Exact State Sensitivity Calculation and its Role in Moving Horizon Estimation for Nonlinear Networked Control Systems

Peter Philipp
Institute of Automatic Control, Technische Universität München
Boltzmannstr. 15, 85748 Garching
E-Mail: peter.philipp@tum.de

This paper deals with the exact state sensitivity calculation required in the moving horizon estimator (MHE) for nonlinear networked control systems (NCS) presented in [4]. This NCS-MHE is capable of dealing with the network induced imperfections by formulating the estimation problem as a suitable optimization problem within a moving horizon framework. The resulting nonlinear program (NLP) can be efficiently solved by an adapted sequential quadratic programming (SQP) approach which exploits the structure in the associated derivatives by utilizing state sensitivities. However, their exact calculation via direct derivation of the state is infeasible as no general closed form solution of the state equation can be given. Therefore, a new method based on first-order sensitivity differential equations is proposed which provides several advantages compared to a finite difference method, like e. g. exactness, reduced numerical complexity and a higher degree of parallelization.

1 Introduction

Networked control systems (NCS) are spatially distributed systems in which the communication between sensors, actuators, and controllers occurs through a shared digital communication network, see [1, 5]. The main winning features of those NCS come from their low cost, their high flexibility and easy re-configurability, their natural reliability and robustness to failure, and their adaptation capability leading to new control oriented possibilities. In contrast to conventional control theory, there are also some drawbacks due to the characteristic features of the communication method adopted, namely there is the need to deal with the following problems: (1) Data transmission through the communication network unavoidably introduces *time delays* and *packet reordering*; (2) Data traffic congestion, data collision or interference cause *packet loss*; (3) *Limited energy supply* of the sensor nodes require strategies which minimize the communication effort while maintaining a certain performance level; (4) Spatially distributed network elements possess *unsynchronized clocks* leading to unsynchronized timescales.
If only the measurements of a wired or wireless sensor are transmitted through a shared multipurpose network, this directly results in the NCS structure depicted in Figure 1. In a previous

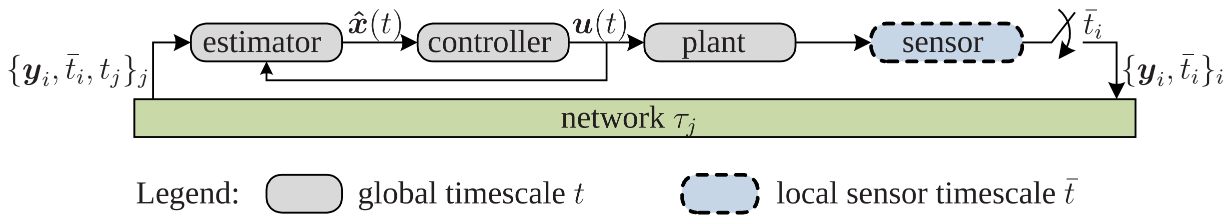

Figure 1: Structure of the Networked Control System.

work [4], a nonlinear estimator has been developed that is capable of dealing with all the aforementioned problems by adapting the conventional MHE framework. The estimator requires an NLP to be solved whenever new measurements arrive at the estimator. Its solution can be efficiently calculated by means of an adapted SQP approach which exploits the structure in the associated derivatives by utilizing state sensitivities. Concerning real-time applications, the performance of the NCS-MHE strongly depends on methods which calculate these state sensitivities fast and accurate. However, the finite difference approach does not fulfill any of these requirements. Therefore, the main objective of this article is to propose a new exact and efficient method to calculate the state sensitivities by utilizing sensitivity differential equations.
The remainder of this paper is organized as follows: Before in section 3 the role of state sensitivities for the NCS-MHE is detailed, the NCS-MHE presented in [4] is summarized in section 2. The exact state sensitivity calculation problem is formulated in section 4. The key goal of this paper, namely the exact state sensitivity computation is presented in section 5 before the paper is concluded in section 6.

2 The Moving Horizon Estimator for Networked Control Systems

In this section, the main part of the NCS-MHE presented in [4] for the NCS structure depicted in Figure 1 is summarized. To this end, the following notations and assumptions are introduced.

Notation 1. For a vector $\boldsymbol{x} \in \mathbb{R}^n$, $\hat{\boldsymbol{x}}$ denotes its estimated value. For two vectors $\boldsymbol{x} \in \mathbb{R}^n$ and $\boldsymbol{y} \in \mathbb{R}^m$, let $col(\boldsymbol{x}, \boldsymbol{y})$ denote the column vector in $\mathbb{R}^{n+m}$ where $\boldsymbol{x}$ and $\boldsymbol{y}$ are stacked into a single column. For two vectors $\boldsymbol{x}, \boldsymbol{z} \in \mathbb{R}^n$, $\boldsymbol{x} \geq \boldsymbol{z}$ denotes componentwise inequality. A function with k continuous derivatives is called a $\mathcal{C}^k$ function. The times t and $\bar{t}$ denote the global time and the local sensor time, respectively.

Assumption 1. The relation between t and $\bar{t}$ in a sufficiently small time interval is given by the clock model $t = s\,\bar{t} + t_o$.

The plant is described by the nonlinear continuous-time system

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{u}(t)) + \boldsymbol{w}(t) \tag{1a}$$
$$\boldsymbol{y}(t) = \boldsymbol{h}(\boldsymbol{x}(t)) + \boldsymbol{v}(t), \tag{1b}$$

where $\boldsymbol{x}(t) \in \mathbb{R}^n$, $\boldsymbol{u}(t) \in \mathbb{R}^m$, $\boldsymbol{w}(t) \in \mathbb{R}^n$, $\boldsymbol{y}(t) \in \mathbb{R}^p$ and $\boldsymbol{v}(t) \in \mathbb{R}^p$ are respectively the state, input, disturbance, measurement and noise vector. The measurement vector $\boldsymbol{y}(t)$ is sampled by the sensor at the local sensor times $\bar{t}_i$ and these sampled measurements are denoted with $\boldsymbol{y}_i$. Afterwards, the sampled measurement $\boldsymbol{y}_i$ is transmitted through the communication network together with its relative time stamp $\bar{t}_i$ as a packet of the form $\{\boldsymbol{y}_i, \bar{t}_i\}_i$. If the packet arrives at the estimator, the packet is augmented by the arrival time stamp t_j to yield the packet $\{\boldsymbol{y}_i, \bar{t}_i, t_j\}_j$. The estimator stores $N+1$ packets in a buffer and sorts them by the relative time stamps $\bar{t}_i$ in ascending order and assigns the indices $k-N, \dots, k$ to them, where k denotes the last packet in the sorted buffer, see Algorithm 1 in [4].

Notation 2. The sets $\mathcal{I} = \{k-N, k-N+1, \dots, k\}$, $\underline{\mathcal{I}} = \mathcal{I} \setminus \{k-N\}$ and $\overline{\mathcal{I}} = \mathcal{I} \setminus \{k\}$ denote sets of indices corresponding to the k-th buffer. For a global time depending estimated vector $\hat{\boldsymbol{x}}(t)$, $\hat{\boldsymbol{x}}_i$ denotes its value at the estimated global time $\hat{s}\,\bar{t}_i + \hat{t}_o$ according to the i-th packet in the buffer.

The NCS-MHE consists of two steps, see Algorithm 3 in [4]. The current state $\hat{\boldsymbol{x}}(t)$ is propagated in the *prediction step* by solving the model equations forward in time. Whenever the buffer changes, the *update step* of the NCS-MHE is performed, where the following NLP has to be solved

$$\min_{\hat{s},\hat{t}_o,\hat{\boldsymbol{x}}_{k-N},\dots,\hat{\boldsymbol{x}}_k,\hat{\boldsymbol{w}}_{k-N},\dots,\hat{\boldsymbol{w}}_{k-1}} \Gamma(\hat{s},\hat{t}_o,\hat{\boldsymbol{x}}_{k-N}) + \sum_{i=k-N}^{k} \Upsilon_i(\boldsymbol{y}_i,\hat{\boldsymbol{x}}_i) + \sum_{i=k-N}^{k-1} \Psi_i(\hat{\boldsymbol{w}}_i) \tag{2a}$$

subject to:

$$\hat{\boldsymbol{x}}_{i+1} - \hat{\boldsymbol{x}}_i - \int_{\hat{s}\,\bar{t}_i+\hat{t}_o}^{\hat{s}\,\bar{t}_{i+1}+\hat{t}_o} \boldsymbol{f}(\hat{\boldsymbol{x}}(t),\boldsymbol{u}(t))\,dt - \hat{\boldsymbol{w}}_i = \boldsymbol{0}, \quad i \in \overline{\mathcal{I}} \tag{2b}$$

$$\boldsymbol{c}_i(\hat{\boldsymbol{x}}_i,\hat{\boldsymbol{w}}_i) \geq \boldsymbol{0}, \quad i \in \overline{\mathcal{I}} \tag{2c}$$

$$\boldsymbol{c}_k(\hat{\boldsymbol{x}}_k) \geq \boldsymbol{0} \tag{2d}$$

$$\boldsymbol{d}(\hat{s},\hat{t}_o) \geq \boldsymbol{0}, \tag{2e}$$

where $\boldsymbol{c}_i$ and $\boldsymbol{d}$ are for $i \in \mathcal{I}$ inequality constraints. Subsequently, the current state $\hat{\boldsymbol{x}}(t)$ is updated based on the solution of this NLP. A detailed explanation of this NLP can be found in [4] and is omitted here due to its minor relevance for this paper.
For the functions involved in this NLP, the following assumption is made:

Assumption 2. The functions $\boldsymbol{f},\boldsymbol{h},\Gamma,\Upsilon_i,\Psi_i,\boldsymbol{c}_i$ and $\boldsymbol{d}$ are at least $\mathcal{C}^2$ functions.

