Problem formulation and efficient numerical methods for real-time estimation based on DAE process models

Seminar: Numerical methods for nonlinear parameter estimation

Ihno Schrot

July 28, 2018

Contents

List of Variables

Symbol	Description	Element of/size
C_L	Arrival cost of time window $[\tau_L, \tau_k]$; the superscripts "$*$" and "$\prime$" denote variants	$\mathbb{R}$
$\boldsymbol{D}_k$	Online data in the MHE at time τ_k (see equation (27))	No actual mathematical dimension
$\Delta \boldsymbol{r}_k^i$	i-th increment for $\boldsymbol{r}_k$ in the Gauß-Newton method for the MHE	see $\boldsymbol{r}_k$
$\boldsymbol{F}$	Iteration function for $\boldsymbol{x}_j$ usually obtained from numerical integration	$\mathbb{R}^{n_x} \times \mathbb{R}^{n_z} \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_p} \to \mathbb{R}^{n_x}$
$\boldsymbol{f}$	Function in the DAE system	$\mathbb{R}^{n_x} \times \mathbb{R}^{n_z} \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_p} \to \mathbb{R}^{n_x}$
$\boldsymbol{G}$	Collected equality constraints in terms of $\boldsymbol{r}_k$ and $\boldsymbol{D}_k$ in the MHE	$(M-1)n_x + Mn_z$ equations
$\boldsymbol{g}$	Function in the DAE system	$\mathbb{R}^{n_x} \times \mathbb{R}^{n_z} \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_p} \to \mathbb{R}^{n_z}$
$\boldsymbol{H}$	Collected inequality constraints in terms of $\boldsymbol{r}_k$ and $\boldsymbol{D}_k$ in the MHE	$2M(2n_x + n_z) + 2n_p$ inequalities
$\boldsymbol{h}$	Ideal measurement function (collecting all devices in one vector)	$\mathbb{R}^{n_x} \times \mathbb{R}^{n_z} \times \mathbb{R}^{n_p} \to \mathbb{R}^{n_h}$
H_p	Sensitivity matrix of the ideal measurement function with states at time τ_{L+1} with respect to the parameters $\boldsymbol{p}$	$\mathbb{R}^{n_h \times n_p}$
H_x	Sensitivity matrix of the ideal measurement function with states at time τ_{L+1} with respect to the initial values $\boldsymbol{x}(\tau_L)$	$\mathbb{R}^{n_h \times n_x}$
J	General cost function for the optimal control problem	$\mathbb{R}^{n_x} \times \mathbb{R}^{n_z} \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_p} \to \mathbb{R}$
k	Index of last measurement in time window of MHE	$\mathbb{R}$
L	$:= k - M + 1$; Index of first measurement in time window of MHE	$\mathbb{R}$

Symbol	Description	Element of/size
M	Number of measurements in time window of MHE	$\mathbb{R}$
M_{QR}	Matrix which is decomposed with a QR-decomposition. See equation (12)	$\mathbb{R}^{(2n_x+2n_p+n_h)\times(2n_x+2n_p)}$
$\boldsymbol{p}$	System parameters	$\mathbb{R}^{n_p}$
P_L	Weighting matrix for the arrival cost C_L	$\mathbb{R}^{(n_x+n_p)\times(n_x+n_p)}$
$\bar{\boldsymbol{p}}_L$	Parameter for the arrival cost C_L	$\mathbb{R}^{n_p}$
$\hat{\boldsymbol{p}}$	Estimation of the parameters	$\mathbb{R}^{n_p}$
$\boldsymbol{\Phi}$	Basis functions for the controls in the multiple shooting approach	$\mathbb{R} \to \mathbb{R}^{n_u}$
Q	Orthogonal matrix of QR-decomposition of M_{QR}	$\mathbb{R}^{(2n_x+2n_p+n_h)\times(2n_x+2n_p+n_h)}$
$\boldsymbol{q}$	Optimization variables for the discretised controls in the multiple shooting method	$\mathbb{R}^{n_u}$
$\mathcal{R}_1,\ \mathcal{R}_{12},\ \mathcal{R}_2$	Matrices occuring in the QR-decomposition of M_{QR} (see equation (13))	$\mathbb{R}^{(n_x+n_p)\times(n_x+n_p)}$
$\rho_1,\ \rho_2,\ \rho_3$	Needed in the computation of the arrival cost; see equation (14) for definition	$\mathbb{R}^{(n_x+n_p)}$ for $\rho_1,\ \rho_2$, $\mathbb{R}^{n_h}$ for ρ_3
$\boldsymbol{r}_k$	Collected optimization variables in the MHE at time τ_k (see equation (26))	$\mathbb{R}^{(M)*(n_x+n_z)+(M-1)*n_x+n_p}$
$\boldsymbol{r}_k^i$	i-th iterate for $\boldsymbol{r}_k$ in the Gauß-Newton method for the MHE	see $\boldsymbol{r}_k$
$\boldsymbol{s}^x,\ \boldsymbol{s}^z$	Initial states at the shooting nodes in the multiple shooting method	$\mathbb{R}^{n_x},\ \mathbb{R}^{n_z}$
T	End time (in the offline case)	$\mathbb{R}$
t	Time	$\mathbb{R}$
t_0	Starting time (in the offline case)	$\mathbb{R}$
t_{M}	Length of time window of MHE	$\mathbb{R}$
τ_j	j-th sampling time/shooting node of the multiple shooting method	$\mathbb{R}$
$\boldsymbol{u}$	Controls	$\mathbb{R}^{n_u}$
V	Weighting matrix of the measurement noise. Usually square root of the inverse of the covariance matrix of $\boldsymbol{v}$	$\mathbb{R}^{n_h\times n_h}$

