sensors

MDPI

Article

Nonlinear Constrained Moving Horizon Estimation Applied to Vehicle Position Estimation

Jonathan Brembeck

Institute of System Dynamics and Control, Robotics and Mechatronics Center, German Aerospace Center (DLR), 82234 Weßling, Germany; jonathan.brembeck@dlr.de; Tel.: +49-8153-28-2472

Received: 21 March 2019; Accepted: 11 May 2019; Published: 16 May 2019

Abstract: The design of high–performance state estimators for future autonomous vehicles constitutes a challenging task, because of the rising complexity and demand for operational safety. In this application, a vehicle state observer with a focus on the estimation of the quantities position, yaw angle, velocity, and yaw rate, which are necessary for a path following control for an autonomous vehicle, is discussed. The synthesis of the vehicle's observer model is a trade-off between modelling complexity and performance. To cope with the vehicle still stand situations, the framework provides an automatic event handling functionality. Moreover, by means of an efficient root search algorithm, map-based information on the current road boundaries can be determined. An extended moving horizon state estimation algorithm enables the incorporation of delayed low bandwidth Global Navigation Satellite System (GNSS) measurements—including out of sequence measurements—as well as the possibility to limit the vehicle position change through the knowledge of the road boundaries. Finally, different moving horizon observer configurations are assessed in a comprehensive case study, which are compared to a conventional extended Kalman filter. These rely on real-world experiment data from vehicle testdrive experiments, which show very promising results for the proposed approach.

Keywords: automotive applications; nonlinear observer; Kalman filter; constrained estimation; nonlinear gradient descent search; vehicle state estimation; moving horizon estimation; GNSS; IMU; INS

1. Introduction

Many demanding mechatronic systems, like the German Aerospace Center's (DLR's) ROMO (short for ROboMObil—DLR's robotic electric vehicle) [1], employ state dependent nonlinear optimization-based control (e.g., [2]), which needs an accurate knowledge of the system states. Often, these states cannot be gathered directly through sensors, as an appropriate measurement principle for the searched quantity is not available (e.g., determination of the state of charge of a battery), or the sensor is expensive and therefore it is desirable to be economized (e.g., vehicle over ground velocity sensing). Especially for future autonomous vehicles, it is necessary to determine an accurate vehicle position so as to guarantee a reliable vehicle path/trajectory generation, and to follow control functionality cf. [2–4].

Here, a novel nonlinear position estimator, relying on a moving horizon estimator approach, which fuses inertial navigation system (INS) and Global Navigation Satellite System (GNSS) sensor data based on a model-based observer framework [2] is proposed. This framework relies on multiphysical prediction model design in Modelica [5], and an automated tool chain to incorporate these by means of the Functional Mockup Interface (FMI) [6] technology, with well proven nonlinear Kalman filter-based algorithms (cf. e.g., [2,7]).

State of the Art in GNSS/INS Coupled Estimation

The topic of INS- and GNSS-based sensor fusion for different applications in ground or air vehicles was a large scope in the research of the last decades. A quiet comprehensive overview of methods such as dead reckoning can be found in the literature, for example, [8–10]. Moreover, in the literature [11], time delayed sensor fusion for sigma point Kalman filters (SPKF) is presented. In this approach, the system state is augmented with an N samples delayed state, and a cross correlation of the lagged measurement with the covariance at t_{k-N} and with the covariance of the actual time instance t_k is performed. In the literature [12], the inertial and magnetometer measurements are delayed to correspond with the GNSS data, and then fused in an observer, resulting in delayed estimates. The current estimates are afterwards calculated integrating the IMU data with the delayed estimates as the start values. Both approaches try to incorporate the delayed information of the GNSS data with a postponed prediction [12] or correction [11].

On the contrary, here, a methodology is proposed that incorporates all of the past, delayed data in every estimation step by means of a nonlinear multiphysical vehicle prediction model and a real-time capable nonlinear moving horizon estimation (MHE) algorithm. Additionally, the knowledge of the road boundaries, for example, provided by a map system with a virtual horizon, are incorporated so as to restrict the vehicle position within the carriage way. A different approach incorporating a direct collocation technique in the MHE formulation was used for the estimation of an aircraft trajectory in the literature [13]. On the contrary, in [14] a Gauss–Newton method was used to solve the nonlinear least-squares problem of the MHE formulation of a Global Positioning System (GPS) and dead reckoning integrated navigation system, by linearizing the optimization problem along the previous state estimation in each iteration. Another method to compensate the missing data of the slow sampled measurements in a (linear) MHE formulation with bounded measurement noise is proposed by the authors of [15], by means of the introduction of prediction values in the missing data gaps. The interested reader can find examples of the recent research advances in the development of new moving horizon estimation techniques in the literature [16,17].

2. The Observer Setup–Problem Formulation of the MHE-Based GNSS/INS Fusion

In a realistic application with a consumer quality GPS sensor, a latency between 50–1000 ms [11,12] is usually considered. This implies a drastic delay in comparison to other in-vehicle sensor systems, for example, the inertial measurement unit (IMU). Here, this fact motivates the discussed observer example with highly delayed measurements in a vehicle position estimation application.

For the experimental evaluation of the constrained state estimation of DLR's ROboMObil, several test campaigns were carried out at the ADAC (General German Automobile Club) vehicle testing ground in Kempten, Germany. Here, tests of an advanced version of the interactive vehicle path following control (PFC) [2,3] have been performed. In the top left of Figure 1, a portrait of the testing track is shown. The green line depicts the preplanned vehicle path, while the road boundaries are represented by the dotted-orange lines.

The ground-truth data was gathered with help of a differential GPS in order to guarantee high fidelity vehicle state measurements for the experimental validation. In Figure 2, the situation of the MHE estimator proposed here with delayed measurements is depicted. At the current time instance t_k, the estimator receives the measurements of the current vehicle yaw rate state ($y_k = \dot{\psi}^{\mathrm{C}}_{\mathrm{IMU}}$) and the delayed measurements ($\boldsymbol{y}^*_k = \boldsymbol{p}^{\mathrm{I}}_{\mathrm{C}_{\mathrm{GPS}}} = \left\{x^{\mathrm{I}}_{\mathrm{c}_{\mathrm{GPS}}}, y^{\mathrm{I}}_{\mathrm{c}_{\mathrm{GPS}}}\right\}$), which are delayed for n_{d} samples, and thus belong to the past vehicle state ($\boldsymbol{x}_{k-n_{\mathrm{d}}}$). Later, it will be shown that the incorporation of these delayed measurements in a conventional one-step estimation approach (i.e., a Kalman filter) leads to a poor performance or even instability.

