2011 American Control Conference
on O'Farrell Street, San Francisco, CA, USA
June 29 - July 01, 2011

Structure Exploiting Derivative Computation for Moving Horizon Estimation

Peter Philipp

Abstract— **This paper deals with the solution of the optimization problem resulting from the moving horizon estimation technique. A method is presented for the most crucial and computational costly part, namely the first and second-order derivative calculation. It fully utilizes the properties of the sequential single shooting based problem formulation by exploiting the structure in the derivatives as well as in the associated first-order sensitivities. This results in a highly efficient method suitable for parallelization. Furthermore, the generalized problem formulation offers the possibility to consider computational delays and allows the application of the method to both continuous-time and discrete-time systems. Simulation results confirm the effectiveness of the proposed method.**

I. INTRODUCTION

Knowledge of all state variables of dynamical systems is often needed in theory as well as in practice, while only some of them can be directly measured. Using the available measurements to compute all state variables is referred to as state estimation. For linear systems, well-understood techniques such as the Kalman filter exist [3]. However, for nonlinear systems state estimation is still an active area of research. Most methods are extensions of linear state estimators, such as the extended Kalman filter. An efficient nonlinear technique is the moving horizon estimator (MHE) which seeks to minimize an estimation cost function defined on a sliding window involving a finite number of past samples. Several MH observers were developed for continuous-time measurements [14], [1] as well as for discrete-time measurements [8], [9]. The main advantages of MHE are their capability of incorporating constraints on the estimated states, their robustness and their stability properties [11]. Moreover, MHE has been proven to be superior over traditional estimation approaches such as the aforementioned extended Kalman filter [4]. Since MHE is an optimization-based state estimation method, it strongly depends on the underlying numerical optimization schemes. The efficient solution of these nonlinear programs (NLPs) has been recently addressed in [13], [5], [2].

Two basic approaches can be distinguished regarding the treatment of the system equation constraint in the NLPs: the sequential approach and the simultaneous approach. In the former case, in each optimization iteration the two steps, system simulation and optimization are performed sequentially, which leads to a strongly reduced variable space at the expense of costly derivatives of the cost function. In the latter case, optimization and simulation are performed simultaneously. The main two algorithmic concepts for actually solving these NLPs are the sequential quadratic programming (SQP) method and the interior-point (IP) method. It is well known, that in all Newton-type optimization routines the two crucial and computationally most expensive steps are the derivative computation and the solution of the quadratic subproblems [2].

In this article, a structure exploiting method is presented for computing the first and second-order derivatives for the sequential approach. The contribution of this article is threefold: first, it is shown how different discretization schemes and possible computational delays can be incorporated into one general NLP; second, it is shown how the first derivative of the Lagrange function of this NLP can be computed based on first-order sensitivities leading to a new algorithm suitable for parallelization. It should be pointed out here that the number of ODEs to be solved is independent of the number of optimization variables on the sliding time window. Third, it is shown how these sensitivities can be used to compute the second derivative of the Lagrange function.

The remainder of this paper is organized as follows: In section II, the state estimation problem is formulated as an NLP. Before applying the SQP as well as the IP method to this NLP in section III, the sequential and the simultaneous approach are contrasted. The key goal of this paper, namely the computation of the first and second-order derivatives is outlined in Section IV. Subsequently, in Section V a numerical example is given to demonstrate the efficiency of the proposed method before the paper is concluded in Section VI with a summary.

II. PROBLEM FORMULATION

Consider the nonlinear dynamic system described by the continuous-time equations

$$\dot{x}(t) = f(x(t), u(t)) + w(t) \tag{1a}$$

$$y(t) = h(x(t)) + v(t), \tag{1b}$$

where $x \in \mathbb{R}^n$ is the state vector (the initial state x_0 is unknown) and $u \in \mathbb{R}^m$ is the control vector. The vector $w \in \mathbb{R}^n$ is an additive disturbance affecting the system dynamics. The state vector is observed through the measurement equation (1b), where $y \in \mathbb{R}^p$ is the observation vector and $v \in \mathbb{R}^p$ is a measurement noise vector. The functions $f : \mathbb{R}^n \times \mathbb{R}^m \mapsto \mathbb{R}^n$ and $h : \mathbb{R}^n \mapsto \mathbb{R}^p$ are assumed twice differentiable.

The sampled-data representation, obtained by measuring the observation vector at times t_k for $k = 0, 1, \ldots$, is derived

The author is with the Institute of Automatic Control, Technische Universität München, Boltzmann Str. 15, D-85748 Garching, Germany peter.philipp@tum.de

by integrating (1a) over the interval $[t_k, t_{k+1}]$

$$x_{k+1} = x_k + \int_{t_k}^{t_{k+1}} f(x(t), u(t))\, dt + w_k \tag{2a}$$

$$y_k = h(x_k) + v_k. \tag{2b}$$

Note that this general continuous-time formulation includes several special cases arising from discretization, like e. g. the zero-order hold (ZoH) discretization or step-invariant transformation, where $u(t)$ is assumed to be a constant vector u_k over the time interval $[t_k, t_{k+1}]$ (see the "ZoH discretization case" in Fig. 1). However, this assumption implies instantaneous feedback at each sampling time, i. e. no computational delays τ_k between the arrival of y_k and the computation of x_k and u_k, e. g. due to the solution of an optimization problem. This scenario leads to a delayed applied u_k and can be seen in Fig. 1 as the "computational delay case". Moreover, the formulation (2a) also allows other discretization schemes, with or without incorporating possible time delays or even a completely continuous control vector (see the "continuous case" in Fig. 1).

