Pathfollowing approach for a sensitivity based moving horizon estimator

Tamal Das and Johannes Jäschke

Abstract— **In moving horizon estimation (MHE), a computationally expensive nonlinear program (NLP) is solved at each sampling time to determine the current state of the system. To overcome the computational challenges, an advanced-step MHE (asMHE) framework has been proposed in the literature. asMHE consists of a computationally expensive offline part and a fast NLP sensitivity based online part. We propose a predictor-corrector pathfollowing method for the online part within asMHE. In this method, we solve a few quadratic programs sequentially in order to follow the optimal solution of the NLP for tracking a parameter change, which is the difference between a predicted measurement value and the real measurement value corresponding to the latest sample. This allows it to track the active set changes as they occur. To demonstrate the method, we performed simulations on a gas phase three component batch reaction model. We compare the solutions from the ideal-MHE and the pathfollowing based MHE. The results indicate that the pathfollowing based MHE is able to effectively trace the exact solution and the changes in active set in an efficient manner.**

I. INTRODUCTION

In moving horizon estimation (MHE), estimates for states and parameters are found by minimizing an objective function, which is a summation of weighted least squares of process noises and measurement noises in a horizon consisting of several measurement samples from immediate past, subject to model equations, output equations and constraints. When a new measurement arrives, the new sample is included in the horizon while the oldest sample is discarded. Thus, the horizon keeps moving one sampling time forward with each sample, thereby limiting the computational expense. This optimization problem is discretized into a nonlinear program (NLP) if the model equations are nonlinear.

Although MHE has evolved in the past decade as a useful tool for estimation in constrained nonlinear systems, its large computational expense and the cost associated with model development have deterred its application for online estimation in large scale systems. Even for a relatively short horizon with few states, the optimization problem can take a non-negligible amount of time to solve. A delayed estimate leads to a delayed control action, which can deteriorate the closed loop performance and potentially lead to instability [1]. Two approaches have been developed in the literature to address the issue of computational burden associated with MHE problems. Both approaches include an offline phase (preparation phase) and an online phase (estimation phase).

This work was supported by the project SUBPRO (Subsea production and processing) and the Norwegian Research Council.

T. Das and J. Jäschke are with the Department of Chemical Engineering, Norwegian University of Science and Technology, 7491 Trondheim, Norway {tamal.das,johannes.jaschke}@ntnu.no

The first approach is known as the real time iteration (RTI) scheme [2]. In RTI, the idea is to limit the computational expense of solving the NLP to one sequential quadratic programming (SQP) iteration i.e. one quadratic program (QP), which results in an efficient approximate solution. The preparation phase carries out the linearizations necessary to set up the Newton step for the KKT system, while the estimation phase embeds the final measurement as it arrives and computes the step to update the solution from the previous time step. Note that here the full NLP is never solved. In RTI, the gap between the approximate solution and the optimal solution can be narrowed by a better initial guess at each sample time by a procedure called warm-starting.

The other approach to handle computational expense of the MHE problem is to use the advanced step MHE (asMHE) [3], wherein the full NLP is solved to a given optimality tolerance in the offline part, using a predicted value for the upcoming measurement in order to obtain an approximate solution. This solution is referred as *predicted solution* and the corresponding problem *predicted NLP*. In the online part, the predicted solution is updated using the measurement (as it arrives) and the optimal sensitivity of the predicted solution to a change in the final measurement.

In comparison to other estimation techniques, MHE offers a distinct advantage in terms of the ability to handle inequality constraints. This is especially important for applications in which the variables of interest tend to be very close to the bounds. Such applications include high purity separation processes, such as distillation or reactors in which one or more species is completely consumed. In the RTI scheme, handling of inequality constraints is natural due to its adoption of QPs as a tool to solve the optimization problems. In asMHE, however, changes in the active set (set of active inequality constraints) cause a change in the dimension of the KKT matrix. The KKT matrix structure is consequently changed in the online part by applying Schur complement techniques to the factorized KKT matrix obtained from the offline part [3], which is somewhat heuristic. A rigorous approach to handle active set changes was presented within advanced step model predictive control framework in [4] and [5], where the authors presented a pathfollowing based sensitivity update.

Inspired by [4] and [5], we propose the pathfollowing advanced step MHE (pasMHE), in which we use a predictor-corrector pathfollowing method to compute a change of the predicted NLP solution in the online part. In our pathfollowing method, a series of QPs are solved to trace the NLP solution for a change in parameter, i.e. such that the latest measurement in predicted NLP is corrected for the real measurement value.

The main contribution of this paper is to present a fast MHE method that ensures that the solution can effectively track the changes in active set as they occur while adapting the final measurement variable in the online part. This is achieved by including strongly active inequality constraints as equality constraints and weakly active ones as inequality constraints in the pathfollowing QPs.

This paper is structured in the following way. In section II, we formulate the MHE problem. In section III, we present the sensitivity properties of the associated NLP. In section IV, we discuss pathfollowing approaches for NLP. We present pathfollowing advanced step MHE algorithm in section V. In section VI, the proposed algorithm is applied to a state estimation case study, for which the results can be found in section VII. We conclude the paper with a discussion and final remarks in section VIII.

II. MOVING HORIZON ESTIMATION PROBLEM FORMULATION

A. A general Moving Horizon Estimation framework

We consider NLP formulation (1) for MHE with discrete time dynamics spanning a horizon length of N finite elements, such that at time instant i, x_i and y_i represent the state and the output, respectively and u_i represents piecewise constant input between time instants i and $i+1$. $f\colon \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_x}$ represents discretized state dynamics, whereas $h\colon \mathbb{R}^{n_x} \to \mathbb{R}^{n_y}$ represents state to output mapping. The MHE objective function spans over two time segments: $t_{past} = \{i \mid 0 \le i < k-N\}$ and $t_{horizon} = \{i \mid k-N \le i \le k\}$, where k represents the current time step and the horizon consists of latest $N+1$ measurement samples. The contribution to the objective function from t_{past} is summarized in the arrival cost, whereas that from $t_{horizon}$ in the stage costs.