Symbol	Description	Element of/size
$\boldsymbol{v}$	Measurement noise	$\mathbb{R}^{n_h}$
$\boldsymbol{w}$	State noise	$\mathbb{R}^{n_x}$
$\boldsymbol{w}^p$	Parameter noise	$\mathbb{R}^{n_p}$
W	Weighting matrix of the state noise. Usually square root of the inverse of the covariance matrix of the state noise $\boldsymbol{w}$	$\mathbb{R}^{n_x \times n_x}$
$\bar{W}$	Weighting matrix of the combined state and parameter noise. Usually square root of the inverse of the combined covariance matrix of $\boldsymbol{w}$ and $\boldsymbol{w}^p$	$\mathbb{R}^{(n_x+n_p)\times(n_x+n_p)}$
$\boldsymbol{x}$	Differential states	$\mathbb{R}^{n_x}$
$\boldsymbol{x}_0$	Initial values for differential states	$\mathbb{R}^{n_x}$
X_p	Sensitivity matrix of the states at time τ_{L+1} with respect to the parameters $\boldsymbol{p}$	$\mathbb{R}^{n_x \times n_p}$
X_x	Sensitivity matrix of the states at time τ_{L+1} with respect to the initial values $\boldsymbol{x}(\tau_L)$	$\mathbb{R}^{n_x \times n_x}$
$\bar{\boldsymbol{x}}_L$	Parameter for the arrival cost C_L	$\mathbb{R}^{n_x}$
$\hat{\boldsymbol{x}}$	Estimation of the differential states	$\mathbb{R}^{n_x}$
$\boldsymbol{\xi}$	Collected optimization variables in the multiple shooting method (see equation (24))	bla
$\boldsymbol{y}_j$	Output of measurement devices	$\mathbb{R}^{n_h}$
$\boldsymbol{z}$	Algebraic states	$\mathbb{R}^{n_z}$
$\hat{\boldsymbol{z}}$	Estimation of the algebraic states	$\mathbb{R}^{n_z}$

Note: A subscript either indicates the interval index for which the variable holds or that the variables was taken at time τ_j unless specified otherwise.

1 Introduction

Many processes relevant in practical application are described using differential-algebraic equations (DAEs), which in general cannot be solved analytically. Still, estimating state variables and possibly system parameters of those processes is a key problem, especially in control engineering. Given measurements of some quantities of the process this problem leads to the so called state and parameter estimation problem. Whereas powerful methods, such as the multiple shooting method Bock and Plitt [1984], exist even for nonlinear problems in the classical, i.e. off-line, case, the online case involves more difficulties. Although linear systems can be tackled quite well, e.g. with Kalman Filters Gelb [1974], the above is especially true for nonlinear systems. Actually, modifications of Kalman Filters, e.g. extended Kalman filtering (EKF) or unscented Kalman filtering (UKF), and several other approaches, such as particle filtering (PF) or different variants of observers, exist, but many of them suffer from the curse of dimensionality Daum [2005], Rawlings and Bakshi [2006]. Thus usage of those methods for real-time estimation, where one likes to achieve to gain knowledge of a process without a severe delay, can be problematic. In Moving Horizon Estimation (MHE) one circumvents the curse of dimensionality by only considering measurements within a window of fixed size, thus fixing the number of measurements.

Kühl et al. presented a real-time algorithm based on the Moving Horizon approach and the multiple shooting method, which performs well in several test cases. In the following we especially focus on this specific algorithm due to its good performance. Before we actually look at the real-time algorithm for MHE, we start off with the precise problem formulation, followed by a brief description of the multiple shooting method, which lays the foundation of presented real-time algorithm. We then conclude with a short discussion of the algorithm.

2 Problem formulation

To develop a better understanding of the more complex formulation of the online state and parameter estimation problem we first formulate the classical problem and proceeding from there derive the online version.

2.1 Classical case

First of all we introduce a model for the dynamics of the system of interest. More precisely we use differential-algebraic equations

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{z}(t), \boldsymbol{u}(t), \boldsymbol{p}) + \boldsymbol{w}(t), \qquad t \in [t_0, T], \tag{1a}$$

$$\boldsymbol{x}_0 = \boldsymbol{x}(t_0), \tag{1b}$$

$$\boldsymbol{0} = \boldsymbol{g}(\boldsymbol{x}(t), \boldsymbol{z}(t), \boldsymbol{u}(t), \boldsymbol{p}), \qquad t \in [t_0, T], \tag{1c}$$

to model the relation between the differential states $\boldsymbol{x} \in \mathbb{R}^{n_x}$, the algebraic states $\boldsymbol{z} \in \mathbb{R}^{n_z}$, the controls $\boldsymbol{u} \in \mathbb{R}^{n_u}$ and the systems parameters $\boldsymbol{p} \in \mathbb{R}^{n_p}$. Here we also regard state noise by adding $\boldsymbol{w} \in \mathbb{R}^{n_x}$ to the function $\boldsymbol{f}$. Note that we assume $\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{x}}$ to be regular.

In the classical case we have a fixed set of outputs $\boldsymbol{y}_j := \boldsymbol{y}(\tau_j) \in \mathbb{R}^{n_h},\ j = 1, \ldots, m$, which we obtain by measuring specific quantities at times $\tau_j \in [t_0, T]$. So to say we observed the system behaviour for some time and *afterwards* want to estimate the states and parameters during that time. The corresponding ideal measurement function is denoted by $\boldsymbol{h}(\boldsymbol{x}(t), \boldsymbol{z}(t), \boldsymbol{p}) \in \mathbb{R}^{n_h}$. As the ideal measurements may be perturbed by some output noise $\boldsymbol{v}_j \in \mathbb{R}^{n_h}$, we have that

$$\boldsymbol{y}_j = \boldsymbol{h}(\boldsymbol{x}(\tau_j), \boldsymbol{z}(\tau_j), \boldsymbol{p}) + \boldsymbol{v}_j,\ \ j = 1, \ldots, m. \tag{2}$$

Naturally one wishes to minimize the deviation between the outputs and the ideal measurement values given by the function $\boldsymbol{h}$ with estimated $\boldsymbol{x},\ \boldsymbol{z},\ \boldsymbol{p}$. In contrast to the measurement noise $\boldsymbol{v}$ we have to include the state noise $\boldsymbol{w}$ into the optimization variables as otherwise equation (1a) would not be well-defined. We introduced the notation $||\boldsymbol{r}||_A := \boldsymbol{r}^T A^T A \boldsymbol{r}$ for a vector $\boldsymbol{r} \in \mathbb{R}^n$ and a matrix $A \in \mathbb{R}^{n \times n}$ to obtain a clearer presentation. Using the Least Squares approach and minimizing the size of $\boldsymbol{w}$ with weighting matrices V_j and W we thus obtain the optimization problem