Remark 1. For simplicity of presentation, only one sensor is considered. It is straightforward to extend the concept to the general case of several different sensors, each with different timescales $\bar{t}$.

3 The role of the state sensitivities for the MHE-NCS

This section concentrates on the relation between the state sensitivities and the NCS-MHE. To this end, the following notations are introduced.

Notation 3. The vector $\boldsymbol{p} \in \mathbb{R}^{(N+1)n+2}$ denotes the collected optimization variables, i. e. $\boldsymbol{p} = col(\hat{s},\hat{t}_o,\hat{\boldsymbol{x}}_{k-N},\hat{\boldsymbol{w}}_{k-N},\dots,\hat{\boldsymbol{w}}_{k-1})$. The function $\boldsymbol{c} : \mathbb{R}^{(N+1)n+2} \mapsto \mathbb{R}^{(N+1)n+2}$ denotes the overall inequalities, i. e. $\boldsymbol{c} = col\left(\boldsymbol{c}_{k-N}(\hat{\boldsymbol{x}}_{k-N},\hat{\boldsymbol{w}}_{k-N}),\dots,\boldsymbol{c}_{k-1}(\hat{\boldsymbol{x}}_{k-1},\hat{\boldsymbol{w}}_{k-1}),\boldsymbol{c}_k(\hat{\boldsymbol{x}}_k),\boldsymbol{d}(\hat{s},\hat{t}_o)\right)$. The sets $\mathbb{T}_i = \{t \in \mathbb{R}^n | \hat{s}\,\bar{t}_i + \hat{t}_o < t \leq \hat{s}\,\bar{t}_{i+1} + \hat{t}_o\}$ define for $i \in \overline{\mathcal{I}}$ all admissible global times between two consecutive measurements in the k-th buffer. The union of these sets $\mathbb{T} = \bigcup_{i\in\overline{\mathcal{I}}} \mathbb{T}_i$ denotes all admissible global times in the k-th buffer. The scalars $T_i = \hat{s}(\bar{t}_{i+1} - \bar{t}_i)$ denote for $i \in \overline{\mathcal{I}}$ the times between two consecutive measurements in the k-th buffer. The input $\boldsymbol{u}(t)$ for $t \in \mathbb{T}$ is denoted by $\boldsymbol{u}_k$.

The equality constraint (2b) uniquely determines all the states $\hat{\boldsymbol{x}}_i$ in the current moving horizon if the vectors $\boldsymbol{p}$ and $\boldsymbol{u}_k$ are fixed. Thus, an implicit function $\tilde{\boldsymbol{x}}_i(\boldsymbol{p},\boldsymbol{u}_k)$ that satisfies (2b) for all $\boldsymbol{p}$ and $\boldsymbol{u}_k$ can be defined. Consequently, the constraint (2b) can be replaced in the optimization problem by substituting the function $\tilde{\boldsymbol{x}}_i(\boldsymbol{p},\boldsymbol{u}_k)$ with $\hat{\boldsymbol{x}}_i$. Hence, the NLP can be reduced to

$$\min_{\boldsymbol{p}} \Gamma(\hat{s},\hat{t}_o,\hat{\boldsymbol{x}}_{k-N}) + \sum_{i=k-N}^{k} \Upsilon_i\left(\boldsymbol{y}_i,\tilde{\boldsymbol{x}}_i(\boldsymbol{p},\boldsymbol{u}_k)\right) + \sum_{i=k-N}^{k-1} \Psi_i(\hat{\boldsymbol{w}}_i) \tag{3a}$$

$$\text{subject to:}\quad \boldsymbol{c}\left(\tilde{\boldsymbol{x}}_{k-N}(\boldsymbol{p},\boldsymbol{u}_k),\ldots,\tilde{\boldsymbol{x}}_k(\boldsymbol{p},\boldsymbol{u}_k),\hat{\boldsymbol{w}}_{k-N},\ldots,\hat{\boldsymbol{w}}_{k-1},\hat{s},\hat{t}_o\right)\geq \boldsymbol{0}. \tag{3b}$$

The advantage of this approach compared to the original problem is the by Nn reduced dimension of the optimization variable space. To further simplify the analysis, the arguments of all functions are in the following suppressed from the notation when the meaning is otherwise clear.

The NLP (3) can be iteratively solved by applying the *sequential quadratic programming* (SQP) method. The basic idea of the SQP approach is to linearize in every iteration step the *Karush-Kuhn-Tucker* (KKT) conditions. It turns out that the resulting linear complementary system can be interpreted as the KKT conditions of the following *quadratic program* (QP)

$$\min_{\Delta \boldsymbol{p}} \frac{\partial \mathcal{L}}{\partial \boldsymbol{p}}^T \Delta \boldsymbol{p} + \frac{1}{2}\Delta \boldsymbol{p}^T \frac{\partial^2 \mathcal{L}}{\partial \boldsymbol{p}^2}\Delta \boldsymbol{p}, \quad \text{subject to: } \boldsymbol{c} + \frac{\partial \boldsymbol{c}}{\partial \boldsymbol{p}}\Delta \boldsymbol{p} \geq \boldsymbol{0}. \tag{4a}$$

Thereby, $\mathcal{L}$ is the *Lagrange function*

$$\mathcal{L}(\boldsymbol{p},\boldsymbol{\lambda}) = \Gamma + \sum_{i=k-N}^{k} \Upsilon_i + \sum_{i=k-N}^{k-1} \Psi_i - \boldsymbol{\lambda}^T \boldsymbol{c}, \tag{4b}$$

where $\boldsymbol{\lambda} = col(\boldsymbol{\lambda}_{k-N},\ldots,\boldsymbol{\lambda}_k,\boldsymbol{\mu}) \in \mathbb{R}^{(N+1)n+2}$ is a vector of Lagrange multipliers. The accuracy as well as the execution time of the SQP method mainly depend on an accurate and fast computation of the required derivatives $\partial^i \mathcal{L}/\partial \boldsymbol{p}^i, i=1,2$ of the Lagrange function.

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

$$\frac{\partial \mathcal{L}}{\partial p_i} \approx \frac{\mathcal{L}(\boldsymbol{p}+\epsilon\, \boldsymbol{e}_i) - \mathcal{L}(\boldsymbol{p}-\epsilon\, \boldsymbol{e}_i)}{2\epsilon}, \qquad i=1,\ldots,(N+1)n+2, \tag{5}$$

where ϵ is a small positive scalar and $\boldsymbol{e}_i$ is the i-th unit vector. However, it is not recommended to use this method here due to its high numerical complexity and its poor accuracy. The evaluation of (5) is as costly as solving $2n((N+1)n+2)$ ODEs over $\mathbb{T}$.

A more subtle approach is to calculate the exact derivatives of (4b) by means of state sensitivities which leads to the following proposition.