$$
\begin{aligned}
&\left\{\hat{x}_{k-N|k}, \hat{v}_{k-N|k}, \hat{w}_{k-N|k}, \ldots, \hat{w}_{k-1|k}, \hat{x}_{k|k}, \hat{v}_{k|k}\right\} = \\
&\arg \min_{\{X_{k-N}^{k}, V_{k-N}^{k}, W_{k-N}^{k-1}\}} \overbrace{\left[\sum_{i=k-N}^{k-1} w_i^T Q^{-1} w_i + \sum_{i=k-N}^{k} v_i^T R^{-1} v_i\right]}^{\text{Stage costs}} \\
&\overbrace{+ \left\|x_{k-N} - \hat{x}_{k-N|k-1}\right\|^2_{\Pi^{-1}_{k-N|k-1}} - \|\mathcal{Y} - \mathcal{O} x_{k-N}\|^2_{\mathcal{W}^{-1}}}^{\text{Arrival cost}} \\
&\text{s.t.} \quad x_{i+1} = f(x_i, u_i) + w_i \\
&\qquad\quad y_i = h(x_i) + v_i \\
&\qquad\quad x_{lb} \le x_i \le x_{ub}
\end{aligned}
\tag{1}
$$

We consider additive Gaussian noises denoted by $w_i \sim \mathcal{N}(0, Q)$ for process noise and $v_i \sim \mathcal{N}(0, R)$ for measurement noise. The decision variables in (1) are the sequence of states $X_{k-N}^{k} = \{x_{k-N}, \ldots, x_k\}$, process noises $W_{k-N}^{k-1} = \{w_{k-N}, \ldots, w_{k-1}\}$ and measurement noises $V_{k-N}^{k} = \{v_{k-N}, \ldots, v_k\}$ in $t_{horizon}$ denoted by $\left\{X_{k-N}^{k}, V_{k-N}^{k}, W_{k-N}^{k-1}\right\}$. The smoothed and filtered estimates of the states and the noises in $t_{horizon}$ conditioned on measurement data from $t_{horizon}$ are obtained by solving the optimization problem (1) and are denoted as $\left\{\hat{x}_{k-N|k}, \hat{v}_{k-N|k}, \hat{w}_{k-N|k}, \ldots, \hat{w}_{k-1|k}, \hat{x}_{k|k}, \hat{v}_{k|k}\right\}$, where, $\hat{x}_{k|k}$ represents the state estimate of x_k given all the measurement information until time point k. Bound constraints on the states are included in the optimization problem as inequality constraints, whereas the discretized model equations and output equations appear as equality constraints. The lower and upper bounds on the states are denoted by x_{lb} and x_{ub}, respectively. Note that in (1) the weighted 2-norm $\|z\|_Z^2$ expands as $z^T Z z$.

We consider a smoothed arrival cost approximation in our NLP formulation (1) for MHE [6], assuming a normally distributed probability density function for $x_{k-N|k-1} \sim \mathcal{N}\left(\hat{x}_{k-N|k-1}, \Pi_{k-N|k-1}\right)$. In the arrival cost, the first term represents a penalty on the deviation of the first state x_{k-N} from the smoothed estimate $\hat{x}_{k-N|k-1}$ obtained from the NLP solved at time instant $k-1$. The covariance matrix $\Pi_{k-N|k-1}$ is extracted from the reduced Hessian of the Lagrangian of the NLP for time instant $k-1$ as in [7]. The second term in the arrival cost is a correction to the first term, such that the measurement data Y_{k-N}^{k-1} are not counted twice in the arrival cost. For the exact definition of the matrices $\mathcal{Y}, \mathcal{O}$ and $\mathcal{W}$, refer to the Appendix.

B. Ideal Moving Horizon Estimation

Ideal moving horizon estimation (ideal MHE) are MHE problems that are solved at each sampling times k with the measurement sequence and arrival cost updated at each sampling time. As soon as the new measurement arrives, the NLP is assumed to be solved with no time delay.

C. Advanced step Moving Horizon Estimation framework

In reality, the MHE problem is impossible to be solved with no time delay. To reduce the time delay between receiving the new measurement value and having the new state available, the advanced step moving horizon estimation (asMHE) was proposed [3]. It involves the following two steps:

1) Offline step: In this step, the MHE is solved based on a prediction of the newest measurement, rendering an approximate solution together with the corresponding NLP sensitivity of the solution to a change in the parameter i.e. the final measurement.
2) Online step: In this step, the approximate solution from the offline step is updated (or corrected) using the real measurement and the NLP sensitivity.

III. SENSITIVITY IN PARAMETRIC NONLINEAR PROGRAMMING

The MHE problems solved in the offline step can be framed as a parametric NLP as follows:

$$
\begin{aligned}
&\min_{\boldsymbol{X}} \; F(\boldsymbol{X}) \\
&s.t. \; C(\boldsymbol{X}, \mathbf{p}) = 0 \\
&\qquad G(\boldsymbol{X}, \mathbf{p}) \le 0
\end{aligned}
\tag{2}
$$

Here, $\boldsymbol{X} \in \mathbb{R}^{n_{\boldsymbol{X}}}$ is a vector of decision variables (primal variables) containing state sequence and the process

and measurement noise sequences in the horizon window. $\mathbf{p} \in \mathbb{R}^{n_y}$ is a parameter vector corresponding to the final measurement. $F\colon \mathbb{R}^{n_X} \to \mathbb{R}$ is the scalar objective function, $C\colon \mathbb{R}^{n_X} \times \mathbb{R}^{n_y} \to \mathbb{R}^{n_C}$ denotes the equality constraints corresponding to the model equations and $G\colon \mathbb{R}^{n_X} \times \mathbb{R}^{n_y} \to \mathbb{R}^{n_G}$ denotes the inequality constraints corresponding to the constraints on states.