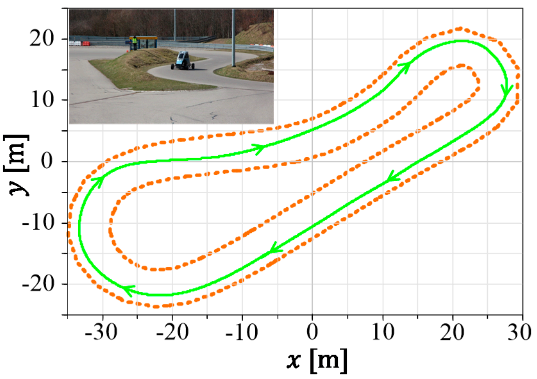


Figure 1. Path following control (PFC) experiment with ROMO (short for ROboMObil—German Aerospace Center's (DLR's) robotic electric vehicle) at the General German Automobile Club's (ADAC's) test facilities in Kempten.

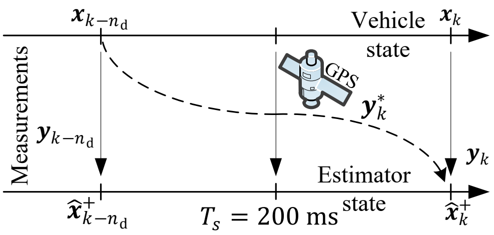


Figure 2. Delayed Global Positioning System (GPS) measurements incorporation in vehicle position estimation.

2.1. The Extended Single Track Prediction Model

As the scope of the proposed observer (as implied in the introduction of this section) lies on the observation of the actual vehicle states, $p^{\mathrm{I}}_{\mathrm{C_{act}}}, \psi^{\mathrm{I}}_{\mathrm{C_{act}}}, v^{\mathrm{C}}_{\mathrm{act}}, \dot{\psi}^{\mathrm{C}}_{\mathrm{act}}$ for path following controllers in autonomous vehicles, as proposed by the authors of [2], an observer synthesis model reproduction depth has been chosen that meets the requirements of the computational efficiency and fidelity. An extended single track model (ESTM) was chosen for the forward driving operation mode $v \geq 0$, which incorporates vehicle standstill functionalities, rolling resistance, drag forces, and Pacejka's Magic Formula [18] lateral characteristics of the tires. The following nomenclature is defined (cf. Figure 3):

- $(\cdot)^{\mathrm{C}}$: Quantity expressed in the car coordinate system with origin in the center of gravity
- $(\cdot)_{\mathrm{C}}$: Car quantity expressed in the inertial coordinate system—short for $(\cdot)^{\mathrm{I}}_{\mathrm{C}}$

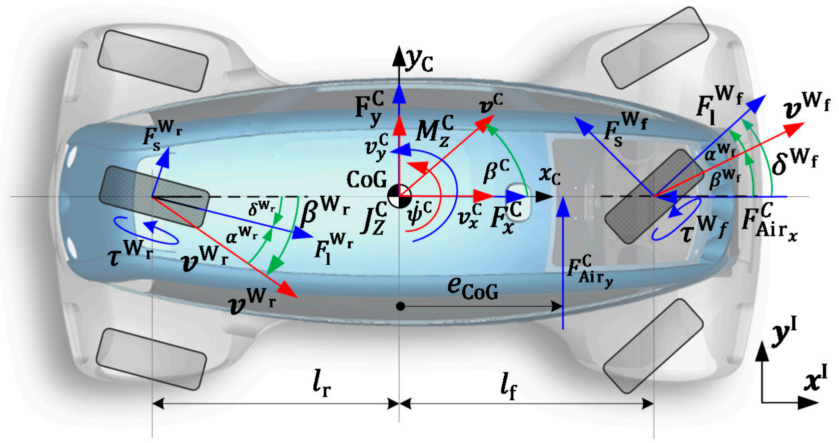


Figure 3. Vehicle dynamics quantities of the extended single track model.

The details of the extended single track model equation derivation are given in the Appendix A. It is worth mentioning that with the Modelica modeling technology, it was easily possible to integrate "if-else" constructs, which are efficiently processed by the event handling features of the compiler, while experimenting with varying discretization methods and sample rates, as the model is in continuous-time formulation. The vehicle state vector is denoted as follows, and is graphically exemplified in Figure 3:

$$x^{\mathrm{C}} = \left\{\beta^{\mathrm{C}}, v^{\mathrm{C}}, \dot{\psi}^{\mathrm{C}}, \psi_{\mathrm{C}}, x_{\mathrm{C}}, y_{\mathrm{C}}\right\} \tag{1}$$

2.2. Road Boundaries Constraint Formulation and Evaluation

In this section, a methodology is described for incorporating advance-known street boundaries (e.g., from digital maps in combination with vision sensors) into the vehicle position estimation. It is assumed that the initial vehicle position ($\boldsymbol{p}_{\mathrm{C}}^{\mathrm{I}}$) is sufficiently precisely known, and the corresponding path parameter (s^*) (cf. Equation (2)) can be determined by means of a time independent path interpolation (TIPI) [2,3], which is explained in the following.

Ideally, the actual path position and the position of the car coincide, $\boldsymbol{p}_{\mathrm{P}}(s^*) = \boldsymbol{p}_{\mathrm{C}}$ (cf. Figure 4). To approximate this condition, it is necessary to minimize the displacement ($\boldsymbol{e}(s)$) between the vehicle and reference path, as depicted in the following:

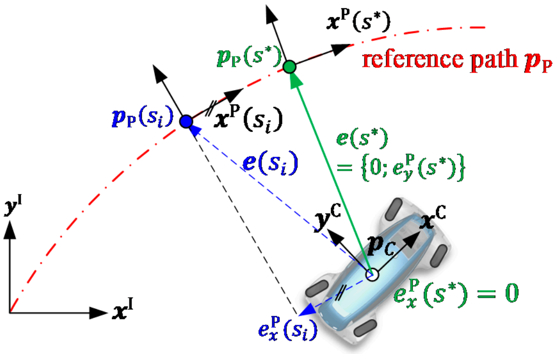


Figure 4. Graphical representation of the dynamic root finding to determine s^*.

The geometrical interpretation of this minimization objective is that $\boldsymbol{p}_{\mathrm{P}}(s^*)$ can be determined by projecting $\boldsymbol{p}_{\mathrm{C}}$ orthogonally on the path $\boldsymbol{p}_{\mathrm{P}}(s)$. For the TIPI this condition implies that the inverse of the maximal curvature of the demanded vehicle path defines the maximum lateral displacement for which s^* exists: $e_y^{\mathrm{P}}(s^*) \leq 1/\kappa_{\mathrm{P}}(s^*)$.

$$s^* = \underset{s}{\operatorname{argmin}} \| \underbrace{\boldsymbol{p}_{\mathrm{P}}(s) - \boldsymbol{p}_{\mathrm{C}}}_{\boldsymbol{e}(s)} \|_{2} \tag{2}$$

The interested reader can find details on how this optimization problem is solved in the literature [3].

