A new real-time perspective on non-linear model predictive control. *J. Proc. Control* **16**, 615–624.

Diehl, M., H.G. Bock and J.P. Schlöder (2005). A real-time iteration scheme for nonlinear optimization in optimal feedback control. *SIAM J. Cont. Opt.* **43, 5**, 1714–1736.

Diehl, M., R. Findeisen, H.G. Bock, J.P. Schlöder and F. Allgöwer (2005*b*). Nominal stability of the real-time iteration scheme for nonlinear model predictive control. *IEEE Control Theory Appl.* **152, 3**, 296–308.

Fiacco, A. V. (1983). *Introduction to Sensitivity and Stability Analysis in Nonlinear Programming.* Academic Press. New York.

Findeisen, R. and F. Allgöwer (2004). Computational delay in nonlinear model predictive control. In: *Proc. Int. Symp. Adv. Control of Chemical Processes, ADCHEM 03.* Hong Kong.

Forsgren, A., P.E. Gill and M.H. Wright (2002). Interior methods for nonlinear optimization. *SIAM Review* **44**, 525–597.

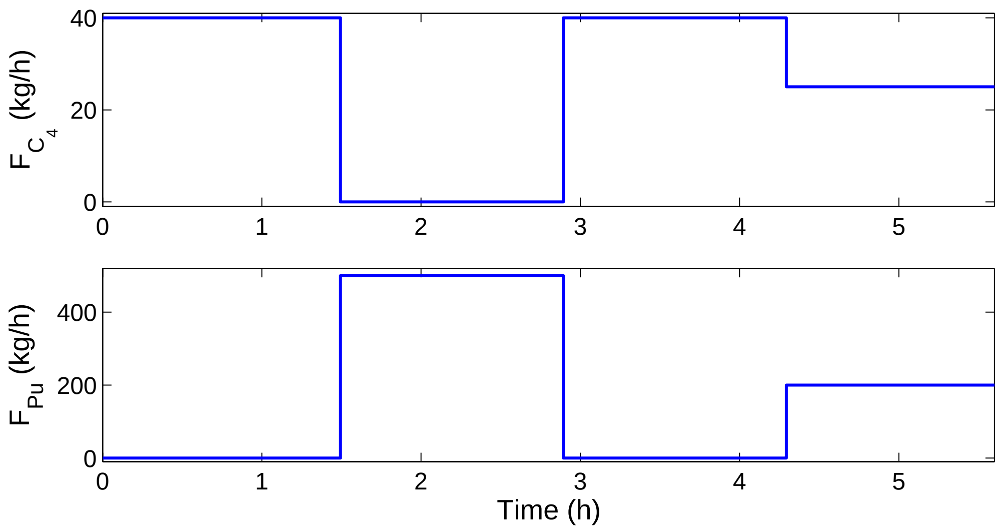

Fig. 5. Simulated step changes for the control variables.

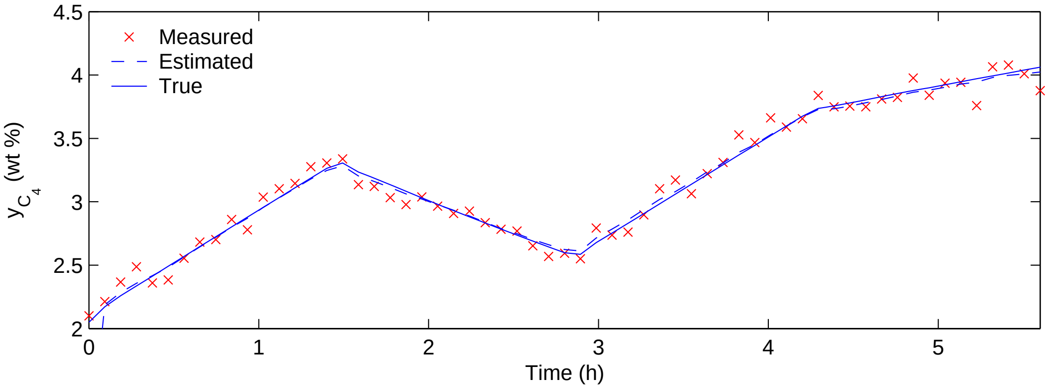

Fig. 6. Measured, estimated and true profiles of the output variable.