The Lagrangian function of (2) is defined below, where $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}$ represent the vectors of Lagrange multipliers (dual variables) for equality constraints and inequality constraints, respectively.

$$\mathcal{L}(\boldsymbol{X}, \mathbf{p}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = F(\boldsymbol{X}) + \boldsymbol{\lambda}^T C(\boldsymbol{X}, \mathbf{p}) + \boldsymbol{\mu}^T G(\boldsymbol{X}, \mathbf{p})$$

The Karush-Kuhn-Tucker (KKT) conditions for (2) are:

$$\begin{aligned} \nabla_{\boldsymbol{X}} \mathcal{L}(\boldsymbol{X}, \mathbf{p}, \boldsymbol{\lambda}, \boldsymbol{\mu}) &= 0 \\ C(\boldsymbol{X}, \mathbf{p}) &= 0 \\ G(\boldsymbol{X}, \mathbf{p}) &\leq 0 \\ \boldsymbol{\mu}^T G(\boldsymbol{X}, \mathbf{p}) &= 0 \\ \boldsymbol{\mu} &\geq 0 \end{aligned} \tag{3}$$

A point $(\boldsymbol{X}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*)$ that satisfies (3) for a given parameter vector $\mathbf{p}^*$ is called a KKT point.

Definition 1: (*Active set*) For problem (2) at a KKT point $\boldsymbol{X}$ and parameter vector $\mathbf{p}$, the *active set* $\mathbb{A}$ refers to the set $\{j \in (1, \ldots, n_G) \mid G_j(\boldsymbol{X}, \mathbf{p}) = 0\}$, and the corresponding vector of active constraints denoted by $G_{\mathbb{A}}(\boldsymbol{X}, \mathbf{p})$ is given by $\{G_j(\boldsymbol{X}, \mathbf{p}) \mid j \in \mathbb{A}\}$. The complement of set $\mathbb{A}$ is *inactive set* $\mathbb{A}^-$, where $\mathbb{A}^- = \{j \in (1, \ldots, n_G) \mid G_j(\boldsymbol{X}, \mathbf{p}) < 0\}$.

Definition 2: (Strongly active set) The *strongly active set* is a subset of set $\mathbb{A}$ given by $\mathbb{K}_+ = \{j \in \mathbb{A} \mid \mu_j > 0\}$ and the corresponding vector of active constraints in set $\mathbb{K}_+$ is denoted by $G_{\mathbb{K}_+}(\boldsymbol{X}, \mathbf{p})$. The *weakly active set* is a subset of set $\mathbb{A}$ given by $\mathbb{K}_W = \{j \in \mathbb{A} \mid \mu_j = 0\}$. Set $\mathbb{K}_0$ is given by $\mathbb{K}_0 = \mathbb{A}^- \bigcup \mathbb{K}_W$. The corresponding vector of inequality constraints in the index set $\mathbb{K}_0$ is denoted by $G_{\mathbb{K}_0}(\boldsymbol{X}, \mathbf{p})$.

Definition 3: (LICQ) For a parameter vector $\mathbf{p}$ and vector $\boldsymbol{X}$, the *linear independence constraint qualification* holds if the vectors in $\{\{\nabla_{\boldsymbol{X}} C_i(\boldsymbol{X}, \mathbf{p})\}_{i \in \{1, \ldots, n_C\}} \bigcup \{\nabla_{\boldsymbol{X}} G_i(\boldsymbol{X}, \mathbf{p})\}_{i:\, i \in \mathbb{A}}\}$ are linearly independent.

LICQ warrants that the Lagrange multipliers $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}$ corresponding to a KKT point are unique.

Definition 4: (SSOSC) The *strong second order sufficient condition* holds at a KKT point $(\boldsymbol{X}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*)$ for all non-zero directions d if $d^T \nabla^2_{\boldsymbol{XX}} \mathcal{L}(\boldsymbol{X}^*, \mathbf{p}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*)\, d > 0$ such that $\nabla_{\boldsymbol{X}} G_{\mathbb{K}_+}(\boldsymbol{X}^*, \mathbf{p}^*)^T\, d = 0$ and $\nabla_{\boldsymbol{X}} C(\boldsymbol{X}^*, \mathbf{p}^*)^T\, d = 0$.

In conjunction with LICQ, SSOSC guarantees that a KKT point is a unique local minimum.

Definition 5: (SC) *Strict complimentarity* holds if for a given parameter vector $\mathbf{p}^*$ the KKT point $(\boldsymbol{X}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*)$ satisfies $\boldsymbol{\mu}_i^* - G_i(\boldsymbol{X}^*, \mathbf{p}) > 0$ for each $i = 1, \ldots, n_G$.

Theorem 1: Let $\boldsymbol{X}^*$ satisfy KKT conditions (3) for a given parameter vector $\mathbf{p}_0$, and LICQ, SSOSC, and SC hold at $\boldsymbol{X}^*(\mathbf{p}_0)$. Further, let F, C, and G be $k+1$ times differentiable in $\boldsymbol{X}$ and k times differentiable in $\mathbf{p}$. Then

- $\boldsymbol{X}^*$ is an isolated minimizer and its corresponding Lagrange multipliers $\boldsymbol{\lambda}^*$ and $\boldsymbol{\mu}^*$ are unique.
- For $\mathbf{p}$ in the neighborhood of $\mathbf{p}_0$, the set $\mathbb{A}$ does not change.
- There exists a k times differentiable function $\sigma(\mathbf{p}) = \left[\boldsymbol{X}^*(\mathbf{p})^T \;\; \boldsymbol{\lambda}^*(\mathbf{p})^T \;\; \boldsymbol{\mu}^*(\mathbf{p})^T\right]^T$ of $\mathbf{p}$ in the neighborhood of $\mathbf{p}_0$, where $\sigma(\mathbf{p})$ corresponds to a unique local minimum for (2).