The parametric path $\boldsymbol{\Lambda}_{\mathrm{c}}(s)$ contains the following quantities for the calculation of the road boundaries constraints: the position $x_{\mathrm{P}}^{\mathrm{I}}(s)$ and $y_{\mathrm{P}}^{\mathrm{I}}(s)$ of the road middle lane, the positions of the left ($l_x^{\mathrm{I}}(s), l_y^{\mathrm{I}}(s)$) and the right ($r_x^{\mathrm{I}}(s), r_y^{\mathrm{I}}(s)$) border, and the corresponding path orientation $\psi_{\mathrm{P}}^{\mathrm{I}}(s)$ and its curvature $\kappa_{\mathrm{P}}(s)$.

The extension of the vehicle prediction model (cf. Appendix A) with the roadway boundaries interpolation yields an extra state ($\dot{s}$), which is a non-physical state that belongs to the estimation task. The experimental tests considering $\dot{s}$ as an estimated state showed that this configuration leads to unsatisfactory results in the overall observer performance. Therefore, this state is separated from the state correction step in the Kalman algorithms (cf. Section 3.2).

In Figure 5, the calculation of the roadway border constraints is graphically shown. By means of the above described algorithm, a path parameter (s_i) can be found for which the longitudinal derivation error of the vehicle's actual position ($\boldsymbol{p}_{\mathrm{C}}^{\mathrm{I}}$) tends to zero.

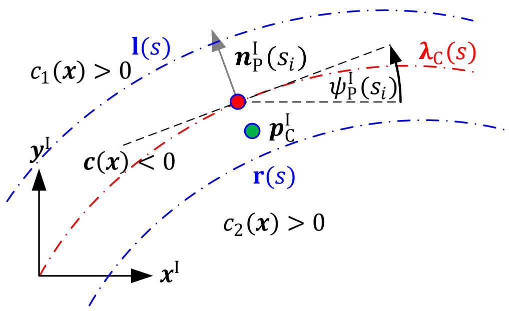

$\mathbf{l}(s)$
$c_1(\boldsymbol{x}) > 0$
$\boldsymbol{n}^{\mathrm{I}}_{\mathrm{P}}(s_i)$
$\boldsymbol{\lambda}_{\mathrm{C}}(s)$
$\psi^{\mathrm{I}}_{\mathrm{P}}(s_i)$
$\boldsymbol{y}^{\mathrm{I}}$
$\boldsymbol{c}(\boldsymbol{x}) < 0$
$\boldsymbol{p}^{\mathrm{I}}_{\mathrm{C}}$
$\mathbf{r}(s)$
$c_2(\boldsymbol{x}) > 0$
$\boldsymbol{x}^{\mathrm{I}}$

Figure 5. Graphical analysis of the street boundary calculation.

By means of the path normal vector ($\boldsymbol{n}^{\mathrm{I}}_{\mathrm{P}}(s_i)$), represented in the inertial coordinate system (Equation (2)), an inequality function $\boldsymbol{c}(\boldsymbol{x}) \leq 0$ is calculated that penalizes the positions outside of the roadway borders, as follows:

$$\boldsymbol{n}^{\mathrm{I}}_{\mathrm{P}}(s_i) = \left\{-\sin\left(\psi^{\mathrm{I}}_{\mathrm{P}}(s_i)\right), \cos\left(\psi^{\mathrm{I}}_{\mathrm{P}}(s_i)\right)\right\} \tag{3}$$

$$\boldsymbol{c}(\boldsymbol{x}) = \begin{bmatrix} -l^{\mathrm{I}}_x(s_i) + x^{\mathrm{I}}_{\mathrm{Cact}} & -l^{\mathrm{I}}_y(s_i) + y^{\mathrm{I}}_{\mathrm{Cact}} \\ r^{\mathrm{I}}_x(s_i) - x^{\mathrm{I}}_{\mathrm{Cact}} & r^{\mathrm{I}}_y(s_i) - y^{\mathrm{I}}_{\mathrm{Cact}} \end{bmatrix} \cdot \boldsymbol{n}^{\mathrm{I}}_{\mathrm{P}}(s_i) \tag{4}$$

This nonlinear inequality function, $\boldsymbol{c}(\boldsymbol{x})$, can be handled directly by a posteriori constraint projection method for an (extended) Kalman filter. Details on the simplified Newton descent search algorithm used here are given in Appendix B. For the real-time capable moving horizon estimation algorithm in Section 3. it is necessary to linearize $\boldsymbol{c}(\boldsymbol{x})$ at all of the time instances t_k, where it is likely that a constraint may be violated by the estimator $\boldsymbol{c}(\boldsymbol{x}_k) > -\epsilon$, as follows:

$$\boldsymbol{c}(\boldsymbol{x}) \cong \boldsymbol{c}(\boldsymbol{x}_k) + \left.\frac{\partial \boldsymbol{c}}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}_k} \cdot (\boldsymbol{x} - \boldsymbol{x}_k) \tag{5}$$

By rearranging Equation (4), the linearized inequality description in Equation (5) can be formulated as follows:

$$\boldsymbol{C}_k \cdot \boldsymbol{x} \leq \boldsymbol{d}_k \overset{\text{yields}}{\to} \underbrace{\left.\frac{\partial \boldsymbol{c}}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}_k}}_{\boldsymbol{C}_k} \cdot \boldsymbol{x} \leq \underbrace{\left.\frac{\partial \boldsymbol{c}}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}_k} \cdot \boldsymbol{x}_k - \boldsymbol{c}(\boldsymbol{x}_k)}_{\boldsymbol{d}_k} \tag{6}$$

Figure 6 shows an example for the calculation of the nonlinear constraint function. In the left plot, a street is marked with the left $\boldsymbol{l}(s)$ and the right $\boldsymbol{r}(s)$ street boundaries, while the car (red line with direction arrows) crosses the right boundary in the hairpin curve. This leads to a violation of the right border constraint condition $c_1(\boldsymbol{x}) > 0$, $\forall\, t \in [19.6\ 22.1] \wedge [29.4\ 30]$, as can be seen in the right plot of Figure 6.

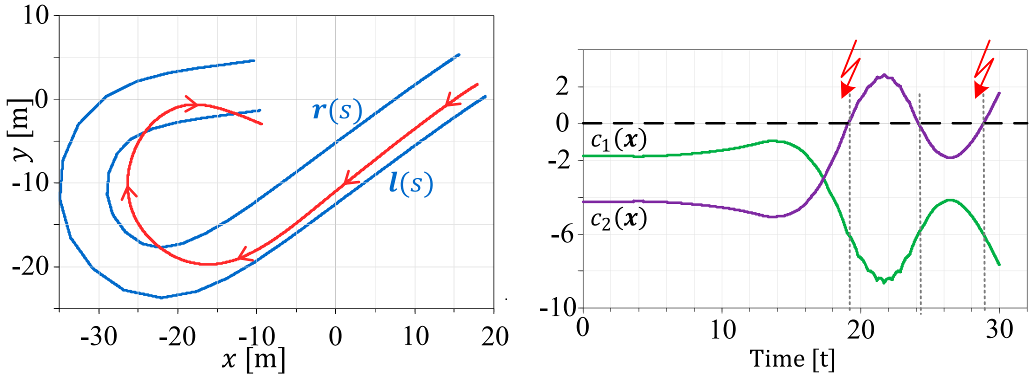


Figure 6. Example path with boundary violation (**Left**) and constraints evaluation (**Right**).