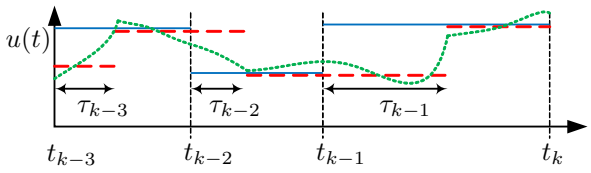

Fig. 1. Cases for $u(t)$: **Solid blue line:** ZoH discretization case. **Dashed red line:** computational delay case. **Dotted green line:** continuous case.

The objective in the MHE framework is to derive for any $k = N, N+1, \ldots$ estimates[1] of the stacked state vector $\hat{\boldsymbol{x}}_k = [\hat{x}_{k-N}^T, \ldots, \hat{x}_k^T]^T \in \mathbb{R}^{(N+1)n}$ and the stacked state disturbance vector $\hat{\boldsymbol{w}}_k = [\hat{w}_{k-N}^T, \ldots, \hat{w}_{k-1}^T]^T \in \mathbb{R}^{Nn}$ on the basis of the information vector

$$\begin{aligned} \boldsymbol{\eta}_k &= [\ \underbrace{[y_{k-N}^T, \ldots, y_k^T]}, \quad \underbrace{u(t)^T}]^T, \quad t \in [t_{k-N}, t_k] \\ &= [\ \qquad \boldsymbol{y}_k^T \qquad , \quad \boldsymbol{u}_k^T \ ]^T \in \mathbb{R}^{(N+1)p+m}, \end{aligned} \tag{3}$$

where $N+1$ measurements and the input vector are collected within a "moving horizon" interval $[k-N, k]$. More specifically, the estimator addresses for any $k = N, N+1, \ldots$ the minimization of the following cost function[2]

$$J_k(\boldsymbol{\eta}_k) = \|\hat{x}_{k-N} - \bar{x}_{k-N}\|_{P_k}^2 + \sum_{i=k-N}^{k} \|\hat{v}_i\|_{R_k}^2 + \sum_{i=k-N}^{k-1} \|\hat{w}_i\|_{Q_k}^2, \tag{4}$$

where the matrices P_k, R_k and Q_k are assumed to be positive definite. The first term, known as the arrival cost, penalizes the distance from the estimate $\hat{x}_{k-N}$ of the state at the beginning of the moving horizon to some prediction $\bar{x}_{k-N}$. This prediction $\bar{x}_{k-N}$ of the initial state is obtained by incorporating past information $y_0, \ldots, y_{k-N-1}$ and $u(t)$ in the interval $t \in [t_0, t_{k-N}[$, which is not explicitly accounted for in the objective function. The second term is a penalization of the measurement noise whereas the third term is a penalization of the state disturbance. In addition to the weighted Euclidian norm, sometimes other penalty functions like e. g. the l_1 penalty or other formulations of the arrival costs are used. For this reason, a generalized formulation of the objective function J_k is considered in this paper

$$J_k(\boldsymbol{\eta}_k) = \Gamma(\hat{x}_{k-N}) + \sum_{i=k-N}^{k} \Upsilon_i(\hat{v}_i) + \sum_{i=k-N}^{k-1} \Psi_i(\hat{w}_i). \tag{5}$$

Consequently, the NLP problem of the MHE can be derived by eliminating $\hat{v}_i$ in (5) using the measurement equation (2b)

$$\min_{\hat{\boldsymbol{x}}_k, \hat{\boldsymbol{w}}_k} \Gamma(\hat{x}_{k-N}) + \sum_{i=k-N}^{k} \Upsilon_i(y_i, \hat{x}_i) + \sum_{i=k-N}^{k-1} \Psi_i(\hat{w}_i) \tag{6a}$$

$$\text{s. t. } \hat{x}_{i+1} - \hat{x}_i - \int_{t_i}^{t_{i+1}} f(\hat{x}(t), u(t))\, dt - \hat{w}_i = 0, \quad \forall i \in \mathcal{K} \tag{6b}$$

$$c_i(\hat{x}_i, \hat{w}_i) \geq 0, \quad \forall i \in \mathcal{K} \tag{6c}$$

$$c_k(\hat{x}_k) \geq 0, \tag{6d}$$

where $\mathcal{K} = \{k-N, \ldots, k-1\}$ is the set of indices corresponding to the current moving horizon and c_i are the inequality constraints[3]. The functions $c_i : \mathbb{R}^n \times \mathbb{R}^n \mapsto \mathbb{R}^n, \forall i \in \mathcal{K}$ and $c_k : \mathbb{R}^n \mapsto \mathbb{R}^n$ are assumed to be twice differentiable. From the solution $\hat{\boldsymbol{x}}_k, \hat{\boldsymbol{w}}_k$ of this problem, the current state $\hat{x}_k$ can be extracted, or in the presence of non-negligible computational delays, the propagation $\hat{x}(t_k + \tau_k)$ calculated.

III. NEWTON TYPE OPTIMIZATION

A. Sequential vs. simultaneous approach

When solving the NLP stated above in (6), it has to be decided first, how to treat the equality constraint (6b). This constraint uniquely determines the vector $\hat{\boldsymbol{x}}_k$ if the vectors $\hat{x}_{k-N}$, $\hat{\boldsymbol{w}}_k$ and $\boldsymbol{u}_k$ are fixed. Thus, an implicit function $\tilde{\boldsymbol{x}}_k(\hat{x}_{k-N}, \hat{\boldsymbol{w}}_k, \boldsymbol{u}_k)$ can be defined that satisfies (6b) for all $\hat{x}_{k-N}$, $\hat{\boldsymbol{w}}_k$ and $\boldsymbol{u}_k$ by a system simulation. Consequently, the constraint (6b) can be replaced in the optimization problem by substituting the function $\tilde{\boldsymbol{x}}_k(\hat{x}_{k-N}, \hat{\boldsymbol{w}}_k, \boldsymbol{u}_k)$ with $\hat{\boldsymbol{x}}_k$. Hence, the NLP can be reduced to