Proof: Refer to Fiacco [8]. ∎

Based on this result, we can compute the sensitivities $\nabla_{\mathbf{p}} \boldsymbol{X}$, $\nabla_{\mathbf{p}} \boldsymbol{\lambda}$, $\nabla_{\mathbf{p}} \boldsymbol{\mu}$ of the optimal solution $\boldsymbol{X}^*$, $\boldsymbol{\lambda}^*$, $\boldsymbol{\mu}^*$ to changes in the parameter vector $\mathbf{p}$ by solving the system of linear equations resulting from the application of the implicit function theorem to KKT conditions (3):

$$K \begin{bmatrix} \nabla_{\mathbf{p}} \boldsymbol{X} \\ \nabla_{\mathbf{p}} \boldsymbol{\lambda} \\ \nabla_{\mathbf{p}} \boldsymbol{\mu} \end{bmatrix} = - \begin{bmatrix} \nabla^2_{\mathbf{p}\boldsymbol{X}} \mathcal{L}(\boldsymbol{X}^*, \mathbf{p}_0, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*) \\ \nabla_{\mathbf{p}} C(\boldsymbol{X}^*, \mathbf{p}_0) \\ \nabla_{\mathbf{p}} G_{\mathbb{A}}(\boldsymbol{X}^*, \mathbf{p}_0) \end{bmatrix} \tag{4}$$

where

$$K = \begin{bmatrix} \nabla^2_{\boldsymbol{XX}} \mathcal{L}(\boldsymbol{X}^*, \mathbf{p}_0, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*) & \nabla_{\boldsymbol{X}} C(\boldsymbol{X}^*, \mathbf{p}_0) & \nabla_{\boldsymbol{X}} G_{\mathbb{A}}(\boldsymbol{X}^*, \mathbf{p}_0) \\ \nabla_{\boldsymbol{X}} C(\boldsymbol{X}^*, \mathbf{p}_0)^T & 0 & 0 \\ \nabla_{\boldsymbol{X}} G_{\mathbb{A}}(\boldsymbol{X}^*, \mathbf{p}_0)^T & 0 & 0 \end{bmatrix} \tag{5}$$

Using the optimal solution and the sensitivities computed in (4), the solution manifold in the neighborhood of $\mathbf{p}_0$ can be estimated using the following equations.

$$\begin{bmatrix} \boldsymbol{X}(\mathbf{p}_0 + \Delta\mathbf{p}) \\ \boldsymbol{\lambda}(\mathbf{p}_0 + \Delta\mathbf{p}) \\ \boldsymbol{\mu}(\mathbf{p}_0 + \Delta\mathbf{p}) \end{bmatrix} = \begin{bmatrix} \boldsymbol{X}^*(\mathbf{p}_0) \\ \boldsymbol{\lambda}^*(\mathbf{p}_0) \\ \boldsymbol{\mu}^*(\mathbf{p}_0) \end{bmatrix} + \begin{bmatrix} \nabla_{\mathbf{p}} \boldsymbol{X} \\ \nabla_{\mathbf{p}} \boldsymbol{\lambda} \\ \nabla_{\mathbf{p}} \boldsymbol{\mu} \end{bmatrix} \Delta\mathbf{p} \tag{6}$$

Within the advanced step MHE framework [3], the online step executes the above mentioned sensitivity update, wherein the upcoming measurement is treated as parameter, making the MHE problem (1) a parametric NLP (2).

Note that if SC does not hold at $\mathbf{p}_0$, above mentioned sensitivity updates by solving (4) will not follow the optimal solution manifold as a step $\Delta\mathbf{p}$ can induce changes in set $\mathbb{A}$. Hence, we propose pathfollowing as an alternative, in which we take multiple smaller steps that are fractions of $\Delta\mathbf{p}$ using quadratic programs. This ensures closer tracking of the optimal solution.

IV. SENSITIVITY USING PATHFOLLOWING

In the more general case where SC does not hold the NLP sensitivity can be obtained from a quadratic program (QP) that gives the change in solution vector $\Delta\boldsymbol{X}$ given a change in the parameter vector $\Delta\mathbf{p}$.

Theorem 2: Let LICQ and SSOSC hold at point $\boldsymbol{X}^*(\mathbf{p}_0)$ for a parameter vector $\mathbf{p}_0$. Let F, C, and G be twice continuously differentiable both in $\boldsymbol{X}$ and $\mathbf{p}$ near point $(\boldsymbol{X}^*, \mathbf{p}_0)$. Then

- The solution function $(\boldsymbol{X}^*(\mathbf{p}), \boldsymbol{\lambda}^*(\mathbf{p}), \boldsymbol{\mu}^*(\mathbf{p}))$ is Lipschitz continuous in the neighborhood of $(\boldsymbol{X}^*, \mathbf{p}_0, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*)$.
- The directional derivative of solution path $(\boldsymbol{X}^*(\mathbf{p}), \boldsymbol{\lambda}^*(\mathbf{p}), \boldsymbol{\mu}^*(\mathbf{p}))$ exists and is uniquely given by the solution of the following QP.

$$
\begin{aligned}
\min_{\Delta \boldsymbol{X}} \quad & \frac{1}{2}\Delta \boldsymbol{X}^T \nabla^2_{\boldsymbol{XX}} \mathcal{L}\left(\boldsymbol{X}^*, \mathbf{p}_0, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*\right) \Delta \boldsymbol{X} \\
& + \Delta \mathbf{p}^T \nabla^2_{\mathbf{pX}} \mathcal{L}\left(\boldsymbol{X}^*, \mathbf{p}_0, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*\right) \Delta \boldsymbol{X} \\
s.t. \quad & \nabla_{\boldsymbol{X}} C\left(\boldsymbol{X}^*, \mathbf{p}_0\right)^T \Delta \boldsymbol{X} + \nabla_{\mathbf{p}} C\left(\boldsymbol{X}^*, \mathbf{p}_0\right)^T \Delta \mathbf{p} = 0 \\
& \nabla_{\boldsymbol{X}} G_{\mathbb{K}_+}\left(\boldsymbol{X}^*, \mathbf{p}_0\right)^T \Delta \boldsymbol{X} + \nabla_{\mathbf{p}} G_{\mathbb{K}_+}\left(\boldsymbol{X}^*, \mathbf{p}_0\right)^T \Delta \mathbf{p} = 0 \\
& \nabla_{\boldsymbol{X}} G_{\mathbb{K}_W}\left(\boldsymbol{X}^*, \mathbf{p}_0\right)^T \Delta \boldsymbol{X} + \nabla_{\mathbf{p}} G_{\mathbb{K}_W}\left(\boldsymbol{X}^*, \mathbf{p}_0\right)^T \Delta \mathbf{p} \leq 0
\end{aligned} \tag{7}
$$

Proof: Refer to [9]. ■

Remark 1: The solution of QP (7) is a solution step $\Delta \boldsymbol{X}$ for a given parameter change $\Delta \mathbf{p}$ in the tangential direction at point $\boldsymbol{X}^*(\mathbf{p}_0)$. The step $\Delta \boldsymbol{X}$ is a predictor step. To this end, the QP (7) is referred as pure-predictor QP [4]. The KKT conditions for QP (7) under limit $\Delta \mathbf{p} \to 0$ leads to (4), whenever SC holds.