Assuming that only the vehicle yaw rate ($\dot{\psi}^{\mathrm{C}}_{\mathrm{act}}$) is available to the (extended) Kalman filter, the position estimate would drift away, as shown in Figure 7 dark green line). Making use of the proposed boundary estimation approach used here, in combination with the inequality handling feature from the author of [2], yields a bounded and valid result (light green line), as follows:

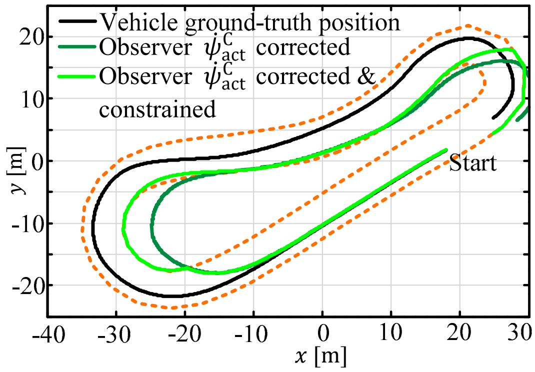


Figure 7. Observer behavior with and without path constraints.

3. Real-Time Nonlinear Moving Horizon Estimation

In this section, a nonlinear moving horizon estimator (MHE), which utilizes multiple past measurements to estimate the state at the current time instance, is introduced. In addition to the state constraints, delayed measurements can be directly incorporated into the problem formulation.

Referring to the literature [19], the approach of the moving horizon estimation is the reformulation of the general optimization objective of the Kalman filter theory, also known as the full-information filter. In the nonlinear case, the minimization problem can be written as follows:

$$\min_{\xi_k} \parallel \boldsymbol{x}_0 - \hat{\boldsymbol{x}}_0 \parallel^2_{\boldsymbol{I}_0^+} + \sum_{i=1}^{N} \parallel \boldsymbol{y}_i^{\mathrm{m}} - \boldsymbol{h}(\boldsymbol{x}_i) \parallel^2_{\boldsymbol{R}^{-1}} + \sum_{i=0}^{N-1} \parallel \boldsymbol{x}_{i+1} - \boldsymbol{f}(\boldsymbol{x}_i, \boldsymbol{u}_i) \parallel^2_{\boldsymbol{Q}^{-1}} \\ \xi_k = \left(\boldsymbol{x}_0^T,\, \boldsymbol{x}_1^T, \ldots, \boldsymbol{x}_N^T\right)^T \tag{7}$$

where in the nomenclature of $\min\limits_{\boldsymbol{x}} \|\boldsymbol{e}(\boldsymbol{x})\|_{\boldsymbol{W}}$ is denoted a with $\boldsymbol{W}$, and is a weighted least squares minimization problem of a function $\boldsymbol{e}$ with respect to $\boldsymbol{x}$. The functions $\boldsymbol{f}$ and $\boldsymbol{h}$ are the nonlinear state functions, resprectivley, of the model output equations. The matrix $\boldsymbol{I}_0^+ = \left(\boldsymbol{P}_0^+\right)^{-1}$ and the vector $\hat{\boldsymbol{x}}_0$ represent the initial guess values of the covariance and the system state. The matrix $\boldsymbol{Q}$ denotes the covariance of the system states, and its entries constitute the confidence in the underlying prediction model and can be tuned by the application engineer. The second tuning matrix $\boldsymbol{R}$ represents the confidence in the actual measurements. The well-known Kalman filter algorithms (e.g., extended Kalman filter (EKF) and unscented Kalman filter (UKF)) are special cases of this optimization objective, in the case that only the measurements at the current time instance (t_k) are available [19]. As the dimension, N, of the optimization problem would grow tremendously with proceeding time, in MHE theory, the time span is limited to a predefined length of previous time instances and is shifted in every sample step.

Therefore, a sliding window with M steps back from the actual time instance (t_k), is considered to smooth the state estimation. In every sample step (T_s), this window is shifted one-step ahead—in Figure 8 this is illustrated for one measurement variable, y^{m}. At the filter initialization (t_0), only one measurement is available, and therefore the measurement storage must be filled with $r < M$ steps before the window starts to move. Afterwards, the window length is kept constant and all of the past M measurements are taken into account (the window with M measurements is colored in green in the subsequent figures). In other words, MHE can be seen as a real-time calculable approximation of the full-information filter.

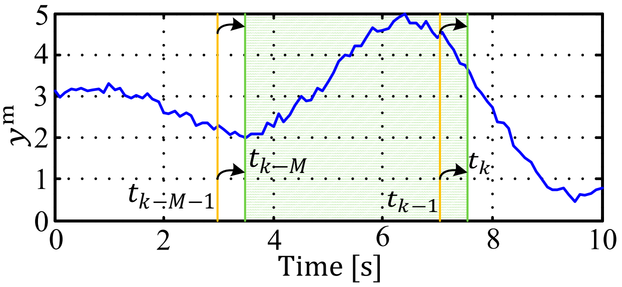


Figure 8. Schematic representation of a moving measurement window.

In this way, the estimate gets more robust against external disturbances, delayed measurements can be incorporated (cf. Section 3.2.3), and also constraints can be imposed directly [19].

By means of the proposed estimation framework presented in [2], it is also possible to incorporate complex nonlinear Modelica-based prediction models to moving the horizon estimation. In the context of toolchains for observer code generation and nonlinear constraint MHE, different research studies were recently published that utilize the ACADO toolbox for embedded [20] and real-time moving horizon estimation [21,22]. They make use of the real-time iteration (RTI) scheme transferred to the MHE approach in he literature [23]. The basic strategy is to discretize the estimation problem with a multiple shooting discretization using numerical integration. Then, the main idea of the RTI scheme is to use the shifted state variables of the previous optimization run as the new linearization point, and to perform only one SQP step per sample time [20].