$$\min_{\substack{\hat{x}_{k-N} \\ \hat{\boldsymbol{w}}_k}} \Gamma(\hat{x}_{k-N}) + \sum_{i=k-N}^{k} \Upsilon_i(y_i, \tilde{x}_i(\hat{x}_{k-N}, \hat{\boldsymbol{w}}_k, \boldsymbol{u}_k)) + \sum_{i=k-N}^{k-1} \Psi_i(\hat{w}_i) \tag{7a}$$

$$\text{s. t.} \quad c_i(\tilde{x}_i(\hat{x}_{k-N}, \hat{\boldsymbol{w}}_k, \boldsymbol{u}_k), \hat{w}_i) \geq 0, \quad \forall i \in \mathcal{K} \tag{7b}$$

$$c_k(\tilde{x}_k(\hat{x}_{k-N}, \hat{\boldsymbol{w}}_k, \boldsymbol{u}_k)) \geq 0. \tag{7c}$$

This leads to the so called "sequential" approach, where in each optimization iteration the two steps, system simulation and optimization are performed sequentially, one after the other. The advantage of this method is the strongly reduced variable space compared to the original problem. The computation of the derivatives is more costly, but there is a certain structure in the problem which will be fully exploited in Section IV.

[1] Estimated values are denoted with a $\hat{}$ to distinguish them from the true values

[2] Given a symmetric positive definite matrix M and a vector z, $\|z\|_M$ denotes the weighted norm of z, $\|z\|_M := (z^T M z)^{1/2}$

[3] $\geq$ denotes componentwise inequality

In contrast, the so called "simultaneous" approach addresses the full NLP (6) directly, i. e. optimization and simulation are performed simultaneously. The full NLP (6) has larger but structured linear subproblems than the sequential approach. Thus, band structure exploiting methods or "condensing" must be used to exploit this fact [2].

B. Sequential quadratic programming and interior-point method

Now the main two algorithmic concepts are applied to the NLP (7). To this end, an overall optimization variable

$$\boldsymbol{p} = [\hat{x}_{k-N}^T, \hat{w}_{k-N}^T, \ldots, \hat{w}_{k-1}^T]^T \in \mathbb{R}^{(N+1)n} \tag{8}$$

is defined. Then the Lagrange function associated to the NLP (7) can be expressed as

$$\mathcal{L}(\boldsymbol{p}, \boldsymbol{\lambda}) = \Gamma(\hat{x}_{k-N}) + \sum_{i=k-N}^{k} \Upsilon_i(y_i, \tilde{x}_i(\boldsymbol{p}, \boldsymbol{u}_k)) + \sum_{i=k-N}^{k-1} \Psi_i(\hat{w}_i) - \sum_{i=k-N}^{k-1} \lambda_i^T c_i(\tilde{x}_i(\boldsymbol{p}, \boldsymbol{u}_k), \hat{w}_i) - \lambda_k^T c_k(\tilde{x}_k(\boldsymbol{p}, \boldsymbol{u}_k)), \tag{9}$$

where $\boldsymbol{\lambda} = [\lambda_{k-N}^T, \ldots, \lambda_k^T]^T \in \mathbb{R}^{(N+1)n}$ is a vector of Lagrange multipliers. To simplify the analysis, the arguments of all functions are in the following suppressed from the notation when the meaning is otherwise clear.

Any solution of the NLP (7) should satisfy the famous Karush-Kuhn-Tucker (KKT) conditions

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{p}} = 0 \tag{10a}$$

$$0 \leq c_i \quad \perp \quad \lambda_i \geq 0, \quad i = k-N, \ldots, k, \tag{10b}$$

where the symbol $\perp$ between the two vector valid inequalities states that also the complementary condition

$$c_{i,j}\, \lambda_{i,j} = 0, \quad i = k-N, \ldots, k, \quad j = 1, \ldots, n \tag{10c}$$

shall hold and where $c_{i,j}$ and $\lambda_{i,j}$ denotes the j-th element of c_i and λ_i respectively.

A first variant to iteratively solve this KKT system is to linearize all nonlinear functions appearing in (10). It turns out that the resulting linear complementary system can be interpreted as the KKT conditions of the following quadratic program (QP)

$$\min_{\Delta \boldsymbol{p}} \frac{\partial \mathcal{L}}{\partial \boldsymbol{p}}^T \Delta \boldsymbol{p} + \frac{1}{2} \Delta \boldsymbol{p}^T \frac{\partial^2 \mathcal{L}}{\partial \boldsymbol{p}^2} \Delta \boldsymbol{p} \tag{11a}$$

$$\text{s.t. } c_i + \frac{\partial c_i}{\partial \boldsymbol{p}} \Delta \boldsymbol{p} \geq 0, \quad i = k-N, \ldots, k. \tag{11b}$$

This general approach to address the nonlinear optimization problem is called SQP.

An alternative way to solve the KKT system iteratively is the IP method which introduces slack vectors $s_i \in \mathbb{R}^n$ and transforms the KKT conditions into the following form (primal-dual formulation)

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{p}} = 0 \tag{12a}$$

$$c_i - s_i = 0, \quad i = k-N, \ldots, k \tag{12b}$$

$$s_{i,j}\, \lambda_{i,j} = \mu, \quad i = k-N, \ldots, k, \quad j = 1, \ldots, n, \tag{12c}$$

where $s_{i,j}$ denotes the j-th element of s_i and μ is a positive parameter. The inequalities $s_{i,j} \geq 0$ and $\lambda_{i,j} \geq 0$ are required in all iterations. Note that if $\mu = 0$, the transformed KKT conditions (12) coincide with the KKT conditions (10).