Remark 2: Since a QP allows inequality constraints in problem definition, QP (7) provides the flexibility of closely tracking the correct solution manifold under changes in the set $\mathbb{A}$ induced by large parameter perturbations. This property is absent in (4) and (6).

Iteratively applying a pure-predictor pathfollowing QP resembles an Euler scheme of integration. To further improve the approximation accuracy, we can include some corrector elements in the optimization problem as shown in e.g. [10]. Formulating (2) as a QP by linearizing the constraints at point $(\boldsymbol{X}^*, \mathbf{p}_0, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*)$ leads to several terms in the objective function of the QP vanishing as the parameter $\mathbf{p}$ enters linearly into the constraints of (2), leaving us with (8). Here, note that we retain the classification of inequality constraints in sets $\mathbb{K}_+$ and $\mathbb{K}_0$.

$$
\begin{aligned}
\min_{\Delta \boldsymbol{X}} \frac{1}{2}\Delta \boldsymbol{X}^T \nabla^2_{\boldsymbol{XX}} \mathcal{L}\left(\boldsymbol{X}^*, \mathbf{p}_0, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*\right) \Delta \boldsymbol{X} + \nabla_{\boldsymbol{X}} F\left(\boldsymbol{X}^*\right)^T \Delta \boldsymbol{X} \\
s.t. \quad \nabla_{\mathbf{p}} C\left(\boldsymbol{X}^*, \mathbf{p}_0\right) \Delta \mathbf{p} + \nabla_{\boldsymbol{X}} C\left(\boldsymbol{X}^*, \mathbf{p}_0\right)^T \Delta \boldsymbol{X} = 0 \\
\nabla_{\mathbf{p}} G_{\mathbb{K}_+}\left(\boldsymbol{X}^*, \mathbf{p}_0\right) \Delta \mathbf{p} + \nabla_{\boldsymbol{X}} G_{\mathbb{K}_+}\left(\boldsymbol{X}^*, \mathbf{p}_0\right)^T \Delta \boldsymbol{X} = 0 \\
G_{\mathbb{K}_0}\left(\boldsymbol{X}^*, \mathbf{p}_0\right) + \nabla_{\mathbf{p}} G_{\mathbb{K}_0}\left(\boldsymbol{X}^*, \mathbf{p}_0\right) \Delta \mathbf{p} + \nabla_{\boldsymbol{X}} G_{\mathbb{K}_0}\left(\boldsymbol{X}^*, \mathbf{p}_0\right)^T \Delta \boldsymbol{X} \leq 0
\end{aligned} \tag{8}
$$

Since the parameter $\mathbf{p}$ enters the constraints in (2) linearly, $\nabla_{\mathbf{p}} C\left(\boldsymbol{X}^*, \mathbf{p}_0\right) \Delta \mathbf{p}$ can be replaced by $C\left(\boldsymbol{X}^*, \mathbf{p}_f\right) - C\left(\boldsymbol{X}^*, \mathbf{p}_0\right)$, where $C\left(\boldsymbol{X}^*, \mathbf{p}_0\right) = 0$. Similar substitutions can be made for $\nabla_{\mathbf{p}} G_{\mathbb{K}_+}\left(\boldsymbol{X}^*, \mathbf{p}_0\right) \Delta \mathbf{p}$ and $\nabla_{\mathbf{p}} G_{\mathbb{K}_0}\left(\boldsymbol{X}^*, \mathbf{p}_0\right) \Delta \mathbf{p}$ to arrive at formulation (9). Here, note that $\nabla_{\boldsymbol{X}}$ or $\nabla^2_{\boldsymbol{XX}}$ terms are free of parameter $\mathbf{p}$, which means we can replace $\mathbf{p}_0$ with $\mathbf{p}_f$ without any change.

$$
\begin{aligned}
\min_{\Delta \boldsymbol{X}} \frac{1}{2}\Delta \boldsymbol{X}^T \nabla^2_{\boldsymbol{XX}} \mathcal{L}\left(\boldsymbol{X}^*, \mathbf{p}_f, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*\right) \Delta \boldsymbol{X} + \nabla_{\boldsymbol{X}} F\left(\boldsymbol{X}^*\right)^T \Delta \boldsymbol{X} \\
s.t. \quad C\left(\boldsymbol{X}^*, \mathbf{p}_f\right) + \nabla_{\boldsymbol{X}} C\left(\boldsymbol{X}^*, \mathbf{p}_f\right)^T \Delta \boldsymbol{X} = 0 \\
G_{\mathbb{K}_+}\left(\boldsymbol{X}^*, \mathbf{p}_f\right) + \nabla_{\boldsymbol{X}} G_{\mathbb{K}_+}\left(\boldsymbol{X}^*, \mathbf{p}_f\right)^T \Delta \boldsymbol{X} = 0 \\
G_{\mathbb{K}_0}\left(\boldsymbol{X}^*, \mathbf{p}_f\right) + \nabla_{\boldsymbol{X}} G_{\mathbb{K}_0}\left(\boldsymbol{X}^*, \mathbf{p}_f\right)^T \Delta \boldsymbol{X} \leq 0
\end{aligned} \tag{9}
$$

We call problem (9) predictor-corrector QP. However, we can give an alternative form of (9) in which $\Delta \mathbf{p}$ appears in the formulation and we enforce it to be equal to some value ϵ. This is necessary if we do not want to take a full step $\left(\mathbf{p}_f - \mathbf{p}_0\right)$, instead multiple smaller steps $\epsilon = \left(\mathbf{p}_f - \mathbf{p}_0\right)/m$, where $(m-1) \in \mathbb{N}$ and iteratively update the solution.