Proposition 1. *The exact gradient and Hessian of the Lagrange function $\mathcal{L}$ defined in* (4b) *with respect to $\boldsymbol{p}$ are*

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{p}} = \frac{\partial \Gamma}{\partial \boldsymbol{p}} + \sum_{i=k-N}^{k} \frac{\partial \tilde{\boldsymbol{x}}_i}{\partial \boldsymbol{p}}^T \frac{\partial \Upsilon_i}{\partial \tilde{\boldsymbol{x}}_i} + \sum_{i=k-N}^{k-1} \frac{\partial \Psi_i}{\partial \boldsymbol{p}} - \sum_{i=k-N}^{k} \frac{\partial \tilde{\boldsymbol{x}}_i}{\partial \boldsymbol{p}}^T \frac{\partial \boldsymbol{c}_i}{\partial \tilde{\boldsymbol{x}}_i}^T \boldsymbol{\lambda}_i - \frac{\partial \boldsymbol{d}}{\partial \boldsymbol{p}}^T \boldsymbol{\mu} \tag{6a}$$

$$\begin{aligned}\frac{\partial^2 \mathcal{L}}{\partial \boldsymbol{p}^2} = {} & \frac{\partial^2 \Gamma}{\partial \boldsymbol{p}^2} + \sum_{i=k-N}^{k} \frac{\partial \tilde{\boldsymbol{x}}_i}{\partial \boldsymbol{p}}^T \frac{\partial^2 \Upsilon_i}{\partial \tilde{\boldsymbol{x}}_i^2} \frac{\partial \tilde{\boldsymbol{x}}_i}{\partial \boldsymbol{p}} + \sum_{i=k-N}^{k-1} \frac{\partial^2 \Psi_i}{\partial \boldsymbol{p}^2} - \sum_{i=k-N}^{k} \sum_{j=1}^{n} \lambda_{i,j} \frac{\partial \tilde{\boldsymbol{x}}_i}{\partial \boldsymbol{p}}^T \frac{\partial^2 c_{i,j}}{\partial \tilde{\boldsymbol{x}}_i^2} \frac{\partial \tilde{\boldsymbol{x}}_i}{\partial \boldsymbol{p}} \\ & - \sum_{j=1}^{q+2} \mu_j \frac{\partial^2 d_j}{\partial \boldsymbol{p}^2} + \sum_{i=k-N}^{k} \sum_{j=1}^{n} \left(\frac{\partial \Upsilon_i}{\partial \tilde{x}_{i,j}} - \left(\boldsymbol{\lambda}_i^T \frac{\partial \boldsymbol{c}_i}{\partial \tilde{x}_{i,j}} \right) \right) \frac{\partial^2 \tilde{x}_{i,j}}{\partial \boldsymbol{p}^2}. \end{aligned} \tag{6b}$$

where μ_j, d_j, $c_{i,j}$, $\lambda_{i,j}$ and $\tilde{x}_{i,j}$ denote the j-th element of $\boldsymbol{\mu}$, $\boldsymbol{d}$, $\boldsymbol{c}_i$, $\boldsymbol{\lambda}_i$ and $\tilde{\boldsymbol{x}}_i$, respectively.

Proof. Equations (6a) and (6b) result from differentiating (4b) with respect to $\boldsymbol{p}$ once and twice, respectively, where the dependence of $\tilde{\boldsymbol{x}}_i$ on $\boldsymbol{p}$ has to be taken into account. $\square$

Note that the complete gradient and almost the complete Hessian information depend on the first-order state sensitivities $\partial\tilde{\boldsymbol{x}}_i/\partial\boldsymbol{p}$. Therefore, it is often admissible to approximate or even neglect the second-order sensitivities $\partial^2\tilde{x}_{i,j}/\partial\boldsymbol{p}^2$, see [3]. In other words, one major advantage of this method is that the calculation of the first-order state sensitivities, which are necessary for the gradient of $\mathcal{L}$, often provides an excellent approximation of the Hessian of $\mathcal{L}$ for free.

4 Problem formulation

Before dealing with the first-order state sensitivities, the implicit function $\tilde{\boldsymbol{x}}_i(\boldsymbol{p}, \boldsymbol{u}_k)$ in the current moving horizon has to be investigated first. The nonlinear continuous-time system defined in (1) can be seen in the upper left of Figure 2. The sampled and afterwards transmitted mea-

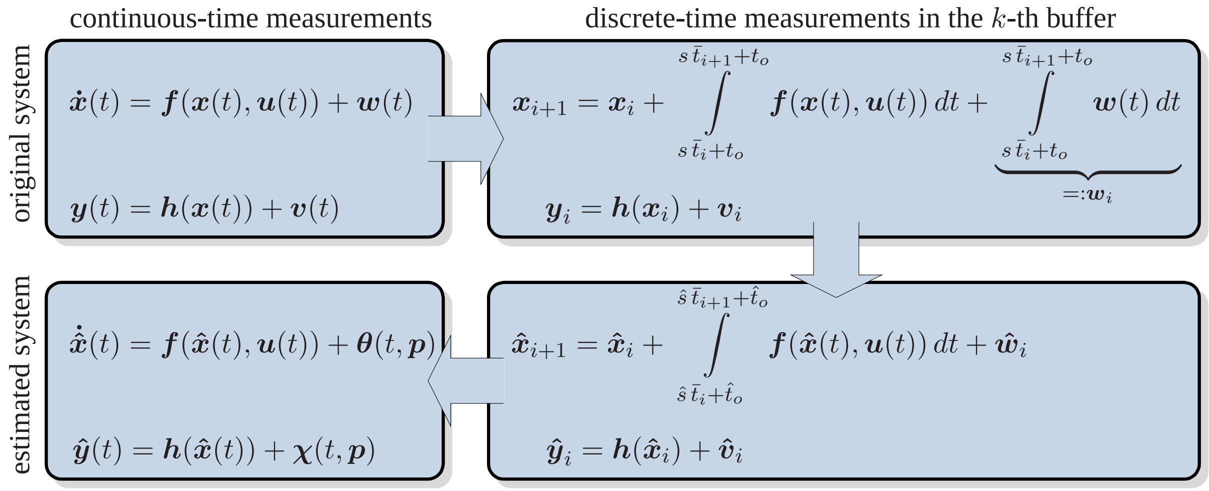

Figure 2: Derivation of the system equation in the update step of the NCS-MHE for $t \in \mathbb{T}$.

surements $\boldsymbol{y}_i$ can be derived by integrating (1) over the sets $\mathbb{T}_i$ and be seen in the upper right of Figure 2. Note that $\boldsymbol{w}_i$ is the integral value of $\boldsymbol{w}(t)$ over the set $\mathbb{T}_i$. The NCS-MHE tries to mimic this behavior and thus utilizes an estimation of this integrated system dynamics, see (2b) and the lower right of Figure 2. In order to evaluate this integrated system dynamics, however, $\hat{\boldsymbol{x}}(t)$ is required which is the solution of the associated estimated continuous-time system. This system is represented as an ODE and is depicted in the lower left of Figure 2, where $\boldsymbol{\theta} : \mathbb{T} \times \mathbb{R}^{(N+1)n+2} \mapsto \mathbb{R}^n$ and $\boldsymbol{\chi} : \mathbb{T} \times \mathbb{R}^{(N+1)n+2} \mapsto \mathbb{R}^p$ are functions using the information in the optimization variables $\boldsymbol{p}$ of the current moving horizon to generate an estimation of the disturbance vector $\boldsymbol{w}(t)$ and the noise vector $\boldsymbol{v}(t)$, respectively. It is important to note, that without any further knowledge about the characteristics of $\boldsymbol{w}(t)$ and $\boldsymbol{v}(t)$, the mappings $\boldsymbol{\theta}(t, \boldsymbol{p})$ and $\boldsymbol{\chi}(t, \boldsymbol{p})$ and thus the ODE cannot be defined uniquely although their integrated representation is unique. This is due to the fact that only measurements at discrete times are available which means, that only the integral value of $\boldsymbol{w}(t)$ over the set $\mathbb{T}_{i-1}$ and only $\boldsymbol{v}(\hat{s}\,\bar{t}_i + \hat{t}_o)$ influences the measurement $\boldsymbol{y}_i$.
Fortunately, the NCS-MHE requires the estimated measurements only at discrete times and thus the mapping $\boldsymbol{\chi}(t, \boldsymbol{p})$ is not required. However, the mapping $\boldsymbol{\theta}(t, \boldsymbol{p})$ is needed for the estimated continuous-time system and thus for the calculation of $\hat{\boldsymbol{y}}_i$. Therefore, the following assumption is made:

Assumption 3. The function $\boldsymbol{\theta} : \mathbb{T} \times \mathbb{R}^{(N+1)n+2} \mapsto \mathbb{R}^n$ in a moving horizon interval is

$$\boldsymbol{\theta}(t, \boldsymbol{p}) = \frac{\hat{\boldsymbol{w}}_i}{\hat{s}\,(\bar{t}_{i+1} - \bar{t}_i)}, \quad \forall t \in \{t \in \mathbb{T}_i | i \in \overline{\mathcal{I}}\},$$

where the affiliation of t to the sets $\mathbb{T}_i$ determines the index i.

This choice of $\boldsymbol{\theta}(t, \boldsymbol{p})$ leads to a piecewise constant disturbance and satisfies the integral relation $\int_{\hat{s}\,\bar{t}_i + \hat{t}_o}^{\hat{s}\,\bar{t}_{i+1} + \hat{t}_o} \boldsymbol{\theta}(t, \boldsymbol{p})\, dt = \hat{\boldsymbol{w}}_i$. This is illustrated in Figure 3.