The system (12) is then solved with Newton's method which involves the same first and second-order derivatives of $\mathcal{L}$ as the SQP approach. The obtained solution is not a solution to the original problem, but to the problem

$$\min_{\boldsymbol{p}, \boldsymbol{s}} J_k - \mu \sum_{i=k-N}^{k} \sum_{j=1}^{n} \ln(s_{i,j}) \tag{13a}$$

$$\text{s.t. } c_i - s_i = 0, \quad i = k-N, \ldots, k. \tag{13b}$$

It is important to note that the performance of the SQP as well as the IP method mainly depends on the accurate and fast computation of the derivatives $\partial \mathcal{L}/\partial \boldsymbol{p}$ and $\partial^2 \mathcal{L}/\partial \boldsymbol{p}^2$.

IV. DERIVATIVE COMPUTATION

A. First-order derivative of the Lagrange function $\mathcal{L}$

A common method to compute the gradient of $\mathcal{L}$ is by finite differences. For instance, the elements of $\partial \mathcal{L}/\partial \boldsymbol{p}$ can be approximated by the central-difference formula

$$\frac{\partial \mathcal{L}}{\partial p_i} \approx \frac{\mathcal{L}(\boldsymbol{p} + \epsilon\, e_i) - \mathcal{L}(\boldsymbol{p} - \epsilon\, e_i)}{2\epsilon}, \quad i = 1, \ldots, (N+1)n \tag{14}$$

where ϵ is a small positive scalar and e_i is the i-th unit vector. However, it is strongly advised not to use this method here due to the high numerical complexity and the poor achieved accuracy. The evaluation of (14) is as costly as solving $2(N+1)n^2$ ODEs over the complete moving horizon interval $[t_{k-N}, t_k]$.

The exact derivatives of (9) can be calculated by applying the chain rule whereby the dependence of $\tilde{x}_i$ on $\boldsymbol{p}$ has to be considered. This leads to the exact gradient

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{p}} = \frac{\partial \Gamma}{\partial \boldsymbol{p}} + \sum_{i=k-N}^{k} \frac{\partial \tilde{x}_i}{\partial \boldsymbol{p}}^T \frac{\partial \Upsilon_i}{\partial \tilde{x}_i} + \sum_{i=k-N}^{k-1} \frac{\partial \Psi_i}{\partial \boldsymbol{p}} - \sum_{i=k-N}^{k} \frac{\partial \tilde{x}_i}{\partial \boldsymbol{p}}^T \frac{\partial c_i}{\partial \tilde{x}_i}^T \lambda_i \tag{15}$$

where

$$\frac{\partial \tilde{x}_i}{\partial \boldsymbol{p}} = \left[\frac{\partial \tilde{x}_i}{\partial \hat{x}_{k-N}}, \quad \frac{\partial \tilde{x}_i}{\partial \hat{w}_{k-N}}, \quad \ldots, \quad \frac{\partial \tilde{x}_i}{\partial \hat{w}_{k-1}} \right] \tag{16}$$

are for $i = k-N, \ldots, k$ the first-order sensitivities of $\tilde{x}_i$ with respect to $\boldsymbol{p}$ evaluated at the corresponding sampling times on the moving horizon interval.

In the following, a new approach for computing these sensitivities is proposed. To this end, the notations below are introduced.

Notation 1: The abbreviation $X(t)$ and ${}^j Z(t)$ denotes the first-order sensitivity $\partial \tilde{x}(t)/\partial \hat{x}_{k-N}$ and $\partial \tilde{x}(t)/\partial \hat{w}_j$, respectively, where $j = k-N, \ldots k-1$.

Lemma 1: The first-order sensitivity $X(t) \in \mathbb{R}^{n \times n}$ satisfies the following first-order sensitivity matrix differential equation on the moving horizon interval $t \in [t_{k-N}, t_k]$

$$\dot{X}(t) = \frac{\partial f}{\partial \hat{x}}(t)\, X(t) \tag{17a}$$

with the initial value

$$X(t_{k-N}) = I. \tag{17b}$$

The unique solution is

$$X(t) = \Phi(t, t_{k-N})\, X(t_{k-N}), \tag{17c}$$

where $\Phi(\cdot) \in \mathbb{R}^{n\times n}$ denotes the time-varying transition matrix.

Proof: The proof is sketched as follows: Equation (17a) results from substituting $x(t)$ with $\hat{x}(t)$ in (1a) and a subsequent differentiation with respect to $\hat{x}_{k-N}$. Equation (17b) results from differentiating the initial value on the moving horizon. The solution approach for linear time-varying systems (see [7]) applied to (17a) combined with (17b) leads to (17c). ■

Note that the transition matrix in (17c) is identical to the transition matrix of a linear time-varying system with the dynamic matrix $\partial f/\partial \hat{x}$. For later use, some properties of this transition matrix $\Phi(\cdot)$ are stated from [7] in Lemma 2.

Lemma 2: The time-varying transition matrix satisfies:

(1) transition property $\quad \Phi(t_2, t_0) = \Phi(t_2, t_1)\,\Phi(t_1, t_0)$
(2) inversion property $\quad \Phi(t_1, t_0)^{-1} = \Phi(t_0, t_1)$
(3) identity property $\quad \Phi(t_0, t_0) = I$

For the first-order sensitivities ${}^{j}Z(t)$ one has to infer from $\hat{w}_j$ to $\hat{w}(t)$ which cannot be done exactly in the continuous-time case without any knowledge of the characteristics of $\hat{w}(t)$. For the discrete-time case, the underlying discretization scheme yields the desired relation, e. g. in the ZoH case: $\hat{w}(t) = \hat{w}_j/(t_{j+1} - t_j), t \in]t_j, t_{j+1}]$. More generally, such a relation can be formulated for both cases by the function $\theta_j(t, \hat{w}_j) : \mathbb{T}_j \times \mathbb{R}^n \mapsto \mathbb{R}^n$ where $\mathbb{T}_j = \{t \in \mathbb{R} | t_j < t \leq t_{j+1}\}$. The derivative of $\theta_j(t, \hat{w}_j)$ with respect to $\hat{w}_j$ is denoted by $\theta_j'(t, \hat{w}_j)$.