$$\begin{aligned}
\min_{\boldsymbol{x},\ \boldsymbol{z},\ \boldsymbol{p},\ \boldsymbol{w}} \quad & \sum_{j=1}^{m} ||\boldsymbol{y}_j - \boldsymbol{h}(\boldsymbol{x}(\tau_j), \boldsymbol{z}(\tau_j), \boldsymbol{p})||_{V_j}^2 + \int_{t_0}^{T} ||\boldsymbol{w}(t)||_W^2 \ \mathrm{d}t && \text{(3a)}\\
\text{s.t.} \quad & \dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{z}(t), \boldsymbol{u}(t), \boldsymbol{p}) + \boldsymbol{w}(t), \qquad t \in [t_0, T], && \text{(3b)}\\
& \boldsymbol{x}_0 = \boldsymbol{x}(t_0), && \text{(3c)}\\
& \boldsymbol{0} = \boldsymbol{g}(\boldsymbol{x}(t), \boldsymbol{z}(t), \boldsymbol{u}(t), \boldsymbol{p}), \qquad t \in [t_0, T], && \text{(3d)}\\
& \boldsymbol{y}_j = \boldsymbol{h}(\boldsymbol{x}(\tau_j), \boldsymbol{z}(\tau_j), \boldsymbol{p}) + \boldsymbol{v}_j, \qquad j = 1, \ldots, m. && \text{(3e)}
\end{aligned}$$

The weighting matrices are determined by statistical arguments and usually chosen to be the square root of the inverse of the covariance matrix of the respective noise.

2.2 Moving Horizon formulation

In the online case continuously new outputs $\boldsymbol{y}_j$ become known and we wish to estimate the states and parameters while the process is still running (actually we finally hope for a real-time estimation). In the Moving Horizon approach we, at time $\hat{t}$, only consider a time window $[\hat{t} - t_{\mathrm{M}}, \hat{t}]$ and those outputs $\boldsymbol{y}_j,\ j = L := k - M + 1, \ldots, k$ which lay within this window. Still, we want to take advantage of actually having more data available from prior times. To that end we include a window dependent, so called, *arrival cost* C_L in the objective function, which so to say summarises the prior information. We modify the problem (3) to

$$\begin{aligned}
\min_{\boldsymbol{x},\ \boldsymbol{z},\ \boldsymbol{p},\ \boldsymbol{w}} \quad & C_L + \sum_{j=L}^{k} ||\boldsymbol{y}_j - \boldsymbol{h}(\boldsymbol{x}(\tau_j), \boldsymbol{z}(\tau_j), \boldsymbol{p})||^2_{V_j} + \int_{\hat{t}-t_{\mathrm{M}}}^{\hat{t}} ||\boldsymbol{w}(t)||^2_W \, \mathrm{d}t && \text{(4a)}\\
\text{s.t.} \quad & \dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{z}(t), \boldsymbol{u}(t), \boldsymbol{p}) + \boldsymbol{w}(t), \qquad t \in [\hat{t} - t_{\mathrm{M}}, \hat{t}], && \text{(4b)}\\
& \boldsymbol{x}_0 = \boldsymbol{x}(t_0), && \text{(4c)}\\
& \boldsymbol{0} = \boldsymbol{g}(\boldsymbol{x}(t), \boldsymbol{z}(t), \boldsymbol{u}(t), \boldsymbol{p}), \qquad t \in [\hat{t} - t_{\mathrm{M}}, \hat{t}], && \text{(4d)}\\
& \boldsymbol{y}_j = \boldsymbol{h}(\boldsymbol{x}(\tau_j), \boldsymbol{z}(\tau_j), \boldsymbol{p}) + \boldsymbol{v}_j, \qquad j = L, \ldots, k. && \text{(4e)}
\end{aligned}$$

As the samples are taken at discrete time steps and solving the optimization problem, as it is infinite dimensional, needs a discretisation anyway a time-discrete form of the optimization problem (4) is more convenient actually. The sampling times τ_j are a natural choice for the time grid. We will denote the states at time τ_j with $\boldsymbol{x}_j$ and $\boldsymbol{z}_j$. The same notation we apply to all the other variables, too. Note that we assume the parameters $\boldsymbol{p}$ to be constant and thus only use $\boldsymbol{p}$ instead of $\boldsymbol{p}_j$. We choose the window such that it starts at node τ_L and ends at the current time τ_k. We now replace equation (4b) by the iteration scheme

$$\boldsymbol{x}_{j+1} = \boldsymbol{F}(\boldsymbol{x}_j, \boldsymbol{z}_j, \boldsymbol{u}_j, \boldsymbol{p}) + \boldsymbol{w}_j, \ \ j = L, \ldots, k-1 \tag{5}$$

where $\boldsymbol{F}$ is an iteration function which is usually obtained from numerical integration. The arrival costs are commonly formulated as

$$C_L = \left|\left| \begin{array}{c} \boldsymbol{x}_L - \bar{\boldsymbol{x}}_L \\ \boldsymbol{p}_L - \bar{\boldsymbol{p}}_L \end{array} \right|\right|^2_{P_L}. \tag{6}$$

Choosing $\bar{\boldsymbol{x}}_L$, $\bar{\boldsymbol{p}}_L$ and P_L is a crucial step in the Moving Horizon Estimation and will be discussed in further detail below. Note that this form for the arrival costs is mainly motivated by its convexity. The theoretically "‘ideal"’ arrival costs can in general not be represented in that way, but still can be approximated in this form as we also will see later. Of course it is up to interpretation what is "‘ideal"’ regarding the arrival costs. For us that would be that the arrival costs allow us to find a problem formulation on given time window which is equivalent to a problem solved on the entire process time. We will explain this idea in further detail in the next subsection. Using this conventions we obtain the optimization problem

$$\min_{\boldsymbol{x}_j,\ \boldsymbol{z}_j,\ \boldsymbol{p},\ \boldsymbol{w}_j} \left(\left\| \begin{matrix} \boldsymbol{x}_L - \bar{\boldsymbol{x}}_L \\ \boldsymbol{p}_L - \bar{\boldsymbol{p}}_L \end{matrix} \right\|_{P_L}^2 + \sum_{j=L}^{k} ||\boldsymbol{y}_j - \boldsymbol{h}(\boldsymbol{x}(\tau_j), \boldsymbol{z}(\tau_j), \boldsymbol{p})||_{V_j}^2 + \sum_{j=L}^{k-1} ||\boldsymbol{w}_j||_W^2 \right) \tag{7a}$$

$$\text{s.t.} \quad \boldsymbol{x}_{j+1} = \boldsymbol{F}(\boldsymbol{x}_j, \boldsymbol{z}_j, \boldsymbol{u}_j, \boldsymbol{p}) + \boldsymbol{w}_j, \qquad j = L, \dots, k-1, \tag{7b}$$