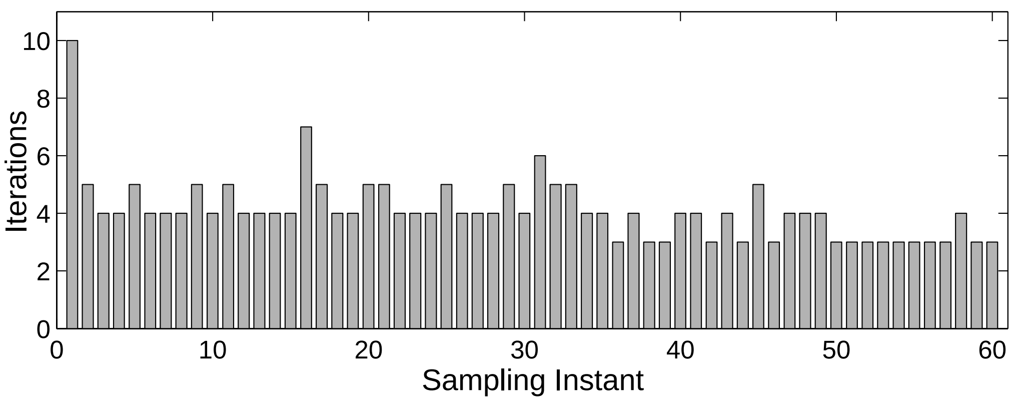

Fig. 7. Number of iterations required for the background solution of the nominal problem.

Haseltine, E. L. and J. B. Rawlings (2005). Critical evaluation of extended kalman filtering and moving horizon estimation.. *Ind. Eng. Chem. Res* **44**, 2451–2460.

Kadam, J. V. and W. Marquardt (2004). Sensitivity-based solution updates in closed-loop dynamic optimization. In: *Proc. of DYCOPS 7.* (S.L. Shah and J.F. McGregor.(Ed)), Cambridge, USA.

Kraus, T., P. Kuehl, L. Wirsching, H. S. Bock and M. Diehl (2006). A moving horizon state estimation algorithm applied to the Tennessee Eastman benchmark process. In: *IEEE Conference on Multisensor Fusion and Integration.* Heidelberg, Germany.

Michalska, H. and D. Q. Mayne (1995). Moving horizon observers and observer-based control. *IEEE Trans. Automat. Contr.)* **40**, 995-1006.

Rao, C. V. and J. B. Rawlings (2000). Nonlinear moving horizon state estimation. In: *Nonlinear Model Predictive Control.* (F. Allgöwer and A. Zheng. (Ed)), Birkhauser, Basel, Switzerland.

Rao, C. V., J. B. Rawlings and D. Q. Mayne (2003). Constrained state estimation for nonlinear discrete-time systems: Stability and moving horizon approximations. *IEEE Trans. Automat. Contr.)* **48**, 246–258.

Rawlings, J. B. and B. R. Bakshi (2006). Particle filtering and moving horizon estimation. *Comp. Chem. Eng.* **30**, 1529-1541.

Robertson, D. G., J. H. Lee and J. B. Rawlings (1996). A moving horizon based approach for least-squares state estimation. *AIChE J.* **42**, 2209-2224.

Wächter, A. and L.T. Biegler (2006). On the implementation of a primal-dual interior point filter line search algorithm for large-scale nonlinear programming. *Math. Program.* **106 (1)**, 25–57.

Zavala, V. M., C. D. Laird and L. T. Biegler (2006). Fast solvers and rigorous models: Can both be accomodated in NMPC?. *Submitted for Publication.*

Zavala, V.M. and L.T. Biegler (2006). Large-scale parameter estimation in low-density polyethylene tubular reactors. *Ind. Eng. Chem. Res.* **45**, 7867–7881.