Lemma 3: The first-order sensitivity ${}^{j}Z(t) \in \mathbb{R}^{n\times n}$ satisfies the following first-order sensitivity matrix differential equation on the moving horizon interval $t \in [t_{k-N}, t_k]$

$${}^{j}\dot{Z}(t) = \frac{\partial f}{\partial \hat{x}}(t)\ {}^{j}Z(t) + \begin{cases} 0, & t \leq t_j \\ \theta_j'(t, \hat{w}_j), & t_j < t \leq t_{j+1} \\ 0, & t > t_{j+1} \end{cases} \tag{18a}$$

with the initial value

$${}^{j}Z(t_{k-N}) = 0. \tag{18b}$$

The unique solution is

$${}^{j}Z(t) = \begin{cases} 0, & t \leq t_j \\ \int_{t_j}^{t} \Phi(t, \tau)\, \theta_j'(\tau, \hat{w}_j)\, d\tau, & t_j < t \leq t_{j+1} \\ \int_{t_j}^{t_{j+1}} \Phi(t, \tau)\, \theta_j'(\tau, \hat{w}_j)\, d\tau, & t > t_{j+1} \end{cases} \tag{18c}$$

where $\Phi(\cdot) \in \mathbb{R}^{n\times n}$ denotes the time-varying transition matrix.

Proof: The proof is straightforward and uses the same techniques as the proof of Lemma 1 and is thus omitted due to space limitations. ■

Evaluating the non-zero elements in the first-order sensitivities is as costly as solving $(\frac{N}{2} + \frac{3}{2})n^2 + n$ ODEs over the complete moving horizon interval $[t_{k-N}, t_k]$. To further reduce this complexity, the idea now is to break down the problem of determining the sensitivities $X(t_i)$ and ${}^{j}Z(t_i)$ on the complete interval $[t_{k-N}, t_k]$ to independent problems on the intervals $[t_i, t_{i+1}]$, $i \in \mathcal{K}$. Afterwards, the solutions to these subproblems are assembled in a suitable manner to yield the wanted sensitivities. The advantage of this procedure is that several of these subproblems are identical due to the common underlying structure and thus need to be solved only once. To this end, the following notations are introduced.

Notation 2: The abbreviation X_b^a and ${}^{j}Z_b^a$ denotes the solution of $\dot{X}(t) = \frac{\partial f}{\partial \hat{x}}(t)\, X(t)$ and ${}^{j}\dot{Z}(t) = \frac{\partial f}{\partial \hat{x}}(t)\ {}^{j}Z(t) + \theta_j'(t, \hat{w}_j)$ at the time t_b with the initial value $X(t_a) = I$ and ${}^{j}Z(t_a) = 0$, respectively.

Theorem 1: The first-order sensitivities defined in (16) are

$$X(t_i) = X_i^{i-1} X_{i-1}^{i-2} \dots X_{k-N+2}^{k-N+1} X_{k-N+1}^{k-N} \tag{19a}$$

$${}^{j}Z(t_i) = \begin{cases} 0, & i < j+1 \\ X_i^{i-1} X_{i-1}^{i-2} \dots X_{j+2}^{j+1}\ {}^{j}Z_{j+1}^{j}, & i \geq j+1 \end{cases} \tag{19b}$$

for $i = k-N, \dots, k$ and $j = k-N, \dots, k-1$.

Proof: Based on Lemma 1, the solution $X(t_i)$ is transformed by using Lemma 2 in the following way

$$\begin{aligned} X(t_i) &= \Phi(t_i, t_{k-N}) X(t_{k-N}) \\ &= \Phi(t_i, t_{i-1})\, \Phi(t_{i-1}, t_{i-2}) \dots \Phi(t_{k-N+1}, t_{k-N}) I \\ &= \underbrace{\Phi(t_i, t_{i-1}) I}_{X_i^{i-1}}\ \underbrace{\Phi(t_{i-1}, t_{i-2}) I}_{X_{i-1}^{i-2}} \dots \underbrace{\Phi(t_{k-N+1}, t_{k-N}) I}_{X_{k-N+1}^{k-N}} \\ &= X_i^{i-1} X_{i-1}^{i-2} \dots X_{k-N+2}^{k-N+1} X_{k-N+1}^{k-N} \end{aligned}$$

The solution ${}^{j}Z(t_i) = 0$ for $i < j+1$, i. e. $t_i \leq t_j$, directly results from Lemma 3. For the case $i > j+1$, i. e. $t_i > t_{j+1}$, the solution ${}^{j}Z(t_i)$ is transformed by using Lemma 2 to yield

$$\begin{aligned} {}^{j}Z(t_i) &= \int_{t_j}^{t_{j+1}} \Phi(t_i, \tau)\, \theta_j'(\tau, \hat{w}_j)\, d\tau \\ &= \Phi(t_i, t_{j+1})\ {}^{j}Z_{j+1}^{j} \\ &= \Phi(t_i, t_{i-1})\, \Phi(t_{i-1}, t_{i-2}) \dots \Phi(t_{j+2}, t_{j+1})\ {}^{j}Z_{j+1}^{j} \\ &= \underbrace{\Phi(t_i, t_{i-1}) I}_{X_i^{i-1}}\ \underbrace{\Phi(t_{i-1}, t_{i-2}) I}_{X_{i-1}^{i-2}} \dots \\ &\quad \dots \underbrace{\Phi(t_{j+2}, t_{j+1}) I}_{X_{j+2}^{j+1}}\ {}^{j}Z_{j+1}^{j} \\ &= X_i^{i-1} X_{i-1}^{i-2} \dots X_{j+2}^{j+1}\ {}^{j}Z_{j+1}^{j} \end{aligned}$$