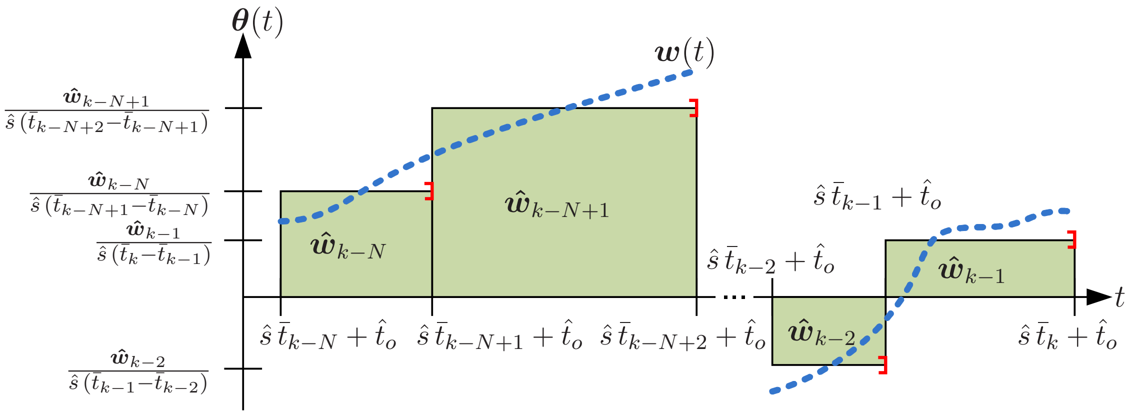

Figure 3: Relation between $\boldsymbol{w}(t)$, $\hat{\boldsymbol{w}}_i$ and $\boldsymbol{\theta}(t)$.

Now the implicit function $\tilde{\boldsymbol{x}}(t, \boldsymbol{p}, \boldsymbol{u}_k)$ and the first-order state sensitivities can be defined precisely:

Definition 1. The implicit function $\tilde{\boldsymbol{x}}(t, \boldsymbol{p}, \boldsymbol{u}_k)$ is a mapping $\tilde{\boldsymbol{x}} : \mathbb{T} \times \mathbb{R}^{(N+1)n+2} \times \mathbb{R}^m \mapsto \mathbb{R}^n$ that satisfies the estimated nonlinear ODE for $t \in \mathbb{T}$

$$\frac{\partial \tilde{\boldsymbol{x}}}{\partial t}(t, \boldsymbol{p}, \boldsymbol{u}_k) = \boldsymbol{f}(\tilde{\boldsymbol{x}}(t, \boldsymbol{p}, \boldsymbol{u}_k), \boldsymbol{u}(t)) + \boldsymbol{\theta}(t, \boldsymbol{p}) \tag{7}$$

with the initial value $\tilde{\boldsymbol{x}}(\hat{s}\,\bar{t}_{k-N} + \hat{t}_o, \boldsymbol{p}, \boldsymbol{u}_k) = \tilde{\boldsymbol{x}}_{k-N} = \hat{\boldsymbol{x}}_{k-N}$.

Note that in accordance with the introduced notation, $\tilde{\boldsymbol{x}}_i$ denotes $\tilde{\boldsymbol{x}}_i(\boldsymbol{p}, \boldsymbol{u}_k) = \tilde{\boldsymbol{x}}(\hat{s}\,\bar{t}_i + \hat{t}_o, \boldsymbol{p}, \boldsymbol{u}_k)$.

Definition 2. The first-order state sensitivities are the first-order derivatives of the implicit function $\tilde{\boldsymbol{x}}(t, \boldsymbol{p}, \boldsymbol{u}_k)$ with respect to $\boldsymbol{p}$ for $t \in \mathbb{T}$.

The naturally arising question is, how to efficiently calculate the exact first-order state sensitivities required for the derivatives of $\mathcal{L}$ in (6).

5 Exact first-order state sensitivity calculation

The most obvious idea to calculate the first-order state sensitivities would be to directly differentiate the implicit function $\tilde{\boldsymbol{x}}_i$ with respect to $\boldsymbol{p}$. However, this procedure is not viable as no general closed form representation of the implicit function $\tilde{\boldsymbol{x}}_i$ can be given. Due to the fact that $\tilde{\boldsymbol{x}}_i$ is described as a solution to an ODE, a natural approach is to describe the first-order state sensitivities as (possibly appropriately combined) solutions to appropriate ODEs, too. To this end, the first-order state sensitivities

$$\frac{\partial \tilde{\boldsymbol{x}}_i}{\partial \boldsymbol{p}} = \Bigg[\underbrace{\frac{\partial \tilde{\boldsymbol{x}}_i}{\partial \hat{s}}, \frac{\partial \tilde{\boldsymbol{x}}_i}{\partial \hat{t}_o}}_{\text{category 2}},\ \underbrace{\frac{\partial \tilde{\boldsymbol{x}}_i}{\partial \hat{\boldsymbol{x}}_{k-N}}, \frac{\partial \tilde{\boldsymbol{x}}_i}{\partial \hat{\boldsymbol{w}}_{k-N}}, \dots, \frac{\partial \tilde{\boldsymbol{x}}_i}{\partial \hat{\boldsymbol{w}}_{k-1}}}_{\text{category 1}}\Bigg], \quad i \in \mathcal{I} \tag{8}$$

can be classified into the categories 1 and 2, depending on the required mathematical representation level to fully describe the influence of the different parameters in $\boldsymbol{p}$ on $\tilde{\boldsymbol{x}}_i$. In other words, this classification answers the question which level is sufficient to investigate in order to calculate the first-order state sensitivities. In category 1, the influence of the parameters on $\tilde{\boldsymbol{x}}_i$ can be fully covered on the ODE level, i. e. it is sufficient to investigate (7). In category 2, the integrated system dynamics or the so-called weak solutions

$$\tilde{\boldsymbol{x}}_i(\boldsymbol{p}, \boldsymbol{u}_k) = \hat{\boldsymbol{x}}_{k-N} + \int_{\hat{s}\,\bar{t}_{k-N}+\hat{t}_o}^{\hat{s}\,\bar{t}_i+\hat{t}_o} \boldsymbol{f}(\tilde{\boldsymbol{x}}(t, \boldsymbol{p}, \boldsymbol{u}_k), \boldsymbol{u}(t)) + \boldsymbol{\theta}(t, \boldsymbol{p})\, dt, \quad i \in \mathcal{I} \tag{9}$$

are additionally required to fully cover the influence of the corresponding parameters.
In the following, the problem of efficiently computing the exact first-order state sensitivities in category 1 is tackled. To this end, the following notations are introduced.

Notation 4. The abbreviation $\boldsymbol{X}(t)$ and ${}^{j}\boldsymbol{Z}(t)$ denote for $j \in \overline{\mathcal{I}}$ the first-order sensitivity $\partial\tilde{\boldsymbol{x}}(t)/\partial\hat{\boldsymbol{x}}_{k-N}$ and $\partial\tilde{\boldsymbol{x}}(t)/\partial\hat{\boldsymbol{w}}_j$, respectively.

Lemma 2. *The first-order sensitivity $\boldsymbol{X}(t) \in \mathbb{R}^{n\times n}$ satisfies the following first-order sensitivity matrix differential equation for $t \in \mathbb{T}$*

$$\dot{\boldsymbol{X}}(t) = \frac{\partial \boldsymbol{f}}{\partial \hat{\boldsymbol{x}}}(t)\, \boldsymbol{X}(t), \tag{10a}$$

with the initial value

$$\boldsymbol{X}(\hat{s}\,\bar{t}_{k-N} + \hat{t}_o) = \boldsymbol{I}. \tag{10b}$$

The unique solution is

$$\boldsymbol{X}(t) = \boldsymbol{\phi}(t, \hat{s}\,\bar{t}_{k-N} + \hat{t}_o)\, \boldsymbol{X}(\hat{s}\,\bar{t}_{k-N} + \hat{t}_o), \tag{10c}$$

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

Proof. The proof is sketched as follows: Equation (10a) results from differentiating (7) with respect to $\hat{\boldsymbol{x}}_{k-N}$. Equation (10b) results from differentiating the initial value on the moving horizon interval. The solution approach for linear time-varying systems (see [2]) applied to (10a) combined with (10b) leads to (10c).