In this work, the problem formulation in Equation (7) is extended for nonlinear systems incorporating linear state constraints, tailored to meet real-time application restrictions by means of a nonlinear gradient descent search. Details on the reasons for this approach are given in the following section.

$$\begin{aligned} &\min_{\xi_k} g\left(\xi_k = \left(\boldsymbol{x}_{k-M}^T, \boldsymbol{x}_{k-M+1}^T, \dots, \boldsymbol{x}_k^T\right)^T\right) \\ &\text{s.t.} \boldsymbol{A} \cdot \xi_k = \boldsymbol{b} \\ &\quad \boldsymbol{C} \cdot \xi_k \leq \boldsymbol{d} \end{aligned} \tag{8}$$

$$g(\xi_k) = \| \boldsymbol{x}_{k-M} - \hat{\boldsymbol{x}}_{k-M}^{+} \|^2_{\boldsymbol{I}_{k-M}^{+}} + \sum_{i=k-M}^{k} \| \boldsymbol{y}_i^{\mathrm{m}} - \boldsymbol{h}(\boldsymbol{x}_i) \|^2_{\boldsymbol{R}^{-1}} + \sum_{i=k-M+1}^{k} \| \boldsymbol{x}_i - \boldsymbol{x}_{\mathrm{int},i} \|^2_{\boldsymbol{Q}^{-1}} \tag{9}$$

$$\begin{gathered} \boldsymbol{x}_{\mathrm{int},k-M} = \hat{\boldsymbol{x}}_{k-M}^{+} \\ \boldsymbol{x}_{\mathrm{int},i} = \boldsymbol{f}_{i|i-1}(\boldsymbol{x}_{\mathrm{int},i-1}, \boldsymbol{u}_{i-1}) \\ (i = k-M+1, \dots, k) \end{gathered} \tag{10}$$

All of the system function evaluations were done by the FMI [6] (cf. also Section 3.2) and are marked in red in the following section. The optimization vector (ξ_k) is assembled with M subsequent discrete state vectors within the current estimation window. The optimization cost function $g(\xi_k)$ in Equation (9) is composed of the following three parts.

The first argument $\| \cdot \|_{\boldsymbol{I}_{k-M}^{+}}$ is the arrival cost, which summarizes all of the available information prior to the estimation window; this can also be seen as a regulation term on the states at t_{k-M} [21]. It is introduced so as to guarantee that the oldest estimation ($\boldsymbol{x}_{k-M}$) is coincident with the corrected Kalman filter state estimation ($\hat{\boldsymbol{x}}_{k-M}^{+}$) weighted with the information matrix ($\boldsymbol{I}_{k-M}^{+}$). Note that in every moving horizon estimation step, a Kalman filter step from $k-M-1$ to $k-M$ is performed to fulfill the Kalman state estimation theory—Equation (7). The information matrix, $\boldsymbol{I}_{k-M}^{+}$, is calculated via the inverse of the covariance matrix $\left(\boldsymbol{P}_{k-M}^{+}\right)^{-1}$. As direct matrix inversion should be avoided, because of the numerical stability and accuracy, the author proposes using a square root (SR)-UKF or SR-EKF Kalman filter algorithm that uses square-root decomposition and rank 1 updates to propagate the covariance matrix in a lower triangular form, $\boldsymbol{P} = \boldsymbol{L} \cdot \boldsymbol{U}$. The inverse of the lower triangular ($\boldsymbol{L}$) can be efficiently calculated by the LAPACK routine DTRTRI, and therefore, the propagated information matrix results in $\boldsymbol{I}^{+}_{k-M} = \boldsymbol{L}^{-1} \cdot \boldsymbol{L}$.

The second argument, $\|\cdot\|_{\boldsymbol{R}^{-1}}$ is the, over the time instances ($k - M$ to M) summarized and with $\boldsymbol{R}^{-1}$ weighted, difference between the available measurements $\boldsymbol{y}^{\mathrm{m}}_{i}$ and the output equations of the underlying prediction model $\boldsymbol{h}(\boldsymbol{x}_i)$ as function of the states of the current optimization vector ξ_k.

Finally, the third argument, $\|\cdot\|_{\boldsymbol{Q}^{-1}}$, denotes the, over the time instances ($k - M$ to M) summarized and with $\boldsymbol{Q}^{-1}$ weighted, difference between the optimized states $\boldsymbol{x}_i$ and the open loop integrated prediction model states $\boldsymbol{x}_{\mathrm{int}}$ in Equation (10). $\boldsymbol{x}_{\mathrm{int}}$ is calculated only once per optimization step by a simulation using $\hat{\boldsymbol{x}}^{+}_{k-M}$ as start vector and $\boldsymbol{u}_i$ as input vector.

In Figure 9, a qualitative graphical interpretation of the optimization problem for a scalar problem with $n_x = n_y = 1$ is shown. The circle points of the quantities denote the particular values at a discrete-time instance, which is evaluated in the objective function Equation (9).

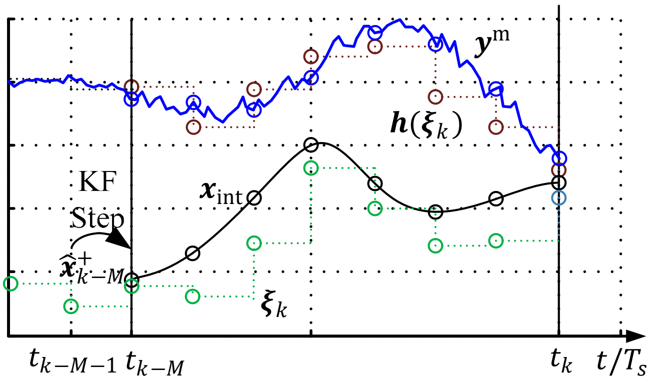

$\boldsymbol{y}^{\mathrm{m}}$
$\boldsymbol{h}(\boldsymbol{\xi}_k)$
KF
Step
$\boldsymbol{x}_{\mathrm{int}}$
$\hat{\boldsymbol{x}}^{+}_{k-M}$
$\boldsymbol{\xi}_k$
t_{k-M-1}
t_{k-M}
t_k
t/T_s

Figure 9. Schematic representation of a moving measurement window.

The complete algorithm is summarized in Algorithm 1. In the first step, all of the past measurements are stored in a first in first out (FIFO) ring buffer. As long as not enough measurements for the complete window M are available, the measurements and the model inputs are appended to the buffer.