$$\boldsymbol{0} = \boldsymbol{g}(\boldsymbol{x}_j, \boldsymbol{z}_j, \boldsymbol{u}_j, \boldsymbol{p}), \qquad j = L, \dots, k, \tag{7c}$$

$$\boldsymbol{x}_{j,\min} \le \boldsymbol{x}_j \le \boldsymbol{x}_{j,\max}, \qquad j = L, \dots, k, \tag{7d}$$

$$\boldsymbol{z}_{j,\min} \le \boldsymbol{z}_j \le \boldsymbol{z}_{j,\max}, \qquad j = L, \dots, k, \tag{7e}$$

$$\boldsymbol{w}_{j,\min} \le \boldsymbol{w}_j \le \boldsymbol{w}_{j,\max}, \qquad j = L, \dots, k, \tag{7f}$$

$$\boldsymbol{p}_{\min} \le \boldsymbol{p} \le \boldsymbol{p}_{\max}. \tag{7g}$$

2.3 Design of the arrival cost

As the focus of this essay lays on the conceptual understanding of the MHE real-time iteration we only discuss the most important aspects of the choice of the arrival cost at this point and refer for further details to Kühl et al. [2011]. The arrival cost should ideally be the sum of the costs of all prior time windows. Consider the optimization problem on an extended time window $[\tau_L, \tau_{k+1}]$. If we find a problem formulation such that the problem on the window $[\tau_L, \tau_{k+1}]$ is equivalent to the problem on the interval $[\tau_{L+1}, \tau_{k+1}]$ we would have found a formulation such that the problem on the interval $[\tau_{L+1}, \tau_{k+1}]$ would be equivalent to the problem on $[\tau_0, \tau_{k+1}]$. This is true as we could successively create equivalent problems on ever smaller time windows starting from the interval $[\tau_0, \tau_{k+1}]$ until we reach the problem on the interval $[\tau_{L+1}, \tau_{k+1}]$.

Looking at the two intervals we notice that we have to summarize the old arrival cost C_L, the output $\boldsymbol{y}_L$, and the state noise $\boldsymbol{w}_L$ in the new arrival cost C_{L+1}. At this point we step back from the assumption that the parameters $\boldsymbol{p}$ are constant in time and consider parameter noise $\boldsymbol{w}_j^p$, too. This means we introduce the additional condition

$$\boldsymbol{p}_{j+1} = \boldsymbol{p}_j + \boldsymbol{w}_j^p \tag{8}$$

allowing for slowly varying parameters. Using arguments of dynamic programming we can now conclude that the optimal arrival cost C_{L+1}^* must fulfil

$$C_{L+1}^* = \min_{\boldsymbol{x}_L,\ \boldsymbol{p}_L} \left(\left\| \begin{matrix} \boldsymbol{x}_L - \bar{\boldsymbol{x}}_L \\ \boldsymbol{p}_L - \bar{\boldsymbol{p}}_L \end{matrix} \right\|_{P_L}^2 + ||\boldsymbol{y}_L - \boldsymbol{h}(\boldsymbol{x}(\tau_L), \boldsymbol{p}_L)||_{V_L}^2 + \left\| \begin{matrix} \boldsymbol{w}_L \\ \boldsymbol{w}_L^p \end{matrix} \right\|_{\bar{W}_L}^2 \right) \tag{9a}$$

$$\text{s.t.} \quad \boldsymbol{w}_L = \boldsymbol{x}_{L+1} - \boldsymbol{F}(\boldsymbol{x}_L, \boldsymbol{u}_L, \boldsymbol{p}_L), \tag{9b}$$

$$\boldsymbol{w}_L^p = \boldsymbol{p}_{L+1} - \boldsymbol{p}_L. \tag{9c}$$

Hereby $\boldsymbol{x}_{L+1} := \boldsymbol{x}(\tau_{L+1})$ and $\boldsymbol{p}_{L+1}$ are obtained from solving the MHE problem on the interval $[\tau_{L+1}, \tau_{k+1}]$ and $\bar{W}$ is a suitable weighting matrix for both the state and the parameter noise. As $\boldsymbol{x}_{L+1}$ is known, $\boldsymbol{w}_L$ is not free any longer and thus not an optimization variable. The algebraic state $\boldsymbol{z}_L$ is fixed by the constraint (7c) and can be expressed using $\boldsymbol{x}_L$ and $\boldsymbol{p}_L$ and thus also is not an optimization variable any longer. As the functions $\boldsymbol{h}$ and $\boldsymbol{x}$ are nonlinear we would again have to apply an iterative procedure to obtain the optimal arrival cost. To avoid the associated computational costs we approximate the ideal arrival costs. Therefore we first linearize $\boldsymbol{h}(\boldsymbol{x}_L, \boldsymbol{z}_L, \boldsymbol{p}_L)$ and the solution of the DAE system on the interval $[\tau_l,\ \tau_{L+1}]$ at time τ_{L+1} with initial values $\boldsymbol{x}_L,\ \boldsymbol{p}_L$ denoted by $\boldsymbol{x}(\tau_{L+1}; \boldsymbol{x}_L, \boldsymbol{p}_L)$ using the estimations of $\boldsymbol{x}^*(\tau_L)$ and $\boldsymbol{p}^*$ from the last iteration. The approximation reads

$$
\begin{aligned}
\boldsymbol{x}(\tau_{L+1}; \boldsymbol{x}_l, \boldsymbol{p}_L) \approx \boldsymbol{x}^*(\tau_{L+1}; \boldsymbol{x}^*(\tau_L), \boldsymbol{p}^*) &+ \underbrace{\frac{d\boldsymbol{x}(\tau_{L+1}; \boldsymbol{x}^*(\tau_L), \boldsymbol{p}^*)}{d\boldsymbol{x}(\tau_L)}}_{:=X_x}(\boldsymbol{x}_L - \boldsymbol{x}^*(\tau_L)) \\
&+ \underbrace{\frac{d\boldsymbol{x}(\tau_{L+1}; \boldsymbol{x}^*(\tau_L), \boldsymbol{p}^*)}{d\boldsymbol{p}}}_{:=X_p}(\boldsymbol{p}_L - \boldsymbol{p}^*) \\
&:= \tilde{\boldsymbol{x}} + X_x \boldsymbol{x}_L + X_p \boldsymbol{p}_L
\end{aligned}
\tag{10}
$$

and analogously

$$
\boldsymbol{h}(\boldsymbol{x}_L, \boldsymbol{z}_L, \boldsymbol{p}_L) \approx \tilde{\boldsymbol{h}} + H_x \boldsymbol{x}_L + H_p \boldsymbol{p}_L. \tag{11}
$$