Lemma 3. *The first-order sensitivity ${}^{j}\boldsymbol{Z}(t) \in \mathbb{R}^{n\times n}$ satisfies the following first-order sensitivity matrix differential equation for $t \in \mathbb{T}$*

$${}^{j}\dot{\boldsymbol{Z}}(t) = \frac{\partial \boldsymbol{f}}{\partial \hat{\boldsymbol{x}}}(t)\ {}^{j}\boldsymbol{Z}(t) + \begin{cases} \boldsymbol{0}, & t \in \bigcup\limits_{i=k-N}^{j-1} \mathbb{T}_i \\ \dfrac{1}{\hat{s}\,T_j}\boldsymbol{I}, & t \in \mathbb{T}_j \\ \boldsymbol{0}, & t \in \bigcup\limits_{i=j+1}^{k-1} \mathbb{T}_i \end{cases} \tag{11a}$$

with the initial value

$$ {}^{j}\boldsymbol{Z}(\hat{s}\,\bar{t}_{k-N} + \hat{t}_o) = \boldsymbol{0}. \tag{11b} $$

The unique solution is

$$ {}^{j}\boldsymbol{Z}(t) = \begin{cases} \boldsymbol{0}, & t \in \bigcup_{i=k-N}^{j-1} \mathbb{T}_i \\ \int_{\hat{s}\,\bar{t}_j+\hat{t}_o}^{t} \boldsymbol{\phi}(t,\tau)\,\frac{1}{\hat{s}\,T_j}\,d\tau, & t \in \mathbb{T}_j \\ \int_{\hat{s}\,\bar{t}_j+\hat{t}_o}^{\hat{s}\,\bar{t}_{j+1}+\hat{t}_o} \boldsymbol{\phi}(t,\tau)\,\frac{1}{\hat{s}\,T_j}\,d\tau, & t \in \bigcup_{i=j+1}^{k-1} \mathbb{T}_i \end{cases} \tag{11c} $$

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

Proof. The proof is sketched as follows: Equation (11a) results from differentiating (7) with respect to $\hat{\boldsymbol{w}}_j$. Equation (11b) results from differentiating the initial value on the moving horizon interval. The solution approach for linear time-varying systems (see [2]) applied to (11a) combined with (11b) leads to (11c).

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 $\mathbb{T}$. To further reduce this complexity, the idea now is to break down the problem of determining the sensitivities $\boldsymbol{X}(\hat{s}\,\bar{t}_i+\hat{t}_o)$ and ${}^{j}\boldsymbol{Z}(\hat{s}\,\bar{t}_i+\hat{t}_o)$ on the set $\mathbb{T}$ to independent problems on the sets $\mathbb{T}_i$, $i \in \overline{\mathcal{I}}$. Afterwards, the solutions to these subproblems are assembled in a suitable manner to yield the desired sensitivities. The advantage of this procedure is two-fold: first, several of these subproblems are identical due to the common underlying structure and thus need to be solved only once; second, the solutions to these subproblems can be used to calculate the first-order sensitivities in category 2. To this end, the following notations are introduced.

Notation 5. The abbreviation $\boldsymbol{X}_b^a$ and ${}^{j}\boldsymbol{Z}_b^a$ denotes the solution of (10a) and (11a) at the time $\hat{s}\,\bar{t}_b + \hat{t}_o$ with the initial value $\boldsymbol{I}$ and $\boldsymbol{0}$ at the initial time $\hat{s}\,\bar{t}_a + \hat{t}_o$, respectively.

Lemma 4. *The solution $\boldsymbol{X}_b^a$ and ${}^{d}\boldsymbol{Z}_b^a$ satisfy for $a, b, c \in \mathcal{I}$ and $d \in \overline{\mathcal{I}}$ with $a \le d < b < c$ the following properties*

$$ \begin{array}{rrcl@{\qquad}rrcl} (i) & \boldsymbol{X}_c^a & = & \boldsymbol{X}_c^b \boldsymbol{X}_b^a & (iv) & {}^{d}\boldsymbol{Z}_b^a & = & {}^{d}\boldsymbol{Z}_b^d \\ (ii) & (\boldsymbol{X}_b^a)^{-1} & = & \boldsymbol{X}_a^b & (v) & {}^{d}\boldsymbol{Z}_b^d & = & \boldsymbol{X}_b^{d+1}\,{}^{d}\boldsymbol{Z}_{d+1}^d \\ (iii) & \boldsymbol{X}_a^a & = & \boldsymbol{I} & (vi) & {}^{d}\boldsymbol{Z}_a^a & = & \boldsymbol{0}. \end{array} $$

Proof. The proof is based on the properties of the state transition matrix $\boldsymbol{\phi}(\cdot)$ which are stated in [2]: (i) $\boldsymbol{\phi}(t_2,t_0) = \boldsymbol{\phi}(t_2,t_1)\,\boldsymbol{\phi}(t_1,t_0)$, (ii) $\boldsymbol{\phi}(t_1,t_0)^{-1} = \boldsymbol{\phi}(t_0,t_1)$ and (iii) $\boldsymbol{\phi}(t_0,t_0) = \boldsymbol{I}$. Furthermore, the Notation 5 implies together with (10c) $\boldsymbol{X}_b^a = \boldsymbol{\phi}(\hat{s}\,\bar{t}_b + \hat{t}_o, \hat{s}\,\bar{t}_a + \hat{t}_o)$. The combination of both facts proofs the properties (i)-(iii). Writing the first case of the solution (11c) in terms of the Notation 5 proofs the property (iv). For $i > j$, the solution (11c) is transformed by using the properties of the state transition matrix

$$ {}^{j}\boldsymbol{Z}_i^{k-N} = \int_{\hat{s}\,\bar{t}_j+\hat{t}_o}^{\hat{s}\,\bar{t}_{j+1}+\hat{t}_o} \boldsymbol{\phi}(\hat{s}\,\bar{t}_i + \hat{t}_o, \tau)\,\frac{1}{\hat{s}\,T_j}\boldsymbol{I}\,d\tau = \boldsymbol{\phi}(\hat{s}\,\bar{t}_i + \hat{t}_o, \hat{s}\,\bar{t}_{j+1} + \hat{t}_o)\,{}^{j}\boldsymbol{Z}_{j+1}^j = \boldsymbol{X}_i^{j+1}\,{}^{j}\boldsymbol{Z}_{j+1}^j $$

which proofs the property (v). The property (vi) is just the initial value $\boldsymbol{0}$. □

Theorem 5. *The first-order sensitivities* $\dfrac{\partial \tilde{\boldsymbol{x}}_i}{\partial \hat{\boldsymbol{x}}_{k-N}}$ *and* $\dfrac{\partial \tilde{\boldsymbol{x}}_i}{\partial \hat{\boldsymbol{w}}_j}$ *in* (8) *are for* $i \in \mathcal{I}$

$$\frac{\partial \tilde{\boldsymbol{x}}_i}{\partial \hat{\boldsymbol{x}}_{k-N}} = \boldsymbol{X}_i^{i-1} \boldsymbol{X}_{i-1}^{i-2} \dots \boldsymbol{X}_{k-N+1}^{k-N} \tag{12a}$$

$$\frac{\partial \tilde{\boldsymbol{x}}_i}{\partial \hat{\boldsymbol{w}}_j} = \begin{cases} 0, & i < j+1 \\ \boldsymbol{X}_i^{i-1} \boldsymbol{X}_{i-1}^{i-2} \dots \boldsymbol{X}_{j+2}^{j+1}\, {}^{j}\boldsymbol{Z}_{j+1}^{j}, & i \geq j+1 \end{cases}, \quad j \in \overline{\mathcal{I}}. \tag{12b}$$

Proof. The proof is sketched as follows: Equation (12a) and (12b) are derived by applying the properties stated in Lemma 4 to the solutions (10c) and (11c), respectively. □

The advantage of this Theorem over the finite difference method and the approach described in Lemma 2 and 3 is two-fold. First, the number of ODEs that have to be solved over $\mathbb{T}$ 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 $\hat{\boldsymbol{w}}_i$ and independent of the number of first-order sensitivities ${}^{j}\boldsymbol{Z}$. Second, each subproblem can be solved independently and thus in parallel.

Now, the first-order sensitivities in category 2 are considered. The weak solution (9) can be interpreted as a parameter integral in terms of $\hat{s}$ and $\hat{t}_o$. Therefore, the Leibnitz-integral rule is stated in the following Lemma.

Lemma 6. *Let* $\alpha(p)$ *and* $\beta(p)$ *be* $\mathcal{C}^1$ *functions. Suppose that both* $\boldsymbol{f}(p,t)$ *and* $\partial \boldsymbol{f}(p,t)/\partial p$ *are continuous in the variables* p *and* t*. Then* $\int_{\alpha(p)}^{\beta(p)} \boldsymbol{f}(p,t)\, dt$ *exists as a continuously differentiable function of* p*, with derivative*

$$\frac{\partial}{\partial p} \int_{\alpha(p)}^{\beta(p)} \boldsymbol{f}(p,t)\, dt = \int_{\alpha(p)}^{\beta(p)} \frac{\partial \boldsymbol{f}(p,t)}{\partial p}\, dt + \frac{\partial \beta(p)}{\partial p} \boldsymbol{f}(p, \beta(p)) - \frac{\partial \alpha(p)}{\partial p} \boldsymbol{f}(p, \alpha(p)). \tag{13}$$

Unfortunately, this Lemma cannot be applied directly to the weak solution (9). Due to the choice of $\boldsymbol{\theta}(t,\boldsymbol{p})$, the integrand $\boldsymbol{f}(\tilde{\boldsymbol{x}}(t,\boldsymbol{p},\boldsymbol{u}_k), \boldsymbol{u}(t)) + \boldsymbol{\theta}(t,\boldsymbol{p})$ and its derivative with respect to $\boldsymbol{p}$ are only continuous in $t \in \mathbb{T}_i$ and not in $t \in \mathbb{T}$. Consequently, the Leibnitz-rule is applied only to

$$\tilde{\boldsymbol{x}}_{i+1}(\boldsymbol{p},\boldsymbol{u}_k) = \tilde{\boldsymbol{x}}_i(\boldsymbol{p},\boldsymbol{u}_k) + \int_{\hat{s}\,\bar{t}_i + \hat{t}_o}^{\hat{s}\,\bar{t}_{i+1} + \hat{t}_o} \boldsymbol{f}(\tilde{\boldsymbol{x}}(t,\boldsymbol{p},\boldsymbol{u}_k), \boldsymbol{u}(t)) + \boldsymbol{\theta}(t,\boldsymbol{p})\, dt, \quad i \in \overline{\mathcal{I}}. \tag{14}$$

For later use, note that there is a difference between $\partial \tilde{\boldsymbol{x}}/\partial \hat{s}|_i$ and $\partial \tilde{\boldsymbol{x}}_i/\partial \hat{s}$ which is explained in Figure 4 and can be derived by means of Lemma 6.