This result is also valid for the case $i = j+1$, i. e. $t_i = t_{j+1}$, because according to Lemma 2 the following equation holds ${}^{j}Z(t_{j+1}) = X_{j+1}^{j+1}\ {}^{j}Z_{j+1}^{j} = \Phi(t_j, t_j) I\ {}^{j}Z_{j+1}^{j} = {}^{j}Z_{j+1}^{j}$ ■

The advantage of this theorem over the finite difference method and the approach described in lemma 1 and 3 is two-fold. First, the number of ODEs that have to be solved over the complete interval $t \in [t_{k-N}, t_k]$ is independent of N, namely $2n^2 + n$. In other words, the complexity of determining the first-order sensitivities is independent of the number of unknowns $\hat{w}_i$ and independent of the number of first-order sensitivities ${}^{j}Z$ itself. Second, each subproblem can be solved independently and thus in parallel.

B. Second-order derivative of the Lagrange function $\mathcal{L}$

The finite differences method can once again be applied to approximate the Hessian of $\mathcal{L}$. However, the computational complexity is even higher than in the former case. An alternative and more commonly used method is to estimate the Hessian by applying a quasi-Newton approximation which measures only the changes in the gradients [10]. The drawback of this approximation method is that it does not explicitly consider the true structure of the Hessian which may result in significant deviations from the real one. As a consequence, the quality of the iterations can be poor, and many costly steps are required to converge to a local optimum. For this reason, the following approach is proposed to improve the convergence speed by exploiting the structure of the Hessian.

Applying the chain rule to (15) yields the Hessian of $\mathcal{L}$

$$\begin{aligned}\frac{\partial^2\mathcal{L}}{\partial\boldsymbol{p}^2} = \frac{\partial^2\Gamma}{\partial\boldsymbol{p}^2} &+ \sum_{i=k-N}^{k}\frac{\partial\tilde{x}_i}{\partial\boldsymbol{p}}^T\frac{\partial^2\Upsilon_i}{\partial\tilde{x}_i^2}\frac{\partial\tilde{x}_i}{\partial\boldsymbol{p}} + \sum_{i=k-N}^{k}\sum_{j=1}^{n}\frac{\partial\Upsilon_i}{\partial\tilde{x}_{i,j}}\frac{\partial^2\tilde{x}_{i,j}}{\partial\boldsymbol{p}^2}\\ &+ \sum_{i=k-N}^{k-1}\frac{\partial^2\Psi_i}{\partial\boldsymbol{p}^2} - \sum_{i=k-N}^{k}\sum_{j=1}^{n}\lambda_{i,j}\frac{\partial\tilde{x}_i}{\partial\boldsymbol{p}}^T\frac{\partial^2 c_{i,j}}{\partial\tilde{x}_i^2}\frac{\partial\tilde{x}_i}{\partial\boldsymbol{p}}\\ &- \sum_{i=k-N}^{k}\sum_{j=1}^{n}\left(\lambda_i^T\frac{\partial c_i}{\partial\tilde{x}_{i,j}}\right)\frac{\partial^2\tilde{x}_{i,j}}{\partial\boldsymbol{p}^2} \end{aligned} \tag{20}$$

where $\partial^2\tilde{x}_{i,j}/\partial\boldsymbol{p}^2$ are the second-order sensitivities of the j-th element of $\tilde{x}_i$. Based on this structure, the Hessian will be partitioned as follows:

$$\frac{\partial^2\mathcal{L}}{\partial\boldsymbol{p}^2} = H_1 + H_2, \tag{21a}$$

where H_1 contains the second derivatives of the objective function and the terms involving the first-order sensitivities:

$$\begin{aligned}H_1 = \frac{\partial^2\Gamma}{\partial\boldsymbol{p}^2} &+ \sum_{i=k-N}^{k}\frac{\partial\tilde{x}_i}{\partial\boldsymbol{p}}^T\frac{\partial^2\Upsilon_i}{\partial\tilde{x}_i^2}\frac{\partial\tilde{x}_i}{\partial\boldsymbol{p}} + \sum_{i=k-N}^{k-1}\frac{\partial^2\Psi_i}{\partial\boldsymbol{p}^2}\\ &- \sum_{i=k-N}^{k}\sum_{j=1}^{n}\lambda_{i,j}\frac{\partial\tilde{x}_i}{\partial\boldsymbol{p}}^T\frac{\partial^2 c_{i,j}}{\partial\tilde{x}_i^2}\frac{\partial\tilde{x}_i}{\partial\boldsymbol{p}}\end{aligned} \tag{21b}$$

and H_2 contains the terms involving the second-order sensitivities:

$$H_2 = \sum_{i=k-N}^{k}\sum_{j=1}^{n}\left(\frac{\partial\Upsilon_i}{\partial\tilde{x}_{i,j}} - \left(\lambda_i^T\frac{\partial c_i}{\partial\tilde{x}_{i,j}}\right)\right)\frac{\partial^2\tilde{x}_{i,j}}{\partial\boldsymbol{p}^2}. \tag{21c}$$

H_1 can be easily calculated due to the already available first-order sensitivities that were necessary for the computation of $\partial\mathcal{L}/\partial\boldsymbol{p}$.