Algorithm 1. Algorithm for a nonlinear MHE

```
Set $k = 0$ ($k \in \mathbb{N}^{+}$) and set $\boldsymbol{x}_k = \boldsymbol{x}_0$
while (brake==false) do
        Fill ring buffer with measurements and system inputs:
    if $k < M$ then
        append $\boldsymbol{u}_k$ to $\boldsymbol{u}$ and $\boldsymbol{y}^{\mathrm{m}}_{k}$ to $\boldsymbol{y}^{\mathrm{m}}$
    else
        left shift one entry of $\boldsymbol{u}$ and $\boldsymbol{y}^{\mathrm{m}}$
        and append $\boldsymbol{u}_k$ respectively $\boldsymbol{y}^{\mathrm{m}}_{k}$
    end if
        Optimize over stored measurement window (Equation (8)):
    if $k > M$ then
        Propagate $\hat{\boldsymbol{x}}^{+}_{k-M-1}$ via a Kalman Filter step (e.g., EKF or UKF cf. 2):
        xKF($\hat{\boldsymbol{x}}^{+}_{k-M-1}$) $\rightarrow \hat{\boldsymbol{x}}^{+}_{k-M}$, $\boldsymbol{I}^{+}_{k-M}$
        Project states on the constrained area (c.f. Appendix B)
        $\min_{\boldsymbol{x}} \|\boldsymbol{x} - \hat{\boldsymbol{x}}^{+}_{k-M}\|$  s.t.  $\boldsymbol{c}(\boldsymbol{x}) \leq 0$
end if
$k = k + 1$
end while
```

3.1. A Nonlinear Gradient Descent Opimization Algorithm for MHE

For the solution of the proposed MHE problem formulation, Equation (8) to Equation (10), a nonlinear gradient descent search algorithm (NG) was chosen, because for this solver, only the first derivatives of the minimization objective (Equations (8)–(10)) are needed, which is an important constraint for the available interfaces of the extended FMI 2.0 co-simulation interface (see [2]—Section 4.2.2). Furthermore, linear equality and inequality constraints can be incorporated easily, and the method can be stopped after every optimization step, still offering a reliable sub-optimal solution. The latter is important if the optimization is not finalized at the next sample instant. The algorithm of the constrained NG is given in Algorithm 2, for more details see, for example [24]. This gradient descent search algorithm is already successfully being used on a rapid-prototyping real-time system. For example, it was employed—similar to the one proposed in this manuscript—in previous research on path planning tasks [4]. This gradient descent optimization has been implemented and successfully tested on a dSpace real-time control system with a sample time of 100 ms. The gradient descent algorithm has a fixed number of maximum iterations so as to guarantee the real-time constraint. Even if the global optimum is not found, the suboptimal solution will be better than the initial start solution. Here, the implemented NG algorithm for the observer framework is tailored for the nonlinear moving horizon estimator to meet the maximum flexibility in the calculation of the gradient, and the objective function calculation. That is, it is possible for the user to modify the calculation in Modelica with replaceable function pointers, without modifying the optimization algorithm itself. This will be shown later in an example in Section 3.2.

Algorithm 2. Nonlinear gradient search algorithm for MHE

```
Set $j = 0$ ($j \in \mathbb{N}^+$) and $\xi_k^0 = \left(x_{\mathrm{int},k-M}^T, \ldots, x_{\mathrm{int},k}^T\right)^T$
while $\left|g\left(\xi_k^j\right) - g\left(\xi_k^{j+1}\right)\right| > 10 \cdot \epsilon$ do
    if unconstrained then
        $r_j = -\nabla g\left(\xi_k^j\right)$
    elseif constrained then
        $r_{\mathrm{pro},j} = -P^{\dagger}(\mathbf{A}) \cdot \nabla g\left(\xi_k^j\right)$
        $r_{\mathrm{f},j} = r_{\mathrm{pro},j} - \left(G_{\mathrm{pa},i} r_{\mathrm{pro},j}\right) G_{\mathrm{pa},i}^T$
    end if
    Determine step size via line search: $\eta_j = \underset{0 \le \eta}{\operatorname{argmin}}\, g\left(\xi_k^j + \eta \cdot r_j\right)$
    Optimization step: $\xi_k^{j+1} = \xi_k^j + \eta_j r_j$
    $j = j + 1$
end while
```

In step 1 (line 1), an initial solution, ξ_k^0, is needed. Well-proven strategies for its calculation are an open loop integration of the prediction model from x_{k-M} to x_k, or a left shift of the last optimization vector $\boldsymbol{\xi}_{\boldsymbol{k-1}}$ and appending an open loop integration from x_{k-1} to x_k. In the unconstrained (lines 3–4) case of the second step, the gradient $\nabla g\left(\xi_k^j\right)$ of the descent direction can be directly calculated, as shown in Algorithm 3.

Also, if equality constraints (lines 5–7) should be incorporated, $A \cdot \xi = b$, the descent direction must be projected on these by means of the Moore–Penrose pseudoinverse [25] $r_{\mathrm{pro},j} = P^{\dagger}(\mathbf{A}) \cdot \nabla g\left(\xi_k^j\right)$. In most of the MHE applications, this step can be neglected (i.e., $P^{\dagger}(\mathbf{A}) = I$), as only inequality constraints are of mayor interest in order to limit the boundaries of the states. To guarantee that the descent direction does not violate the inequality constraints, a set of possible active constraints, $G_{\mathrm{pa},i}$, in the null space of the linear constraints must be determined. The descent direction is projected on the active constraints $\left(G_{\mathrm{pa},i} r_{\mathrm{pro},j}\right) G_{\mathrm{pa},i}^T$. More algorithm implementation details can be found in the literature [24].

Algorithm 3. MHE calculation by means of functional mockup unit (FMU) evaluations

```
Define: $\boldsymbol{R}^* = \left(\boldsymbol{R} \cdot \boldsymbol{R}^T\right)^{-1}$ and $\boldsymbol{Q}^* = \left(\boldsymbol{Q} \cdot \boldsymbol{Q}^T\right)^{-1}$
$\nabla g_{1:n} = \left(\boldsymbol{I}_{k-M}^{+} + (\boldsymbol{I}_{k-M}^{+})^T\right)\left(\boldsymbol{x}_{k-M} - \hat{\boldsymbol{x}}_{k-M}^{+}\right) - 2\frac{\partial h(\boldsymbol{x}_{k-M})}{\partial \boldsymbol{x}_{k-M}}^T \boldsymbol{R}^*\left(\boldsymbol{y}_{k-M}^{\mathrm{m}} - \boldsymbol{h}(\boldsymbol{x}_{k-M})\right)$
for $i = k - M + 1$ to $k$ do
    $\nabla g_{1+(i-k+M)\cdot n:(i-k+M+1)\cdot n} = \frac{\partial g(\xi_k)}{\partial \boldsymbol{x}_i} = 2\boldsymbol{Q}^*\left(\boldsymbol{x}_i - \boldsymbol{x}_{\mathrm{int},i}\right) - 2\frac{\partial \boldsymbol{h}(\boldsymbol{x}_i)^T}{\partial \boldsymbol{x}_i}\boldsymbol{R}^*\left(\boldsymbol{y}_i^{\mathrm{m}} - \boldsymbol{h}(\boldsymbol{x}_i)\right)$
end for
```