$$
\begin{aligned}
\min_{\Delta \boldsymbol{X}} \frac{1}{2}\Delta \boldsymbol{X}^T \nabla^2_{\boldsymbol{XX}} \mathcal{L}\left(\boldsymbol{X}^*, \mathbf{p}_0 + \Delta \mathbf{p}, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*\right) \Delta \boldsymbol{X} + \nabla_{\boldsymbol{X}} F\left(\boldsymbol{X}^*\right)^T \Delta \boldsymbol{X} \\
s.t. \quad C\left(\boldsymbol{X}^*, \mathbf{p}_0 + \Delta \mathbf{p}\right) + \nabla_{\boldsymbol{X}} C\left(\boldsymbol{X}^*, \mathbf{p}_0 + \Delta \mathbf{p}\right)^T \Delta \boldsymbol{X} = 0 \\
G_{\mathbb{K}_+}\left(\boldsymbol{X}^*, \mathbf{p}_0 + \Delta \mathbf{p}\right) + \nabla_{\boldsymbol{X}} G_{\mathbb{K}_+}\left(\boldsymbol{X}^*, \mathbf{p}_0 + \Delta \mathbf{p}\right)^T \Delta \boldsymbol{X} = 0 \\
G_{\mathbb{K}_0}\left(\boldsymbol{X}^*, \mathbf{p}_0 + \Delta \mathbf{p}\right) + \nabla_{\boldsymbol{X}} G_{\mathbb{K}_0}\left(\boldsymbol{X}^*, \mathbf{p}_0 + \Delta \mathbf{p}\right)^T \Delta \boldsymbol{X} \leq 0
\end{aligned} \tag{10}
$$

V. PATHFOLLOWING ADVANCED STEP MOVING HORIZON ESTIMATION

Based on the developments above, we propose the pathfollowing advanced step MHE (pasMHE) algorithm in Algorithm 1, which includes both the offline and the online parts. We start with the solution from the predicted NLP $\boldsymbol{X}^*(\mathbf{p}_0)$ as the initial point for the pathfollowing algorithm. To initiate the pathfollowing algorithm, we provide the number of steps m and the real final measurement y_k, which is used to compute the parameter step length ϵ, as shown in Algorithm 1, where N represents the number of finite elements. (10) is solved m times and after each solution, the pathfollowing solution is updated by adding the optimal change $\Delta \boldsymbol{X}$ to the previous solution. Likewise, the initial parameter $\mathbf{p}_0$ and the Lagrange multipliers are updated. Note that the Lagrange multipliers of (10) are approximations of the NLP Lagrange multipliers themselves and not their change.

Algorithm 1: pasMHE algorithm

input : Initiate the estimator with $\boldsymbol{X}^*(\mathbf{p}_0)$, m and y_k
output: State estimate $\hat{x}^{pf}_{k|k}$ and $\Pi^{pf}_{k|k}$ for each k

while $k \geq N$ **do**
  Offine: solve predicted NLP
  $\mathbf{p}_0 \leftarrow$ predicted measurement
  $\boldsymbol{X}^{pf}(\mathbf{p}_0) \leftarrow$ predicted NLP solution $\boldsymbol{X}^*(\mathbf{p}_0)$
  $\mathbf{p}_f \leftarrow y_k$ as measurement y_k becomes available
  $\Delta \mathbf{p} \leftarrow \left(\mathbf{p}_f - \mathbf{p}_0\right)/m$
  $\epsilon \leftarrow \Delta \mathbf{p}$
  for $i \leftarrow 1$ **to** m **do**
    Online: solve (10) using $\Delta \mathbf{p}$ to get $\Delta \boldsymbol{X}$
    $\boldsymbol{X}^{pf}(\mathbf{p}_0 + \Delta \mathbf{p}) \leftarrow \boldsymbol{X}^{pf}(\mathbf{p}_0) + \Delta \boldsymbol{X}$
    $\boldsymbol{\lambda}(\mathbf{p}_0 + \Delta \mathbf{p}) \leftarrow$ Lagrange multiplier of (10)
    $\boldsymbol{\mu}(\mathbf{p}_0 + \Delta \mathbf{p}) \leftarrow$ Lagrange multiplier of (10)
    $\mathbf{p}_0 \leftarrow \mathbf{p}_0 + \Delta \mathbf{p}$
  end
  $\hat{x}^{pf}_{k|k} \leftarrow \boldsymbol{X}^{pf}\left(\mathbf{p}_f\right)$ pathfollowing solution
  $\Pi^{pf}_{k|k} \leftarrow$ Extracted reduced Hessian at $\boldsymbol{X}^{pf}\left(\mathbf{p}_f\right)$
end

Remark 3: In the pasMHE algorithm, if the QP is infeasible, ϵ should be reduced until it becomes feasible again. After every QP solution in pathfollowing, the algorithm checks for active set changes by monitoring the Lagrange multipliers. If an active set change is observed, sets $\mathbb{K}_+$ and $\mathbb{K}_0$ are updated in the next QP.

VI. CASE STUDY

We consider a batch reaction system in gas phase with three components given by $i \in \{A, B, C\}$ undergoing the following two reactions, where k_j, $j \in \{1, 2, 3, 4\}$ denotes reaction rate constant [11].

$$A \underset{k_2}{\overset{k_1}{\rightleftharpoons}} B + C \qquad 2B \underset{k_4}{\overset{k_3}{\rightleftharpoons}} C \tag{11}$$

The stoichiometry matrix ν and reaction rate matrix r for (11) are as follows:

$$\nu = \begin{bmatrix} -1 & 1 & 1 \\ 0 & -2 & 1 \end{bmatrix}, \qquad r = \begin{bmatrix} k_1 c_A - k_2 c_B c_C \\ k_3 c_B^2 - k_4 c_C \end{bmatrix} \tag{12}$$

The state vector c is a vector of concentrations (in $Molar$) of the components c_i such that $c = \begin{bmatrix} c_A & c_B & c_C \end{bmatrix}^T$. We consider pressure in the reaction vessel as the only measurement given by $P = (\sum c_i) R_g T$, where pressure is in bar, $R_g = 0.08314 \frac{bar}{Molar.K}$ is the universal gas constant and T denotes reaction temperature in the vessel in K. The pressure sensor has a standard deviation of 0.1 bar. Hence, the measurement noise covariance is $R = 0.01$. Further, we assume that the reaction temperature is controlled at 400 K. The reaction rate constants k_j used in the simulation are given below:

k_1	k_2	k_3	k_4
0.5	0.05	0.2	0.01

The model equations are $\frac{dc}{dt} = \nu^T r$ with $c_i \geq 0$, since concentrations cannot be negative. The process noise affects the states directly such that in the plant model we have $\frac{dc}{dt} = \nu^T r + w$, where w is zero mean Gaussian random noise of the form $w \sim \mathcal{N}\left([0\ 0\ 0]^T, 10^{-4} \times [2.5\ 1\ 1]^T\right)$, hence, the process noise covariance matrix is $Q = 10^{-4} \times [2.5\ 1\ 1]^T$.