Three cases can be differentiated for H_2 depending on the characteristic of the dynamic of the system: 1) If $\partial^2 f/\partial\hat{x}^2 = 0$ holds, like e. g. for linear systems, then $H_2 = 0$, because the second-order sensitivities are exactly zero. This fact directly results from investigating the second-order matrix differential equations and is omitted here due to space limitations. 2) If the system has only "weak nonlinearities", then $H_2 \approx 0$. Note that this approach shows strong similarities to the Gauss-Newton approach for solving nonlinear least-squares problems. 3) In the general case, neglecting H_2 may result in a poor performance of the optimization algorithm which motivates the inclusion of H_2. Ideally, the exact second-order sensitivities should be calculated. However, this is only reasonable for small-sized systems due to the high complexity involved for solving the $(n+N)n^2$ ODEs for the second-order sensitivity matrix differential equations. Instead of that approach, a feasible way is to approximate H_2 based on successive evaluations of the gradient. More precisely, applying the BFGS update yields

$$H_2^{i+1} = H_2^i - \frac{H_2^i\, z^i\, z^{i,T}\, H_2^i}{z^{i,T}\, H_2^i\, z^i} + \frac{g^i\, g^{i,T}}{g^{i,T}\, z^i}, \tag{22}$$

where the upper index denotes the iteration step, $z^i = \boldsymbol{p}^{i+1} - \boldsymbol{p}^i$, and $g^i = (\partial\mathcal{L}/\partial\boldsymbol{p})^{i+1} - (\partial\mathcal{L}/\partial\boldsymbol{p})^i - H_1^{i+1}z^i$. Modifications, like e. g. damped BFGS or limited-memory BFGS, or other update schemes, like e. g. SR1 updating, can be used when deemed appropriate.

Remark 1: An alternative way for computing $\partial\mathcal{L}/\partial\boldsymbol{p}$ is the adjoint-based (or sometimes called "backward") method. The numerical complexity is comparable to the proposed method, as the number of ODEs that have to be solved in both cases is $2n^2 + n$. However, the adjoint-based approach computes the gradient directly without producing intermediate information useable for the computation of the Hessian $\partial^2\mathcal{L}/\partial\boldsymbol{p}^2$. Therefore, the adjoint-based method is inferior to the proposed reduced sensitivity-based method.

Remark 2: The proposed concept can be easily extended to the case, where unknown parameters a of the system have to be estimated. It is straightforward to derive the first-order sensitivity matrix differential equation for the sensitivities $\partial\tilde{x}_i/\partial a$ and to integrate them in the proposed concept.

Remark 3: If the cost function (4) is used, $\partial^2 f/\partial\hat{x}^2 = 0$ holds and the inequalities are linear, then the SQP method yields the optimal solution after one iteration as soon as the active set is identified.

V. NUMERICAL CASE STUDY

In this section, the aforementioned improvements concerning the derivative computations are applied to an MH estimator for a nonlinear continuously stirred tank reactor (CSTR) [6], [12]:

$$\begin{aligned}\dot{x}_1(t) &= p_1(x_a - x_1(t)) - p_2 x_1(t)\exp\left(-\frac{E_A}{Rx_2(t)}\right) + w_1(t)\\ \dot{x}_2(t) &= p_1(x_b - x_2(t)) + p_3 x_1(t)\exp\left(-\frac{E_A}{Rx_2(t)}\right)\\ &\quad + p_4(u(t) - x_2(t)) + w_2(t).\end{aligned}$$

The system involves two states $x = [x_1, x_2]^T$, corresponding to the concentration and the temperature, respectively, one control u corresponding to the cooling water temperature and two process noise sequences $w = [w_1, w_2]^T$. The initial condition is $x_0 = [0.005, 445]^T$ and the model parameters are $E_A = 11250$, $R = 1.986$, $x_a = 0.02$, $x_b = 340$, $p_1 = 1$, $p_2 = 10^6$, $p_3 = 4.25 \times 10^9$ and $p_4 = 2$. The temperature is used as the output ($y = x_2$) and measured every $0.5s$. These measurements are generated

from a simulated closed-loop feedback control scenario with Gaussian process noise sequences with standard deviations $\sigma_{w_1} = 0.002$ and $\sigma_{w_2} = \sqrt{250}$. Afterwards, the resulting temperatures are corrupted with different levels of Gaussian noise with standard deviation $\sigma_y = 1$ to simulate measurement noise. The cost function defined in (4) is used for the MHE with $P = \mathrm{diag}(10, 0.1)$, $R = 1/\sigma_y^2$ and $Q = \mathrm{diag}(1/\sigma_{w_1}^2, 1/\sigma_{w_2}^2)$. Furthermore, the states should satisfy the inequality $[0, 300]^T \leq \hat{x}_i \leq [0.03, 500]^T$.

In Fig. 2, the performance of the MH estimator which uses an IP method to solve the NLPs is presented where an observation window of $N = 6$ is used. The optimization variable is initialized with $[0.018, 350, 0, \ldots, 0]^T$. Note that the initial estimates are erroneous and yet the MH estimator quickly recovers the true state. The first-order sensitivities are computed according to Theorem 1 and H_2 is neglected.

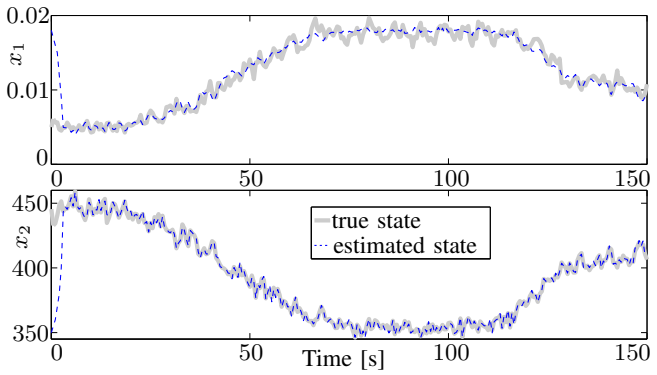

Fig. 2. Comparison between the true and the estimated states.