The matrices X_x and H_x are the sensitivity matrices of the states, and the measurement functions respectively, at time τ_{L+1} with respect to the initial values $x(\tau_L)$. Note that $\boldsymbol{F}$ is then replaced by the linear approximation. Then we replace $\boldsymbol{w}_L$ and $\boldsymbol{w}_L^p$ in the cost function using equation (9c) and the linearized form of equation (9b). We obtain the linear least-squares problem

$$
\min_{\boldsymbol{x}_L,\ \boldsymbol{p}_L} \left\| \begin{pmatrix} -P_L \begin{pmatrix} \bar{\boldsymbol{x}}_L \\ \bar{\boldsymbol{p}}_L \end{pmatrix} \\ V_L(\boldsymbol{y}_L - \tilde{\boldsymbol{h}}) \\ \bar{W}_L \begin{pmatrix} \tilde{\boldsymbol{x}} \\ 0 \end{pmatrix} \end{pmatrix} + \underbrace{\left(\begin{array}{c|c} P_L & 0 \\ -(V_L H_X \mid V_L H_p) & 0 \\ -\bar{W}_L \begin{pmatrix} X_X & X_p \\ 0 & \mathbb{I} \end{pmatrix} & \bar{W}_L \end{array} \right)}_{:=M_{\mathrm{QR}}} \cdot \begin{pmatrix} \boldsymbol{x}_L \\ \boldsymbol{p}_L \\ x_{L+1} \\ \boldsymbol{p}_{L+1} \end{pmatrix} \right\|_2^2 . \tag{12}
$$

Using the QR-decomposition

$$
M_{\mathrm{QR}} = Q \begin{pmatrix} \mathcal{R}_1 & \mathcal{R}_{12} \\ 0 & \mathcal{R}_2 \\ 0 & 0 \end{pmatrix} \tag{13}
$$

we can solve the problem (12) analytically. We introduce

$$\begin{pmatrix}\rho_1\\ \rho_2\\ \rho_3\end{pmatrix} := Q^T \left(\begin{array}{c} -P_L \begin{pmatrix} \bar{\boldsymbol{x}}_L \\ \bar{\boldsymbol{p}}_L \end{pmatrix} \\ V_L(\boldsymbol{y}_L - \tilde{\boldsymbol{h}}) \\ \bar{W}_L \begin{pmatrix} \tilde{\boldsymbol{x}} \\ 0 \end{pmatrix} \end{array} \right), \quad \rho_1, \rho_2 \in \mathbb{R}^{n_x+n_p}, \rho_3 \in \mathbb{R}^{n_h}. \tag{14}$$

In this notation the solution of (12) is given by

$$\begin{pmatrix}\boldsymbol{x}_L\\ \boldsymbol{p}_L\end{pmatrix} = -\mathcal{R}_1^{-1}\left(\rho_1 + \mathcal{R}_{12}\begin{pmatrix}\boldsymbol{x}_{L+1}\\ \boldsymbol{p}_{L+1}\end{pmatrix}\right), \tag{15}$$

with the optimal cost given as

$$C'(\boldsymbol{x}_{L+1}, \boldsymbol{p}_{L+1}) = \left\|\rho_3\right\|_2^2 + \left\|\rho_2 + \mathcal{R}_2\begin{pmatrix}\boldsymbol{x}_{L+1}\\ \boldsymbol{p}_{L+1}\end{pmatrix}\right\|_2^2. \tag{16}$$

Noting that $\left\|\rho_3\right\|_2^2$ is constant and thus does not change the system (7) and recalling the desired form for the arrival cost (6) the solution (16) motivates the choice

$$P_{L+1} := \mathcal{R}_2, \quad \begin{pmatrix}\bar{\boldsymbol{x}}_{L+1}\\ \bar{\boldsymbol{p}}_{L+1}\end{pmatrix} = -\mathcal{R}_2^{-1}\rho_2. \tag{17}$$

Note that as P_L and $\bar{W}_L$ can be chosen such that they have full rank, $\mathcal{R}_2$ is indeed regular. We naturally wish that the influence of the past information does not prevail in the cost function. To check whether our choice of the arrival cost complies with this wish, we look at

$$M_{\mathrm{QR}}^T M_{\mathrm{QR}} = \begin{pmatrix} *^1 & *^2 \\ *^3 & \bar{W}_L^T \bar{W}_L \end{pmatrix} \overset{(13)}{=} \begin{pmatrix} \mathcal{R}_1^T\mathcal{R}_1 & \mathcal{R}_1^T\mathcal{R}_{12} \\ \mathcal{R}_{12}^T\mathcal{R}_1 & \mathcal{R}_{12}^T\mathcal{R}_{12} + \mathcal{R}_2^T\mathcal{R}_2 \end{pmatrix} \tag{18}$$

to observe that for all $v \in \mathbb{R}^{n_x+n_p}$

$$\left\|v\right\|_{P_{L+1}}^2 = v^T P_{L+1}^T P_{L+1} v \overset{(17)}{=} v^T \mathcal{R}_2^T \mathcal{R}_2 v \le \underbrace{v^T \mathcal{R}_{12}^T \mathcal{R}_{12} v}_{\ge 0} + v^T \mathcal{R}_2^T \mathcal{R}_2 v \overset{(18)}{=} \left\|v\right\|_{\bar{W}_L}^2 \tag{19}$$

holds. Inequality (19) tells us that the user chosen weighting matrix $\bar{W}_L$ limits the weight of the arrival cost with the weighting matrix P_{L+1} and as a consequence prevents the influence of the arrival costs from growing too big.

3 Numerical solution

Having formulated the output and state noise MHE we discuss the real-time iteration (RTI) presented in Kühl et al. [2011]. As the RTI bases on the direct multiple shooting method developed by Bock and Plitt we first will briefly explain the multiple shooting method. We will see that the resulting problem structure resembles the one of the MHE. Finally we look at the modification of the multiple shooting method which are applied to obtain the RTI for MHE.