These considerations lead to the following Lemma which forms the basis for the general result of calculating the first-order sensitivity $\partial \tilde{\boldsymbol{x}}_i/\partial \hat{s}$ derived in Theorem 8.

Lemma 7. *Two consecutive first-order state sensitivities* $\dfrac{\partial \tilde{\boldsymbol{x}}_i}{\partial \hat{s}}$ *and* $\dfrac{\partial \tilde{\boldsymbol{x}}_{i+1}}{\partial \hat{s}}$ *satisfy for* $i \in \overline{\mathcal{I}}$

$$\frac{\partial \tilde{\boldsymbol{x}}_{i+1}}{\partial \hat{s}} = \boldsymbol{X}_{i+1}^{i} \frac{\partial \tilde{\boldsymbol{x}}_i}{\partial \hat{s}} + \bar{t}_{i+1} \left(\boldsymbol{f}_{i+1} + \frac{\hat{\boldsymbol{w}}_i}{\hat{s}\, T_i} \right) - \bar{t}_i \boldsymbol{X}_{i+1}^{i} \left(\boldsymbol{f}_i + \frac{\hat{\boldsymbol{w}}_i}{\hat{s}\, T_i} \right) - \frac{1}{\hat{s}}\, {}^{i}\boldsymbol{Z}_{i+1}^{i} \hat{\boldsymbol{w}}_i.$$

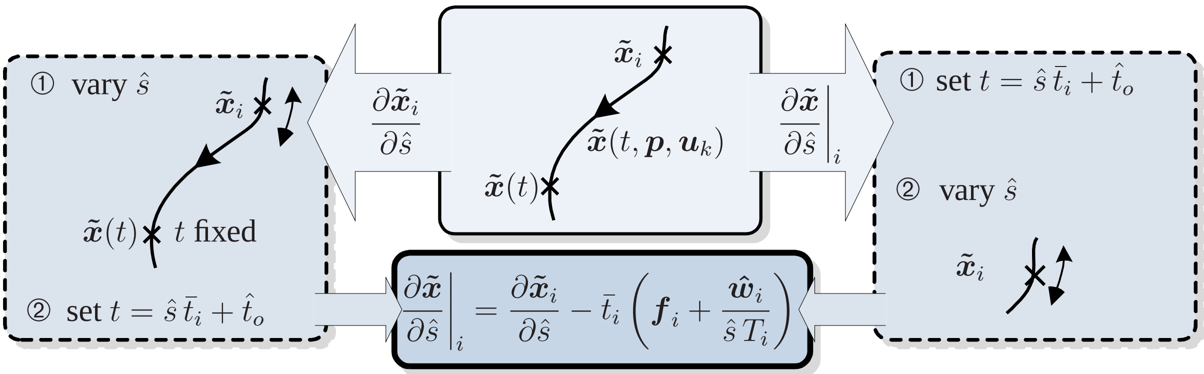

Figure 4: Relation between $\left.\frac{\partial \tilde{\boldsymbol{x}}}{\partial \hat{s}}\right|_i$ and $\frac{\partial \tilde{\boldsymbol{x}}_i}{\partial \hat{s}}$.

Proof. Differentiating (14) with respect to $\hat{s}$ by means of Lemma 6 yields

$$\begin{aligned}
\frac{\partial \tilde{\boldsymbol{x}}_{i+1}}{\partial \hat{s}} &= \frac{\partial}{\partial \hat{s}} \left(\tilde{\boldsymbol{x}}_i + \int_{\hat{s}\,\bar{t}_i + \hat{t}_o}^{\hat{s}\,\bar{t}_{i+1} + \hat{t}_o} (\boldsymbol{f} + \boldsymbol{\theta})\, dt \right) \\
&= \frac{\partial \tilde{\boldsymbol{x}}_i}{\partial \hat{s}} + \int_{\hat{s}\,\bar{t}_i + \hat{t}_o}^{\hat{s}\,\bar{t}_{i+1} + \hat{t}_o} \underbrace{\frac{\partial}{\partial \hat{s}} (\boldsymbol{f} + \boldsymbol{\theta})}_{=:\boldsymbol{a}}\, dt + \bar{t}_{i+1} \left(\boldsymbol{f}_{i+1} + \frac{\hat{\boldsymbol{w}}_i}{\hat{s}\,T_i} \right) - \bar{t}_i \left(\boldsymbol{f}_i + \frac{\hat{\boldsymbol{w}}_i}{\hat{s}\,T_i} \right).
\end{aligned} \tag{15a}$$

Performing the derivation in the intermediate term $\boldsymbol{a}$ by repetitive application of the chain-rule and using the result depicted in Figure 4 leads to

$$\begin{aligned}
\boldsymbol{a} &= \frac{\partial}{\partial \hat{s}} (\boldsymbol{f} + \boldsymbol{\theta}) \\
&= \frac{\partial \boldsymbol{f}}{\partial \tilde{\boldsymbol{x}}} \left(\frac{\partial \tilde{\boldsymbol{x}}}{\partial \hat{s}} + \frac{\partial \tilde{\boldsymbol{x}}}{\partial \boldsymbol{\theta}} \frac{\partial \boldsymbol{\theta}}{\partial \hat{s}} \right) + \frac{\partial \boldsymbol{\theta}}{\partial \hat{s}} \\
&= \frac{\partial \boldsymbol{f}}{\partial \tilde{\boldsymbol{x}}} \frac{\partial \tilde{\boldsymbol{x}}}{\partial\, \tilde{\boldsymbol{x}}|_i} \left.\frac{\partial \tilde{\boldsymbol{x}}}{\partial \hat{s}}\right|_i - \left(\frac{\partial \boldsymbol{f}}{\partial \tilde{\boldsymbol{x}}} \frac{\partial \tilde{\boldsymbol{x}}}{\partial \boldsymbol{\theta}} + \boldsymbol{I} \right) \frac{\hat{\boldsymbol{w}}_i}{\hat{s}^2\,T_i} \\
&= \frac{\partial \boldsymbol{f}}{\partial \tilde{\boldsymbol{x}}} \frac{\partial \tilde{\boldsymbol{x}}}{\partial\, \tilde{\boldsymbol{x}}|_i} \left[\frac{\partial \tilde{\boldsymbol{x}}_i}{\partial \hat{s}} - \bar{t}_i \left(\boldsymbol{f}_i + \frac{\hat{\boldsymbol{w}}_i}{\hat{s}\,T_i} \right) \right] - \left(\frac{\partial \boldsymbol{f}}{\partial \tilde{\boldsymbol{x}}} \frac{\partial \tilde{\boldsymbol{x}}}{\partial \boldsymbol{\theta}} + \boldsymbol{I} \right) \frac{\hat{\boldsymbol{w}}_i}{\hat{s}^2\,T_i}.
\end{aligned} \tag{15b}$$