In Fig. 3, the complexity for computing $\partial\mathcal{L}/\partial\boldsymbol{p}$ according to the sensitivity-based methods 1 (Theorem 1) and 2 (Lemma 1 & 3) compared to the central-difference formula (CDF) (14) is illustrated. The number of measurements $N+1$ is increased in the constant time interval $[0, 100]$ and the time for calculating $\partial\mathcal{L}/\partial\boldsymbol{p}$ is measured. The first and the last measurement are taken at $0s$ and $100s$, respectively, while all other measurements are equally spaced in between. The time needed for the case $N = 1$ where method 1 and 2 are identical corresponds to the complexity 1. The complexity for method 2 and CDF grows linearly with N whereas the slight increase in complexity for method 1 is due to the increasing number of matrix multiplications for forming the overall sensitivities in (19).

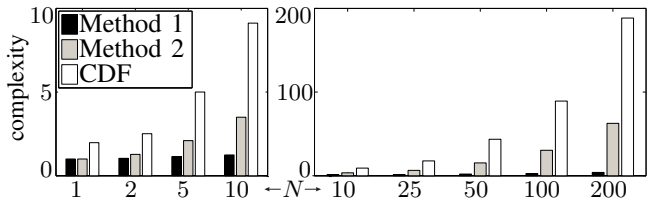

Fig. 3. Comparison of the numerical complexity between different methods for computing $\partial\mathcal{L}/\partial\boldsymbol{p}$. (ODE solver used in all methods: classical 4-step Runge-Kutta)

The solver progress indicated by the cost function value for $N = 30$ can be seen in Table I for two cases of calculating the Hessian. In the first case, the Hessian is approximated by a conventional full BFGS method. In the second case, only the second part of the Hessian H_2 is approximated by the BFGS method (22) and H_1 is calculated by the already available first-order sensitivities (21b). As can be seen, the proposed approach results in less iterations and a faster convergence, especially close to the solution.

TABLE I

COST FUNCTION VALUES FOR THE FIRST 6 ITERATIONS

iteration	1	2	3	4	5	6
full BFGS	112208	38438	2351	557	543	508
only BFGS H_2	112208	9486	727	526	498	[-]

VI. CONCLUSIONS

In this paper, an efficient method for calculating the first and second-order derivatives is presented for the solution of the optimization problem for moving horizon estimation. It is shown, that their computation based on reduced first-order sensitivities is more advantageous than their computation by finite differences, adjoint-based or full quasi-Newton methods. The simulation results substantiate this approach. Future work is directed towards implementing the proposed method in an estimator concept for moving horizon estimation in networked control systems.

ACKNOWLEDGMENTS

The author thanks the German Research Foundation (DFG), Priority Program 1305 for supporting this work.

REFERENCES

[1] M. Alamir, "Optimization based non-linear observers revisited," *Int. J. Control*, vol. 72, no. 13, pp. 1204–1217, 1999.
[2] M. Diehl, H. Ferreau, and N. Haverbeke, "Efficient numerical methods for nonlinear mpc and moving horizon estimation," in *Nonlinear MPC*. Springer Berlin / Heidelberg, 2009, vol. 384, pp. 391–417.
[3] A. Gelb, *Applied Optimal Estimation*. The MIT Press, 1974.
[4] E. Haseltine and J. Rawlings, "Critical evaluation of extended kalman filtering and moving horizon estimation," *Ind. Eng. Chem. Res.*, vol. 44, no. 8, pp. 2451–2460, 2005.
[5] N. Haverbeke, M. Diehl, and B. De Moor, "A structure exploiting interior-point method for moving horizon estimation," in *Proc. of the 48th IEEE Conf. on Decision and Control*, 2009, pp. 1273 –1278.
[6] G. Hicks and W. Ray, "Approximation methods for optimal control synthesis," *Can. J. Chem. Eng.*, vol. 49, pp. 552–529, 1971.
[7] T. Kailath, *Linear Systems*. Prentice Hall, 1980.
[8] H. Michalska and D. Mayne, "Moving horizon observers and observer-based control," *IEEE Trans. Autom. Control*, vol. 40, no. 6, pp. 995–1006, June 1995.
[9] P. Moraal and J. Grizzle, "Observer design for nonlinear systems with discrete-time measurements," *IEEE Trans. Autom. Control*, vol. 40, no. 3, pp. 395–404, March 1995.
[10] J. Nocedal and S. Wright, *Numerical Optimization*. Springer-Verlag, 2000.
[11] C. Rao, J. Rawlings, and D. Mayne, "Constrained state estimation for nonlinear discrete-time systems: Stability and moving horizon approximations," *IEEE Trans. Autom. Control*, vol. 48, no. 2, pp. 246–258, 2003.
[12] A. Uppal, W. Ray, and A. Poore, "On the dynamic behavior of continuous stirred tank reactors," *Chem. Eng. Sci.*, vol. 29, pp. 967–985, 1974.
[13] V. Zavala, C. Laird, and L. Biegler, "A fast moving horizon estimation algorithm based on nonlinear programming sensitivity," *J. of Process Control*, vol. 18, no. 90, pp. 876 – 884, 2008.
[14] G. Zimmer, "State observation by on-line minimization," *Int. J. Control*, vol. 60, no. 4, pp. 595–606, 1994.