3.1 Direct multiple shooting

The multiple shooting method was originally developed for solving optimal control problems of the form

$$\begin{aligned}
&\min_{\boldsymbol{u}\in U,\ \boldsymbol{x},\ \boldsymbol{z},\ \boldsymbol{p}} && J(\boldsymbol{x}(t), \boldsymbol{z}(t), \boldsymbol{u}(t), \boldsymbol{p}) && \text{(20a)}\\
&\text{s.t.} && \dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{z}(t), \boldsymbol{u}(t), \boldsymbol{p}), \quad t \in [t_0, t_0 + T], && \text{(20b)}\\
& && \boldsymbol{x}_0 = \boldsymbol{x}(t_0), && \text{(20c)}\\
& && \boldsymbol{0} = \boldsymbol{g}(\boldsymbol{x}(t), \boldsymbol{z}(t), \boldsymbol{u}(t), \boldsymbol{p}), \quad t \in [t_0, t_0 + T], && \text{(20d)}\\
& && \boldsymbol{0} \geq \boldsymbol{r}(\boldsymbol{x}(t), \boldsymbol{z}(t), \boldsymbol{u}(t), \boldsymbol{p}), \quad t \in [t_0, t_0 + T]. && \text{(20e)}
\end{aligned}$$

Here the notation and dimensions remain as before, but we introduce additional inequality constraints and consider the controls, which are in a domain $U \in \mathbb{R}^{n_u}$, as optimization variables, too. The cost function J is not of special interest at this point and thus is not further specified. The problem (20) is again an infinite dimensional problem and we seek for a way to obtain a finite dimensional problem. A trick we have already seen before is to discretise the time interval $[t_0, T]$. Formally we therefore introduce a grid of $m+1$ shooting nodes τ_j with $t_0 = \tau_0 < \tau_1 < \cdots < \tau_m = T$. Note that the inequality constraints must be fulfilled pointwise. Although it is theoretically possible that the inequality constraints are violated between the nodes, it was shown that the deviation is bounded and can be kept small with sufficiently many shooting nodes Bock [1987]. Still the solution space of the controls is infinite dimensional. The multiple shooting approach therefore introduces basis functions $\boldsymbol{\Phi}_j$ for the controls. A typical choice is to use polynomials of degree d in each element of $\boldsymbol{u}$ for the basis functions, thus reducing the dimension of the controls space to $(d+1)\cdot m \cdot n_u$. In many cases it is sufficient to use piecewise constant controls. We also go along that line here for simplicity and set $\boldsymbol{u}(t) = \boldsymbol{q}_j$ for $t \in [\tau_j, \tau_{j+1}]$. On each of the intervals $[\tau_j, \tau_{j+1}],\ j = 0, \ldots, m-1$ the initial value problem

$$\begin{aligned}
\dot{\boldsymbol{x}} &= \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{z}(t), \boldsymbol{q}_j, \boldsymbol{p}), \quad t \in I_j, && \text{(21a)}\\
\boldsymbol{0} &= \boldsymbol{g}(\boldsymbol{x}(t), \boldsymbol{z}(t), \boldsymbol{q}_j, \boldsymbol{p}), \quad t \in I_j, && \text{(21b)}\\
\boldsymbol{s}_j^x &= \boldsymbol{x}(\tau_j), && \text{(21c)}\\
\boldsymbol{s}_j^z &= \boldsymbol{z}(\tau_j) && \text{(21d)}
\end{aligned}$$

is solved. The initial values $\boldsymbol{s}_j^x$ and $\boldsymbol{s}_j^z$ become optimization variables. To make sure that the resulting overall trajectory is continuous the continuity conditions

$$\boldsymbol{0} = \boldsymbol{x}(\tau_{j+1}; \boldsymbol{s}_j^x, \boldsymbol{s}_j^z, \boldsymbol{q}_j) - \boldsymbol{s}_{j+1}^x,\ j = 0, \ldots, m-1 \tag{22}$$

are introduced. To further make sure that the trajectory also complies with the original constraint (20d) the consistency conditions

$$\mathbf{0} = g(\boldsymbol{s}_j^x, \boldsymbol{s}_j^z, \boldsymbol{q}_j, \boldsymbol{p}),\ j = 0, \ldots, m \tag{23}$$

are introduced. We collect all optimization variables in

$$\boldsymbol{\xi} = \left(\boldsymbol{s}_0^x, \boldsymbol{s}_0^z, \boldsymbol{q}_0, \ldots, \boldsymbol{s}_{m-1}^x, \boldsymbol{s}_{m-1}^z, \boldsymbol{q}_{m-1}, \boldsymbol{s}_m^x, \boldsymbol{s}_m^z, \boldsymbol{p}\right) \tag{24}$$

and obtain the nonlinear program (NLP)

$$\min_{\boldsymbol{\xi}}\ J(\boldsymbol{\xi}) \tag{25a}$$

$$\text{s.t.}\ \mathbf{0} = \boldsymbol{x}(\tau_{j+1}; \boldsymbol{s}_j^x, \boldsymbol{s}_j^z, \boldsymbol{q}_j) - \boldsymbol{s}_{j+1}^x, \quad j = 0, \ldots, m-1, \tag{25b}$$

$$\mathbf{0} = \boldsymbol{x}(t_0) - \boldsymbol{s}_0^x, \tag{25c}$$

$$\mathbf{0} = \boldsymbol{g}(\boldsymbol{s}_j^x, \boldsymbol{s}_j^z, \boldsymbol{q}_j, \boldsymbol{p}), \quad j = 0, \ldots, m, \tag{25d}$$

$$\mathbf{0} \geq \boldsymbol{r}(\boldsymbol{s}_j^x, \boldsymbol{s}_j^z, \boldsymbol{q}_j, \boldsymbol{p}), \quad j = 0, \ldots, m. \tag{25e}$$

Note that the framework of multiple shooting allows for further constraints, such as continuity conditions for the controls. Note further that in other use cases additional modifications, such as introducing separate parameter variables on each interval, might be beneficial. Still, the given NLP has the strongest structural resemblance to the MHE and thus is the form of choice in this essay.

3.2 Direct multiple shooting as framework for MHE

Comparing the structure of the NLP of the multiple shooting method (25) with the output and state noise MHE (7) we note a strong resemblance. Choosing the sampling times as the shooting nodes and replacing the general cost function J in the NLP (25) with the cost function of the MHE are the straightforward identifications. As we mentioned before the function $\boldsymbol{F}$ in the MHE problem (7) is based on the numerical integration of the DAE system. In system (25) $\boldsymbol{x}(\tau_{j+1}; \boldsymbol{s}_j^x, \boldsymbol{s}_j^z, \boldsymbol{q}_j)$ is numerically computed in a similar fashion. Thus we acknowledge that the equations (25b) and (7b) can be identified with each other. Looking at the optimization variables we see that we do not have controls as optimization variables in the MHE as instead we have the state noise $\boldsymbol{w}_j$. But this does not constitute a problem as the controls in the MHE are given from the past and not optimization variables and as we can identify the noise $\boldsymbol{w}_j$ of the MHE with the $\boldsymbol{q}_j$ in the MHE problem (25) without any structural change.