The plant simulations were performed in MATLAB Simulink using a fixed step solver ode3 with a step size of 0.1 min and initial condition $[0.5\ 0.05\ 0]^T$ to generate the data for true states and the pressure measurement. The pressure measurement data were used within the estimator to estimate the three states. For the estimator, an initial estimate of $x_{0|0} = [0.7\ 0.5\ 0.1]^T$ with an initial estimate error covariance $\Pi_{0|0} = 10^{-3} \times diag([10\ 2.5\ 1])$ was provided. The moving horizon estimator was set up considering 5 measurements in the horizon. The MHE was formulated as a NLP by discretizing the continuous time dynamics using direct collocation method. The discretization was performed using an algorithmic differentiation tool CasADi (version 3.0.0) [12]. The formulated NLPs for predicted MHE and the ideal MHE were solved using interior point solver IPOPT [13]. For pasMHE, we chose $m = 2$ and used the MATLAB QP solver quadprog.

Note that in our MHE problems, we consider that the start up phase includes increasing horizon full information estimation (FIE) problems as described in Fig. 1. Hence, the first horizon in the start up phase includes only two measurements. The correction terms in the arrival cost were only implemented in the post start up phase. We also implemented a continuous-discrete Extended Kalman Filter (EKF) [14] to see if it gives infeasible estimates.

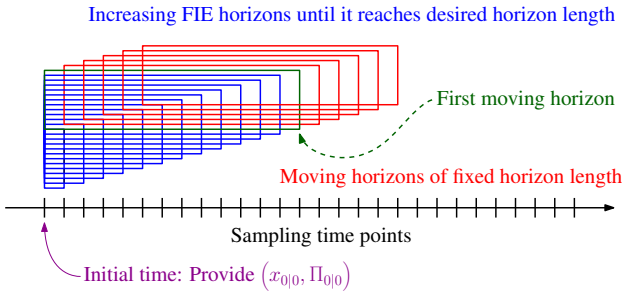

Fig. 1. MHE initialization with increasing horizon FIE problems ($x_{0|0}$ is the initial estimate and $\Pi_{0|0}$ is the initial estimate error covariance matrix)

VII. RESULTS

A. Results from Ideal MHE, predicted MHE and EKF

Firstly, we compare the state estimates obtained from the ideal MHE and EKF to the true values of the three states in the figure below.

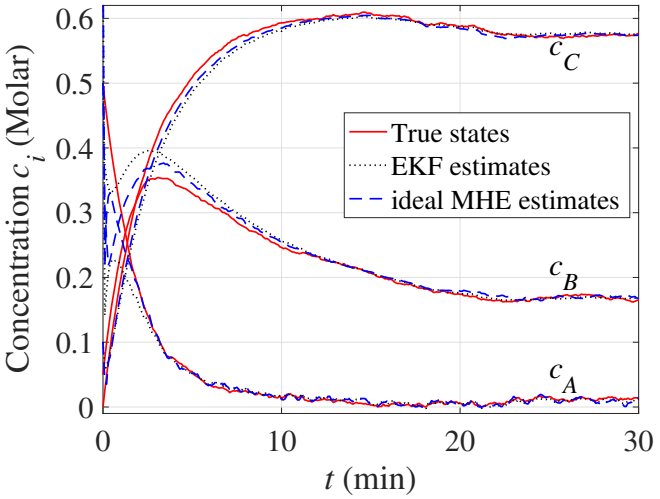

Fig. 2. True states, EKF estimates and ideal MHE estimates

We notice that the rapidly changing concentrations are tracked reasonably well by estimators. We obtained a sum of squared error (SSE) between the estimated states and the true states of 0.5795 for the ideal MHE estimates, of 0.7367 for the predicted MHE estimates and of 1.1957 for EKF estimates. While using the EKF, we obtained infeasible estimates for c_A at the following time points.

Time (min)	17.9	18.0	18.1	21.2	22.5
c_A $(Molar)$	−0.0004	−0.0008	−0.0024	−0.0024	−0.0001

B. Results from pasMHE

Here, we compare the pasMHE estimates to the ideal MHE estimates because pasMHE is proposed as an alternative to the ideal MHE problem. We computed the SSE between the ideal MHE estimates and the pasMHE estimates, which was 6.6743×10^{-7} for 300 sample points. Hence, we conclude that pasMHE is a suitable alternative to the ideal MHE because pasMHE solution closely tracks the ideal MHE solution.

Lastly, the pasMHE method was proposed to show that this method can accurately track the active set changes happening in the ideal MHE solution. We noticed that in the ideal MHE solution, the estimate c_A was active (i.e. equal to 0) at the following time points in minutes: $\{15.2, 17.9, 18.0, 18.1, 21.1, 21.2\}$. Hence, in Fig. 3 we show the error between the pasMHE solution and the ideal MHE solution obtained for c_A for the sample times between 15 and 22 minutes. We notice fairly small errors, hence, active set changes are also very closely tracked by the pasMHE solution. We further notice that at points where $c_A \geq 0$ is active, the error is non-negative, indicating that pasMHE solution is not violating the constraints.