In step 3 (line 9), an iterative free line search via quadratic approximation by a second order Taylor series polynomial is performed (Equation (11)). Unfortunately, the derivatives of $g\left(\xi_k^j\right)$ must be calculated via numerical differences, as the extended FMU 2.0 co-simulation interface only supports directional derivatives with respect to the system states and inputs (see [6]—Section 2.1.9). Note that the necessary determination of $g\left(\xi_k^j + \Delta s\right)$ in the differential quotient causes only evaluations of the algebraic output equations of the FMI (cf. Equation (9)), as the calculation of the initial guess $\boldsymbol{x}_{\mathrm{int},i}$ is only performed once in this approach.

$$
\begin{aligned}
g\left(\xi_k^j + \eta \cdot \boldsymbol{r}_j\right) &\approx G(\eta) g\left(\xi_k^j\right) + g'\left(\xi_k^j\right)\eta + \frac{1}{2} g''\left(\xi_k^j\right)\eta^2 \\
\frac{\partial G(\eta)}{\partial \eta} &\overset{!}{=} 0 = g'\left(\xi_k^j\right) + g''\left(\xi_k^j\right)\eta, \\
\Rightarrow \eta_j &= -\frac{g'\left(\xi_k^j\right)}{g''\left(\xi_k^j\right)}
\end{aligned}
\tag{11}
$$

In step 4 (line 10), the actual optimization step, ξ_k^{j+1}, is computed. Finally, in the last step, it is checked whether the stop criterion has been reached. This can be, on the one hand, with a criterion that controls whether the change in the last iteration, j, is small $\left|g\left(\xi_k^j\right) - g\left(\xi_k^{j+1}\right)\right| < 10 \cdot \epsilon$, and, on the other hand, a real-time constraint that stops the search in order to guarantee the cycle-time of the real-time system. In this case, it is assumed that the initial guess (ξ_k^0), calculated by the high fidelity Modelica model, has been already a valid suboptimal solution, and the iterations of the NG did further improve it (cf. Algorithm 2), before the stop (ξ_k^j).

3.2. Moving Horizon Estimation Algorithm Extensions

Here, for the analyzed ESTM MHE observer, different extensions are discussed, in comparison to the nominal MHE algorithm formulation in Section 3. First, a computationally reliable method is introduced for the calculation of the prediction model and the observer constraints, by means of two multi-rate extended FMUs 2.0 for co-simulation (cf. [2,6,26]). It enables enlarging the sampling time of the observer, inspite of the fact that the constraint calculation needs to be executed in a 50 times faster sampling rate, which gives large benefits to the real-time capability. Second, advanced methods for the coupling of discrete optimization variables are proposed. These enforce the physical coupling of the discrete tuner states (ξ_k) to overcome an unrealistic solution of the state trajectory. Third, a heuristic method is discussed to prevent optimization freezing through intelligent recalculation of the reference trajectory ($\boldsymbol{x}_{\mathrm{int}}$) in segments where no measurements are available.

3.2.1. Constraint Evaluation with a Multi-Rate FMU Model Splitting Concept

The first implementations of the constrained observer were based on a single prediction FMU that combined the ESTM prediction model (Section 3) and the boundary constraint evaluation (Section 2.2). To separate the estimated states from the state of the constraint evaluation (the path parameter s), Modelica's logical vector indexing feature was used. However, a simulation experiment analysis showed that in this configuration, it is necessary to run the whole estimator with a fast sampling rate of $T_s = 4$ ms. This is due to the fast dynamics of the control loop to determine the current path parameter (s) in the constraint calculation module (cf. Section 2.2). To overcome this issue, the ESTM and the boundary constraint (BC) model were split into two separate FMUs with different sample times ($T_s^{\text{ESTM}} =$ 200 ms, $T_s^{\text{BC}} = 4$ ms). The connection of the multi-rate FMUs and the estimation algorithm are sketched in Figure 10. Through the model splitting, the states of the ESTM FMU ($\boldsymbol{x} = \{\beta^{\text{C}}, v^{\text{C}}, \dot{\psi}^{\text{C}}, \psi_{\text{C}}, x_{\text{C}}, y_{\text{C}}\}$) are the inputs of the boundary constraints of FMU, which only has one state, the corresponding path parameter s. The sample time of the constrained FMU is 50 times higher than the one of the ESTM, so as to guarantee numerical stability. $\boldsymbol{Z}$ denotes the permutation matrix between the inputs of the BC FMU and the states of the ESTM FMU.

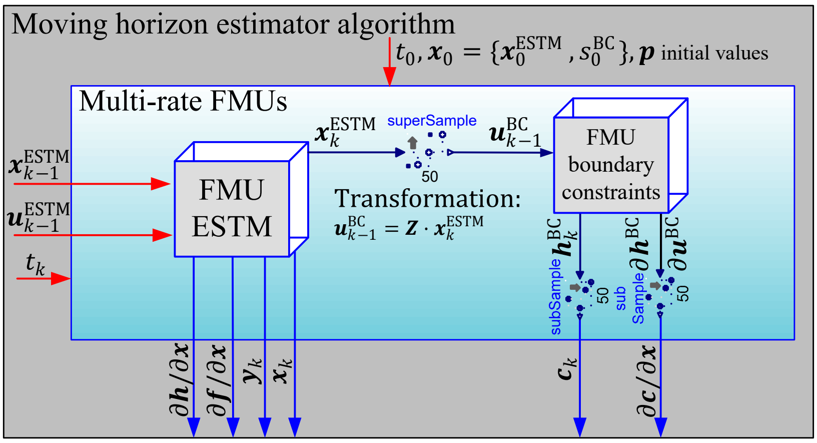

Moving horizon estimator algorithm
$t_0, \boldsymbol{x}_0 = \{\boldsymbol{x}_0^{\text{ESTM}}, s_0^{\text{BC}}\}, \boldsymbol{p}$ initial values
Multi-rate FMUs
$\boldsymbol{x}_{k-1}^{\text{ESTM}}$
$\boldsymbol{u}_{k-1}^{\text{ESTM}}$
t_k
FMU ESTM
$\boldsymbol{x}_k^{\text{ESTM}}$
superSample
50
$\boldsymbol{u}_{k-1}^{\text{BC}}$
FMU boundary constraints
Transformation:
$\boldsymbol{u}_{k-1}^{\text{BC}} = \boldsymbol{Z} \cdot \boldsymbol{x}_k^{\text{ESTM}}$
$\boldsymbol{h}_k^{\text{BC}}$
$\partial \boldsymbol{h}^{\text{BC}}$
$\partial \boldsymbol{u}^{\text{BC}}$
subSample
50
sub Sample
50
$\partial \boldsymbol{h}/\partial \boldsymbol{x}$
$\partial \boldsymbol{f}/\partial \boldsymbol{x}$
$\boldsymbol{y}_k$
$\boldsymbol{x}_k$
$\boldsymbol{c}_k$
$\partial \boldsymbol{c}/\partial \boldsymbol{x}$

Figure 10. Two encapsulated multi-rate functional mockup units (FMUs) as one prediction model.