In the next subsection we turn to the numerical solution of the resulting NLP. But, to avoid a too complicated presentation we introduce the notation

$$\boldsymbol{r}_k = (\boldsymbol{x}_L, \boldsymbol{z}_L, \ldots, \boldsymbol{x}_k, \boldsymbol{z}_k, \boldsymbol{w}_L, \ldots, \boldsymbol{w}_{k-1}, \boldsymbol{p}), \tag{26}$$

$$\boldsymbol{D}_k = (\bar{\boldsymbol{x}}_L, \bar{\boldsymbol{p}}_L, P_L, \boldsymbol{y}_L, V_L, \boldsymbol{u}_L, \ldots, \boldsymbol{y}_k, V_k, \boldsymbol{u}_k). \tag{27}$$

The vector $\boldsymbol{r}_k$ collects all *optimization variables*, whereas D_k collects all needed *input data*. From there we formulate the NLP (7) as

$$\min_{\boldsymbol{r}_k} \; \left\| J(\boldsymbol{r}_k; \boldsymbol{D}_k) \right\|_2^2 \tag{28a}$$

$$\text{s.t.} \;\; \boldsymbol{0} = \boldsymbol{G}(\boldsymbol{r}_k; \boldsymbol{D}_k), \tag{28b}$$

$$\boldsymbol{0} \leq \boldsymbol{H}(\boldsymbol{r}_k; \boldsymbol{D}_k). \tag{28c}$$

3.3 Numerical solution of the multiple shooting NLP

Bock suggested to use an iterative method named the *Generalized Gauß-Newton* method (GGN) to solve the problem (28). The key idea is to linearise the appearing functions around the current, namely the i-th estimate $\boldsymbol{r}_k^i$ and to iterate

$$\boldsymbol{r}_k^{i+1} = \boldsymbol{r}_k^i + \Delta \boldsymbol{r}_k^i \tag{29}$$

where $\Delta \boldsymbol{r}_k^i$ is a solution of the quadratic subproblem

$$\min_{\Delta \boldsymbol{r}_k^i} \; \left\| J(\boldsymbol{r}_k^i; \boldsymbol{D}_k) + \nabla_{\boldsymbol{r}} J(\boldsymbol{r}_k^i; \boldsymbol{D}_k)^T \Delta \boldsymbol{r}_k^i \right\|_2^2 \tag{30a}$$

$$\text{s.t.} \;\; \boldsymbol{0} = \boldsymbol{G}(\boldsymbol{r}_k^i; \boldsymbol{D}_k) + \nabla_{\boldsymbol{r}} \boldsymbol{G}(\boldsymbol{r}_k^i; \boldsymbol{D}_k)^T \Delta \boldsymbol{r}_k^i, \tag{30b}$$

$$\boldsymbol{0} \leq \boldsymbol{H}(\boldsymbol{r}_k; \boldsymbol{D}_k) + \nabla_{\boldsymbol{r}} \boldsymbol{H}(\boldsymbol{r}_k; \boldsymbol{D}_k)^T \Delta \boldsymbol{r}_k^i \tag{30c}$$

until a stopping criterion is met. The problem (30) can then be solved comparatively easy in terms of solving the resulting KKT-system, e.g. using the null space method. For details see e.g. Wright and Nocedal [1999].

3.4 MHE real-time iteration

The RTI of Kühl et al. bases on the GGN, but has one mayor difference in particular. Namely the iteration (29) is only performed *once per interval*. In this way the computational effort can be significantly decreased and allows for an actual real-time estimation. In section 4 we present some thoughts on how we can justify this trade-off and how the RTI performs in real-life examples.

Looking at the online data $\boldsymbol{D}_k$ we see that, if we want to estimate the states and parameters at time τ_k, only the newest output data $\boldsymbol{y}_k$ are missing (V_k and $\boldsymbol{u}_k$ are known from the exterior). Nevertheless we can already update the arrival cost, solve the DAE system on the interval $[\tau_{k-1}, \tau_k]$ and compute all entries of the subproblem (30), except for $V_k(\boldsymbol{y}_k - \boldsymbol{h}(\boldsymbol{x}_k, \boldsymbol{z}_k, \boldsymbol{p}))$ as we recall from subsection 2.3 and the problem (7). Note that even the sensitivities of the subproblem (30) can be computed as the newest output $\boldsymbol{y}_k$ enters the subproblem (30) only linearly and as a consequence do not appear in the sensitivity matrices. The RTI can thus be divided in two phases. The phases are the preparation phase in which the above mentioned steps are executed and the estimation phase which starts as soon as the latest output $\boldsymbol{y}_k$ becomes known and in which the system (30) is solved. The RTI algorithm for the output and state noise MHE is displayed in algorithm 1, which is the version as presented in Kühl et al. [2011] with some presentation related modifications.

Actually the algorithm in this form does not take into account how to deal with the first samples when the overall time does not yet exceed the size of the time horizon. Two remedies are to either fill up the vectors $\boldsymbol{r}_k$ and $\boldsymbol{D}_k$ with initial values or to let their size increase at the first iterations. The latter is labelled as a *growing horizon approach*.

Indeed the algorithm does even comply with the formal definition of the real-time paradigm under certain assumptions e.g. regarding the numerical integration (Kühl et al. [2011]).

4 Discussion

As mentioned before we want to give some thoughts on the justification for computing only one update of $\boldsymbol{r}_k$ per interval. Of course we in general do not solve the problem (28) optimally and at first sight one could think that our estimations thus might be rather bad. But, under the assumption that the optimal solutions of the MHE problem on successive intervals do not differ too strongly, we believe the successive updates of $\boldsymbol{r}_k$ throughout the different intervals to be steps towards a good estimation. We still need sufficiently good initial guesses for that argument to be reasonable. Having once reached a good estimation the old estimations $\hat{\boldsymbol{x}}_{k-1},\ \hat{\boldsymbol{z}}_{k-1},\ \hat{\boldsymbol{p}}_{k-1}$ are already close to a solution of the new MHE. Thus the norm of the increment $\Delta\boldsymbol{r}_k = \Delta\boldsymbol{r}_k^0$ will be rather small and the norm of higher iterates $\Delta\boldsymbol{r}_k^i$ most likely even smaller. Higher iterates $\boldsymbol{r}_k^i$ hence do not differ strongly from $\boldsymbol{r}_k = \boldsymbol{r}_k^0$ which justifies to only perform one iteration.