Noting that $\tilde{\boldsymbol{x}}|_i = \tilde{\boldsymbol{x}}_i$, the desired derivative is obtained as

$$\begin{aligned}
\frac{\partial \tilde{\boldsymbol{x}}_{i+1}}{\partial \hat{s}} = \frac{\partial \tilde{\boldsymbol{x}}_i}{\partial \hat{s}} &+ \underbrace{\int_{\hat{s}\,\bar{t}_i + \hat{t}_o}^{\hat{s}\,\bar{t}_{i+1} + \hat{t}_o} \frac{\partial \boldsymbol{f}}{\partial \tilde{\boldsymbol{x}}} \frac{\partial \tilde{\boldsymbol{x}}}{\partial \tilde{\boldsymbol{x}}_i} dt}_{=:\boldsymbol{B}} \left[\frac{\partial \tilde{\boldsymbol{x}}_i}{\partial \hat{s}} - \bar{t}_i \left(\boldsymbol{f}_i + \frac{\hat{\boldsymbol{w}}_i}{\hat{s}\,T_i} \right) \right] - \underbrace{\int_{\hat{s}\,\bar{t}_i + \hat{t}_o}^{\hat{s}\,\bar{t}_{i+1} + \hat{t}_o} \frac{\partial \boldsymbol{f}}{\partial \tilde{\boldsymbol{x}}} \frac{\partial \tilde{\boldsymbol{x}}}{\partial \boldsymbol{\theta}} + \boldsymbol{I} dt}_{=:\boldsymbol{C}} \frac{\hat{\boldsymbol{w}}_i}{\hat{s}^2\,T_i} \\
&+ \bar{t}_{i+1} \left(\boldsymbol{f}_{i+1} + \frac{\hat{\boldsymbol{w}}_i}{\hat{s}\,T_i} \right) - \bar{t}_i \left(\boldsymbol{f}_i + \frac{\hat{\boldsymbol{w}}_i}{\hat{s}\,T_i} \right),
\end{aligned} \tag{15c}$$

where the terms abbreviated with $\boldsymbol{B}$ and $\boldsymbol{C}$ have to be determined. The term $\boldsymbol{B}$ is equivalent to the integral over the right-hand side of the ODE (10a) for $t \in \mathbb{T}_i$

$$\begin{aligned}
\boldsymbol{B} &= \int_{\hat{s}\,\bar{t}_i + \hat{t}_o}^{\hat{s}\,\bar{t}_{i+1} + \hat{t}_o} \frac{\partial \boldsymbol{f}}{\partial \tilde{\boldsymbol{x}}} \frac{\partial \tilde{\boldsymbol{x}}}{\partial \tilde{\boldsymbol{x}}_i}\, dt \\
&= \boldsymbol{X}^i_{i+1} - \boldsymbol{I}.
\end{aligned} \tag{15d}$$

For the determination of $\boldsymbol{C}$, the ODE derived by differentiating (7) with respect to $\boldsymbol{\theta}$ for $t \in \mathbb{T}_i$

$$\frac{\partial \dot{\tilde{\boldsymbol{x}}}}{\partial \boldsymbol{\theta}} = \frac{\partial \boldsymbol{f}}{\partial \tilde{\boldsymbol{x}}} \frac{\partial \tilde{\boldsymbol{x}}}{\partial \boldsymbol{\theta}} + \boldsymbol{I}, \quad \frac{\partial \tilde{\boldsymbol{x}}}{\partial \boldsymbol{\theta}}(\hat{s}\,\bar{t}_i + \hat{t}_o) = \boldsymbol{0} \tag{15e}$$

is compared to the first-order sensitivity matrix differential equation (11a) multiplied by $\hat{s}\,T_j$ on the same set $\mathbb{T}_i$

$$\hat{s}\,T_i\,{}^i\dot{\boldsymbol{Z}} = \frac{\partial \boldsymbol{f}}{\partial \tilde{\boldsymbol{x}}} \hat{s}\,T_i\,{}^i\boldsymbol{Z} + \boldsymbol{I}, \quad \hat{s}\,T_i\,{}^i\boldsymbol{Z}(\hat{s}\,\bar{t}_i + \hat{t}_o) = \boldsymbol{0}. \tag{15f}$$

Both ODEs are identical if $\partial\tilde{\boldsymbol{x}}/\partial\boldsymbol{\theta} = \hat{s}\,T_i\,{}^i\boldsymbol{Z}$ is chosen. This fact is exploited to calculate $\boldsymbol{C}$ which is the integral over the right-hand side of these ODEs over $\mathbb{T}_i$

$$\begin{aligned}
\boldsymbol{C} &= \int_{\hat{s}\,\bar{t}_i + \hat{t}_o}^{\hat{s}\,\bar{t}_{i+1} + \hat{t}_o} \frac{\partial \boldsymbol{f}}{\partial \tilde{\boldsymbol{x}}} \frac{\partial \tilde{\boldsymbol{x}}}{\partial \boldsymbol{\theta}} + \boldsymbol{I}\,dt \\
&= \int_{\hat{s}\,\bar{t}_i + \hat{t}_o}^{\hat{s}\,\bar{t}_{i+1} + \hat{t}_o} \frac{\partial \boldsymbol{f}}{\partial \tilde{\boldsymbol{x}}} \hat{s}\,T_i\,{}^i\boldsymbol{Z} + \boldsymbol{I}\,dt \\
&= \hat{s}\,T_i\,{}^i\boldsymbol{Z}_{i+1}^i - \hat{s}\,T_i\,{}^i\boldsymbol{Z}_i^i \\
&= \hat{s}\,T_i\,{}^i\boldsymbol{Z}_{i+1}^i.
\end{aligned} \tag{15g}$$

Substituting $\boldsymbol{B}$ and $\boldsymbol{C}$ in (15c) concludes the proof. □

Theorem 8. *The first-order sensitivities* $\dfrac{\partial \tilde{\boldsymbol{x}}_i}{\partial \hat{s}}$ *are for* $i \in \underline{\mathcal{I}}$

$$\frac{\partial \tilde{\boldsymbol{x}}_i}{\partial \hat{s}} = \bar{t}_i \boldsymbol{f}_i - \bar{t}_{k-N} \boldsymbol{X}_i^{k-N} \boldsymbol{f}_{k-N} + \sum_{j=k-N}^{i-1} \left(\bar{t}_{j+1} \boldsymbol{X}_i^{j+1} - \bar{t}_j \boldsymbol{X}_i^j\right) \frac{\hat{\boldsymbol{w}}_j}{\hat{s}\,T_j} - \frac{1}{\hat{s}} \sum_{j=k-N}^{i-1} {}^j\boldsymbol{Z}_i^{k-N} \hat{\boldsymbol{w}}_j.$$

Proof. The proof is done by induction. Let $P(i)$ be the statement in the above Theorem. For $i = k - N$ Lemma 7 leads together with $\partial\tilde{\boldsymbol{x}}_{k-N}/\partial\hat{s} = \boldsymbol{0}$ to

$$\begin{aligned}
\frac{\partial \tilde{\boldsymbol{x}}_{k-N+1}}{\partial \hat{s}} &= \boldsymbol{X}_{k-N+1}^{k-N} \boldsymbol{0} + \bar{t}_{k-N+1} \left(\boldsymbol{f}_{k-N+1} + \frac{\hat{\boldsymbol{w}}_{k-N}}{\hat{s}\,T_{k-N}}\right) - \bar{t}_{k-N} \boldsymbol{X}_{k-N+1}^{k-N} \left(\boldsymbol{f}_{k-N} + \frac{\hat{\boldsymbol{w}}_{k-N}}{\hat{s}\,T_{k-N}}\right) \\
&\quad - \frac{1}{\hat{s}}\,{}^{k-N}\boldsymbol{Z}_{k-N+1}^{k-N} \hat{\boldsymbol{w}}_{k-N},
\end{aligned}$$

which is equivalent to $P(k-N+1)$, i. e. $P(k-N+1)$ is true. This starts the induction. Now, assume that $P(i)$ is true for some $i \in \underline{\mathcal{I}}$. Substituting the assumption in Lemma 7 and applying the properties stated in Lemma 4 yield to

$$\begin{aligned}
\frac{\partial \tilde{\boldsymbol{x}}_{i+1}}{\partial \hat{s}} &= \bar{t}_i \boldsymbol{X}_{i+1}^i \boldsymbol{f}_i - \bar{t}_{k-N} \boldsymbol{X}_{i+1}^i \boldsymbol{X}_i^{k-N} \boldsymbol{f}_{k-N} + \sum_{j=k-N}^{i-1} \left(\bar{t}_{j+1} \boldsymbol{X}_{i+1}^i \boldsymbol{X}_i^{j+1} - \bar{t}_j \boldsymbol{X}_{i+1}^i \boldsymbol{X}_i^j\right) \frac{\hat{\boldsymbol{w}}_j}{\hat{s}\,T_j} \\
&\quad - \frac{1}{\hat{s}} \sum_{j=k-N}^{i-1} \boldsymbol{X}_{i+1}^i\,{}^j\boldsymbol{Z}_i^{k-N} \hat{\boldsymbol{w}}_j + \bar{t}_{i+1} \left(\boldsymbol{f}_{i+1} + \frac{\hat{\boldsymbol{w}}_i}{\hat{s}\,T_i}\right) - \bar{t}_i \boldsymbol{X}_{i+1}^i \left(\boldsymbol{f}_i + \frac{\hat{\boldsymbol{w}}_i}{\hat{s}\,T_i}\right) - \frac{1}{\hat{s}}\,{}^i\boldsymbol{Z}_{i+1}^i \hat{\boldsymbol{w}}_i \\
&= \bar{t}_{i+1} \boldsymbol{f}_{i+1} - \bar{t}_{k-N} \boldsymbol{X}_{i+1}^{k-N} \boldsymbol{f}_{k-N} + \sum_{j=k-N}^{i} \left(\bar{t}_{j+1} \boldsymbol{X}_{i+1}^{j+1} - \bar{t}_j \boldsymbol{X}_{i+1}^j\right) \frac{\hat{\boldsymbol{w}}_j}{\hat{s}\,T_j} - \frac{1}{\hat{s}} \sum_{j=k-N}^{i} {}^j\boldsymbol{Z}_{i+1}^{k-N} \hat{\boldsymbol{w}}_j,
\end{aligned}$$

which is $P(i+1)$, completing the inductive step. Thus, by the principle of induction, $P(i)$ is true for all $i \in \underline{\mathcal{I}}$. □

It is straightforward to compute the first-order state sensitivities $\partial \tilde{\boldsymbol{x}}_i / \partial \hat{t}_o$. In fact, the same method as for the calculation of $\partial \tilde{\boldsymbol{x}}_i / \partial \hat{s}$ can be applied with two differences: first, the corresponding derivative of the bounds of the integral (14) and the corresponding derivative of $\boldsymbol{\theta}$ change from $\bar{t}_i$ to 1 and from $-\frac{1}{s}\boldsymbol{\theta}$ to $\mathbf{0}$, respectively. These changes lead to the Lemma 9 and Theorem 10 which are stated in the following. The corresponding proofs are omitted due space limitation.