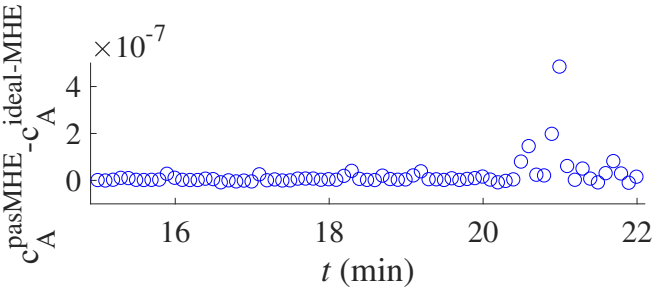

Fig. 3. Error between pasMHE solution and ideal MHE solution for c_A

VIII. CONCLUSION AND DISCUSSION

In this paper, we presented a predictor-corrector based pathfollowing approach within advanced step moving horizon estimation framework. The key motivation is to track the solution of the original NLP as closely as possible by making the strongly active constraints as equality constraints and weakly active constraints as inequality constraints. This procedure ensures that the changes in active set within the online part of the algorithm are tracked accurately, which is important for applications, such as distillation and reaction systems, in which estimated variables tend to be very close to their bounds.

pasMHE and RTI based MHE (RTI-MHE) [2] are similar methods because both solve QPs, however, the real difference between them is two fold. Firstly, pasMHE solves full NLP, whereas RTI does not. Secondly, in pasMHE, the strongly active constraints and the weakly active constraints are treated differently, ensuring that the QPs solved are always strongly convex even if the Hessian of the Lagrangian is not positive definite. This property, which warrants fast convergence at a minor overhead of identifying active constraints outside QP, is missing in reported RTI schemes. The effect of not enforcing strongly active constraints has been demonstrated in Section 3.3 in [5]. In NLP for MHE problems, nonlinear constraints arising from the model can potentially make the Hessian of the Lagrangian indefinite.

Computationally, pasMHE is more costly than RTI-MHE because the NLP is solved at each sample time. However, as long as the NLP can be solved in between two samples, the computational delay will be the time needed to solve the pathfollowing problems, which typically are a few QP iterations.

APPENDIX

Matrices $\mathcal{Y}, \mathcal{O}$ and $\mathcal{W}$ used in (1) are below [7], where H denotes $\frac{\partial h}{\partial x}$, J denotes the Jacobian of the state derivatives with respect to states and ω and ν represent respectively the estimates of process and measurement noises obtained from the previous horizon MHE problem.

$$\mathcal{Y} = \mathcal{O} x_{k-N|k-1} + \mathcal{M}\omega + \nu$$

$$\mathcal{W} = \mathcal{O}\Pi_{k-N|k-1}\mathcal{O}^T + \mathcal{M}\mathcal{Q}\mathcal{M}^T + \mathcal{R}$$

$$\mathcal{O} = \begin{bmatrix} H_{k-N} \\ H_{k-N+1}J_{k-N} \\ \vdots \\ H_{k-1}J_{k-2}J_{k-3}\cdots J_{k-N} \end{bmatrix};\ \omega = \begin{bmatrix} w_{k-N} \\ \vdots \\ w_{k-1} \end{bmatrix};\ \nu = \begin{bmatrix} v_{k-N} \\ \vdots \\ v_{k-1} \end{bmatrix}$$

$$\mathcal{M} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ H_{k-N+1} & 0 & 0 & 0 \\ H_{k-N+2}J_{k-N+1} & H_{k-N+2} & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots \\ H_{k-1}J_{k-2}J_{k-3}\cdots J_{k-N+1} & H_{k-1}J_{k-2}J_{k-3}\cdots J_{k-N+2} & \cdots & H_{k-1} \end{bmatrix}$$

$$\mathcal{Q} = diag(\underbrace{Q, \ldots, Q}_{N-1 \text{ times}});\ \mathcal{R} = diag(\underbrace{R, \ldots, R}_{N \text{ times}})$$

REFERENCES

[1] R. Findeisen and F. Allgöwer, Computational delay in nonlinear model predictive control, *Proc. Int. Symp. Adv. Contr. of Chem. Proc*, 2004, pp 427-432.

[2] P. Kühl, M. Diehl, T. Kraus, J.P. Schlöder and H.G. Bock, A real-time algorithm for moving horizon state and parameter estimation, *Computers & chemical engineering*, vol. 35(1), 2011, pp 71-83.

[3] V.M. Zavala, C.D. Laird and L.T. Biegler, A fast moving horizon estimation algorithm based on nonlinear programming sensitivity, *Journal of Process Control*, vol. 18(9), 2008, pp 876-884.

[4] J. Jäschke, X. Yang and L.T. Biegler, Fast economic model predictive control based on NLP-sensitivities, *Journal of Process Control*, vol. 24(8), 2014, pp. 1260-1272.

[5] E. Suwartadi, V. Kungurtsev and J. Jäschke, Sensitivity-Based Economic NMPC with a Path-Following Approach, *Processes*, vol. 5(1), 8, 2017.

[6] R. López-Negrete, Nonlinear Programming Sensitivity Based Methods for Constrained State Estimation, *PhD Dissertation*, 2011.

[7] R. López-Negrete and L.T. Biegler, A moving horizon estimator for processes with multi-rate measurements: A nonlinear programming sensitivity approach, *Journal of Process Control*, vol. 22(4), 2012, pp 677-688.

[8] A.V. Fiacco, *Introduction to sensitivity and stability analysis in nonlinear programming*, Academic Press, NY; 1984.

[9] K. Jittorntrum, Solution point differentiability without strict complementarity in nonlinear programming, *Sensitivity, Stability and Parametric Analysis*, 1984, pp. 127-138.

[10] V. Kungurtsev and M. Diehl, Sequential quadratic programming methods for parametric nonlinear optimization, *Computational Optimization and Applications*, vol. 59(3), 2014, pp. 475-509.

[11] E.L. Haseltine and J.B. Rawlings, Critical evaluation of extended Kalman filtering and moving-horizon estimation, *Industrial & engineering chemistry research*, vol. 44(8), 2005, pp. 2451-2460.

[12] J. Andersson, J. Åkesson and M. Diehl, CasADi: A symbolic package for automatic differentiation and optimal control, *Recent advances in algorithmic differentiation*, Springer, 2012, pp. 297-307.

[13] A. Wächter L.T. Biegler, On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming, *Mathematical programming*, vol. 106(1), 2006, pp. 25-57.

[14] D. Simon, *Optimal state estimation: Kalman, H infinity, and nonlinear approaches*, John Wiley & Sons, 2006, pp. 405