Besides those theoretical thoughts the very good performance of the RTI in case studies particularly justifies the conception of the RTI. For details of the performance in two case studies see again Kühl et al. [2011].

Still, also the MHE RTI has some disadvantages. As just indicated especially the beginning of the estimation proves difficult and one has to expect sub-optimal solutions if the initial guesses are not eminently good. Moreover, we have to mind that the multiple shooting method and as a consequence the RTI only finds local solutions and does not converge globally to the global minimum. Finally it is not yet known how to determine the best time window size and other algorithm parameters.

Nevertheless, the MHE real-time iteration is already a powerful method for real-time state and parameter estimation and with ongoing research one can hope that some of the named disadvantages can be eradicated in the future.

Algorithm 1: MHE real-time iteration

Input: multiple shooting data vector:

$\boldsymbol{r}_{k-1} = (\boldsymbol{x}_{L-1}, \boldsymbol{z}_{L-1}, \dots, \boldsymbol{x}_{k-1}, \boldsymbol{z}_{k-1}, \boldsymbol{w}_{L-1}, \dots, \boldsymbol{w}_{k-2}, \boldsymbol{p})$

online data set from previous intervals:

$\boldsymbol{D}_{k-1} = (\bar{\boldsymbol{x}}_{L-1}, \bar{\boldsymbol{p}}_{L-1}, P_{L-1}, \boldsymbol{y}_{L-1}, V_{L-1}, \boldsymbol{u}_{L-1}, \dots, \boldsymbol{y}_{k-1}, V_{k-1}, \boldsymbol{u}_{k-1},)$

online data which becomes known during the iteration:

$\boldsymbol{y}_k$, V_k, $\boldsymbol{u}_k$

Output: current estimates $\hat{\boldsymbol{x}}_k$, $\hat{\boldsymbol{z}}_k$, $\hat{\boldsymbol{p}}_k$,
updated $\boldsymbol{r}_k$ and updated $\bar{\boldsymbol{x}}_L$, $\bar{\boldsymbol{p}}_k$ for $\boldsymbol{D}_k$

At initial sample $k = 0$ provide initial guess $\boldsymbol{w}_0$, $\boldsymbol{D}_0$

for *samples* $k = 1, 2, \dots$ **do**

  Preparation phase *for horizon* $[\tau_L, \tau_k]$ ***before*** τ_k **:**

    Update arrival cost data $(\bar{\boldsymbol{x}}_L, \bar{\boldsymbol{p}}_L)$ and P_L with eq. (17)

    Shift data vectors:

      $\boldsymbol{r}_k^- = (\boldsymbol{x}_L, \boldsymbol{z}_L, \dots, \boldsymbol{x}_k^-, \boldsymbol{z}_k^-, \boldsymbol{w}_L, \dots, \boldsymbol{w}_{k-1}^-, \boldsymbol{p})$

      $\boldsymbol{D}_k = (\bar{\boldsymbol{x}}_L, \bar{\boldsymbol{p}}_L, P_L, \boldsymbol{y}_L, V_L, \boldsymbol{u}_L, \dots, *, V_k, \boldsymbol{u}_k)$

      where $*$ is a wildcard for the not yet known output $\boldsymbol{y}_k$

    Solve DAE system on interval $[\tau_{k-1}, \tau_k]$ and set

      $\boldsymbol{x}_k^- := \boldsymbol{x}(\tau_k; \boldsymbol{x}_{k-1}, \boldsymbol{z}_{k-1})$

      $\boldsymbol{w}_{k-1}^- = 0$

      $\boldsymbol{z}_k^- := \boldsymbol{z}(\tau_k; \boldsymbol{x}_{k-1}, \boldsymbol{z}_{k-1})$

    Compute *all but one vector* component of problem (30)

      except for $V_k(\boldsymbol{y}_k - \boldsymbol{h}(\boldsymbol{x}_k, \boldsymbol{z}_k, \boldsymbol{p}))$

    Compute *all matrix* components of problem (30)

  end of preparation phase

  Estimation phase *for horizon* $[\tau_{L+1}, \tau_k]$ ***at*** τ_k **:**

    Complete vector $\boldsymbol{D}_k$ as $\boldsymbol{y}_k$ becomes known

    Solve (30) for $\Delta \boldsymbol{r}_k$

    Update:

      $\boldsymbol{r}_k = \boldsymbol{r}_k^- + \Delta \boldsymbol{r}_k$

      $\hat{\boldsymbol{x}}_k = \boldsymbol{x}_k$

      $\hat{\boldsymbol{z}}_k = \boldsymbol{z}_k$

      $\hat{\boldsymbol{p}}_k = \boldsymbol{p}$

  end of estimation phase

end

5 References

Hans Georg Bock. Numerical treatment of inverse problems in chemical reaction kinetics. In *Modelling of chemical reaction systems*, pages 102–125. Springer, 1981.

Hans Georg Bock. *Randwertproblemmethoden zur Parameteridentifizierung in Systemen nichtlinearer Differentialgleichungen*. Number 183. Der Math.-Naturwiss. Fakultät der Universität Bonn, 1987.

Hans Georg Bock and Karl-Josef Plitt. A multiple shooting algorithm for direct solution of optimal control problems. *IFAC Proceedings Volumes*, 17(2):1603–1608, 1984.

Fred Daum. Nonlinear filters: beyond the kalman filter. *IEEE Aerospace and Electronic Systems Magazine*, 20(8):57–69, 2005.

Arthur Gelb. *Applied optimal estimation*. MIT press, 1974.

Peter Kühl, Moritz Diehl, Tom Kraus, Johannes P. Schlöder, and Hans Georg Bock. A real-time algorithm for moving horizon state and parameter estimation. *Computers & Chemical Engineering*, 35(1):71 – 83, 2011.

James B. Rawlings and Bhavik R. Bakshi. Particle filtering and moving horizon estimation. *Computers & Chemical Engineering*, 30(10):1529 – 1541, 2006.

Stephen Wright and Jorge Nocedal. Numerical optimization. *Springer Science*, 35 (67-68):7, 1999.