Lemma 9. *Two consecutive first-order state sensitivities* $\dfrac{\partial \tilde{\boldsymbol{x}}_i}{\partial \hat{t}_o}$ *and* $\dfrac{\partial \tilde{\boldsymbol{x}}_{i+1}}{\partial \hat{t}_o}$ *satisfy for* $i \in \overline{\mathcal{I}}$

$$\frac{\partial \tilde{\boldsymbol{x}}_{i+1}}{\partial \hat{t}_o} = \boldsymbol{X}_{i+1}^{i} \frac{\partial \tilde{\boldsymbol{x}}_i}{\partial \hat{t}_o} + \left(\boldsymbol{f}_{i+1} + \frac{\hat{\boldsymbol{w}}_i}{\hat{s}\, T_i} \right) - \boldsymbol{X}_{i+1}^{i} \left(\boldsymbol{f}_i + \frac{\hat{\boldsymbol{w}}_i}{\hat{s}\, T_i} \right).$$

Theorem 10. *The first-order sensitivities* $\dfrac{\partial \tilde{\boldsymbol{x}}_i}{\partial \hat{t}_o}$ *are for* $i \in \underline{\mathcal{I}}$

$$\frac{\partial \tilde{\boldsymbol{x}}_i}{\partial \hat{t}_o} = \boldsymbol{f}_i - \boldsymbol{X}_i^{k-N} \boldsymbol{f}_{k-N} + \sum_{j=k-N}^{i-1} \left(\boldsymbol{X}_i^{j+1} - \boldsymbol{X}_i^{j} \right) \frac{\hat{\boldsymbol{w}}_j}{\hat{s}\, T_j}.$$

The overall approach for computing the first-order state sensitivities defined in (8) is stated in the following corollary which is a direct consequence of the Theorems 5, 8 and 10 combined with Lemma 4.

Corollary 1. *The first-order sensitivities in* (8) *are for* $i \in \underline{\mathcal{I}}$

$$\begin{aligned}
\frac{\partial \tilde{\boldsymbol{x}}_i}{\partial \hat{s}} &= \bar{t}_i \boldsymbol{f}_i - \bar{t}_{k-N} \boldsymbol{X}_i^{i-1} \boldsymbol{X}_{i-1}^{i-2} \dots \boldsymbol{X}_{k-N+1}^{k-N} \boldsymbol{f}_{k-N} - \frac{1}{\hat{s}} \sum_{j=k-N}^{i-1} \boldsymbol{X}_i^{i-1} \boldsymbol{X}_{i-1}^{i-2} \dots \boldsymbol{X}_{j+2}^{j+1}{}^{j}\boldsymbol{Z}_{j+1}^{j} \hat{\boldsymbol{w}}_j \\
&\quad + \sum_{j=k-N}^{i-1} \boldsymbol{X}_i^{i-1} \boldsymbol{X}_{i-1}^{i-2} \dots \boldsymbol{X}_{j+2}^{j+1} \left(\bar{t}_{j+1} \boldsymbol{I} - \bar{t}_j \boldsymbol{X}_{j+1}^{j} \right) \frac{\hat{\boldsymbol{w}}_j}{\hat{s}\, T_j} \\
\frac{\partial \tilde{\boldsymbol{x}}_i}{\partial \hat{t}_o} &= \boldsymbol{f}_i - \boldsymbol{X}_i^{i-1} \boldsymbol{X}_{i-1}^{i-2} \dots \boldsymbol{X}_{k-N+1}^{k-N} \boldsymbol{f}_{k-N} + \sum_{j=k-N}^{i-1} \boldsymbol{X}_i^{i-1} \boldsymbol{X}_{i-1}^{i-2} \dots \boldsymbol{X}_{j+2}^{j+1} \left(\boldsymbol{I} - \boldsymbol{X}_{j+1}^{j} \right) \frac{\hat{\boldsymbol{w}}_j}{\hat{s}\, T_j} \\
\frac{\partial \tilde{\boldsymbol{x}}_i}{\partial \hat{\boldsymbol{x}}_{k-N}} &= \boldsymbol{X}_i^{i-1} \boldsymbol{X}_{i-1}^{i-2} \dots \boldsymbol{X}_{k-N+1}^{k-N} \\
\frac{\partial \tilde{\boldsymbol{x}}_i}{\partial \hat{\boldsymbol{w}}_j} &= \begin{cases} \mathbf{0}, & i < j+1 \\ \boldsymbol{X}_i^{i-1} \boldsymbol{X}_{i-1}^{i-2} \dots \boldsymbol{X}_{j+2}^{j+1}{}^{j}\boldsymbol{Z}_{j+1}^{j}, & i \geq j+1 \end{cases}, \quad j \in \overline{\mathcal{I}}
\end{aligned}$$

and for $i = k - N$ *all identical to* $\mathbf{0}$*, except* $\partial \tilde{\boldsymbol{x}}_{k-N} / \partial \hat{\boldsymbol{x}}_{k-N}$ *which is the identity matrix* $\boldsymbol{I}$*.*

Note that the ODEs which are necessary for the calculation of the first-order sensitivities $\partial \tilde{\boldsymbol{x}}_i / \partial \hat{\boldsymbol{x}}_{k-N}$ and $\partial \tilde{\boldsymbol{x}}_i / \partial \hat{\boldsymbol{w}}_j$ are sufficient for the calculation of the first-order sensitivities $\partial \tilde{\boldsymbol{x}}_i / \partial \hat{s}$ and $\partial \tilde{\boldsymbol{x}}_i / \partial \hat{t}_o$, i. e. no additional ODEs have to be solved and the advantages of Theorem 5 still hold. The advantages of calculating the gradient and the Hessian of the Lagrange function $\mathcal{L}$ via proposition 1 and corollary 1 compared to the finite difference method (5) are summarized below:

1) In contrast to the finite difference method, the calculation is exact.

2) The number of ODEs that have to be solved over $\mathbb{T}$ is reduced from $2n((N+1)n+2)$ to $2n^2+n$ and is independent of N.

3) The ODEs necessary for the calculation of the first-order sensitivities are independent of each other and can be solved in parallel on the sets $t \in \mathbb{T}_i$ instead on the set $\mathbb{T}$.

4) The computation of the gradient of $\mathcal{L}$ provides an often sufficient approximation of the Hessian of $\mathcal{L}$ for free and, if $\partial^2 \boldsymbol{f}/\partial \hat{\boldsymbol{x}}^2 = \mathbf{0}$ holds, in fact an exact Hessian.

6 Conclusion

In this paper, the relation between first-order state sensitivities and the NCS-MHE are considered. It is shown, that the gradient and the Hessian of the Lagrange function of the NLP in the update step can be efficiently calculated based on first-order state sensitivities. A new method is presented for the efficient calculation of these sensitivities by exploiting the common underlying structure of the NLP. Future work will involve experiments at a test-rig to validate the results presented in this paper.

Acknowledgements

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

References

[1] J. P. Hespanha, P. Naghshtabrizi, and Y. Xu. A survey of recent results in networked control systems. In *Proc. IEEE*, volume 95, pages 138–162, 2007.

[2] T. Kailath. *Linear Systems*. Prentice Hall, 1980.

[3] P. Philipp. Structure exploiting derivative computation for moving horizon estimation. In *Accepted and to be presented at the American Control Conference*, 2011.

[4] P. Philipp and B. Lohmann. Moving horizon estimation for nonlinear networked control systems with unsynchronized timescales. In *Accepted and to be presented at the IFAC World Congress*, 2011.

[5] S. Zampieri. Trends in networked control systems. In *Proc. of the 17th IFAC World Congress*, pages 2886–2894, 2008.