With this implementation, all of the necessary quantities for the constrained MHE are nested within the multi-rate FMU block, whose interfaces to the outside (denoted with blue arrows) are the same as if no inner model separation was performed. This gives a large benefit in the matter of computational effort, not only caused by the larger integration step, but it also enables the possibility to calculate the constraint only if it is necessary for the estimation algorithm. This benefit is shown in the following simplified flow diagram of the extended MHE algorithm proposed here (see Figure 11). By means of a forward integration in step 1 from t_{k-M} to t_k, it is checked whether any constraint may be potentially activated (c^{a}) and if so, the constraints are linearized along the open loop state trajectory. In step 2, the nonlinear gradient algorithm performs the optimization over the estimation window, incorporating the linearized constraints if necessary. In the last step, step 3, the Kalman filter is updated with the consideration of the system constraints to guarantee that the initial state of the moving window in the next iteration step lies within the feasible region.

Qualitatively, the incorporation of the constraints is depicted in Figure 12. The feasible region of the constraint is limited through the function of c_1 and c_2 (orange-dotted). In this example, the state propagation of the Kalman filter (step 3) causes a violation of c_2, and therefore, the a posteriori propagated state must be constrained by a method (e.g., state constraint projection in Appendix B. The boundary constraints control (step 1) now starts with the corrected a posteriori estimate, $\hat{\boldsymbol{x}}_{k-M}^{+\text{cstr}}$, and detects a constraint violation between the third and fourth sample point. This is only possible as the BC model is integrated fifty times between every ESTM evaluation and correction step. To guarantee that the initial solution of the NG solver lies in the feasible region, the $\boldsymbol{x}_{\text{int}}^{\text{cstr}}$ is limited via the simplified Newton descent search, as described in Appendix B.

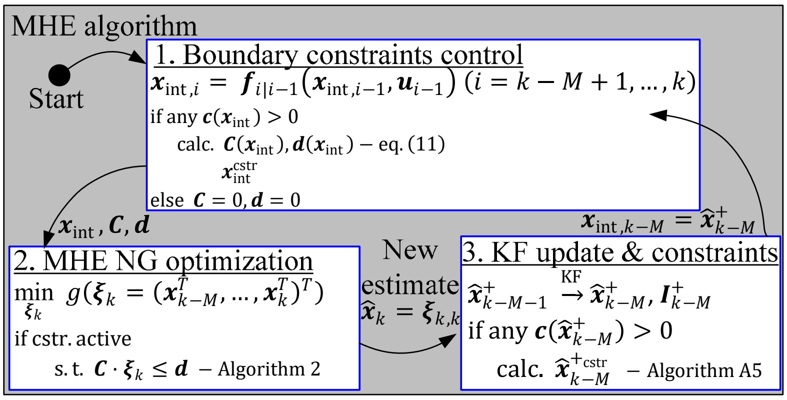

MHE algorithm
Start
1. Boundary constraints control
$\boldsymbol{x}_{\text{int},i} = \boldsymbol{f}_{i|i-1}(\boldsymbol{x}_{\text{int},i-1}, \boldsymbol{u}_{i-1})\ (i = k - M + 1, \ldots, k)$
if any $\boldsymbol{c}(\boldsymbol{x}_{\text{int}}) > 0$
calc. $\boldsymbol{C}(\boldsymbol{x}_{\text{int}}), \boldsymbol{d}(\boldsymbol{x}_{\text{int}})$ – eq. (11)
$\boldsymbol{x}_{\text{int}}^{\text{cstr}}$
else $\boldsymbol{C} = 0, \boldsymbol{d} = 0$
$\boldsymbol{x}_{\text{int}}, \boldsymbol{C}, \boldsymbol{d}$
$\boldsymbol{x}_{\text{int},k-M} = \hat{\boldsymbol{x}}_{k-M}^{+}$
2. MHE NG optimization
$\min_{\boldsymbol{\xi}_k}\ g(\boldsymbol{\xi}_k = (\boldsymbol{x}_{k-M}^{T}, \ldots, \boldsymbol{x}_k^{T})^{T})$
if cstr. active
s. t. $\boldsymbol{C} \cdot \boldsymbol{\xi}_k \leq \boldsymbol{d}$ – Algorithm 2
New estimate
$\hat{\boldsymbol{x}}_k = \boldsymbol{\xi}_{k,k}$
3. KF update & constraints
$\hat{\boldsymbol{x}}_{k-M-1}^{+} \xrightarrow{\text{KF}} \hat{\boldsymbol{x}}_{k-M}^{+}, \boldsymbol{I}_{k-M}^{+}$
if any $\boldsymbol{c}(\hat{\boldsymbol{x}}_{k-M}^{+}) > 0$
calc. $\hat{\boldsymbol{x}}_{k-M}^{+\text{cstr}}$ – Algorithm A5

Figure 11. Flowchart of the extended moving horizon estimator.

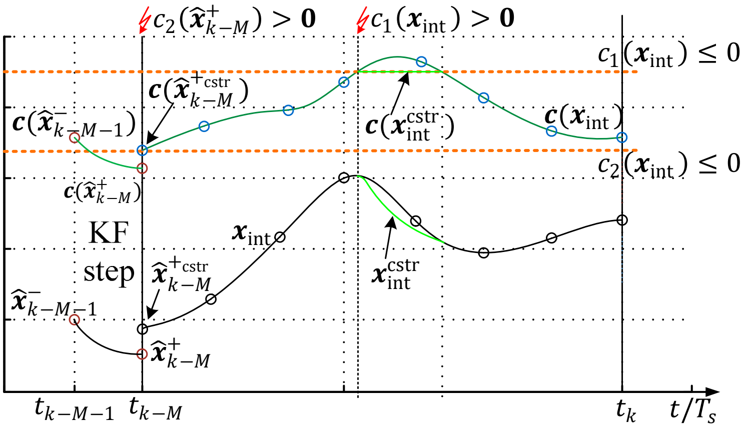

$c_2(\hat{\boldsymbol{x}}_{k-M}^{+}) > \boldsymbol{0}$
$c_1(\boldsymbol{x}_{\text{int}}) > \boldsymbol{0}$
$c_1(\boldsymbol{x}_{\text{int}}) \leq 0$
$\boldsymbol{c}(\hat{\boldsymbol{x}}_{k-M}^{+\text{cstr}})$
$\boldsymbol{c}(\hat{\boldsymbol{x}}_{k-M-1}^{-})$
$\boldsymbol{c}(\boldsymbol{x}_{\text{int}}^{\text{cstr}})$
$\boldsymbol{c}(\boldsymbol{x}_{\text{int}})$
$c_2(\boldsymbol{x}_{\text{int}}) \leq 0$
$\boldsymbol{c}(\hat{\boldsymbol{x}}_{k-M}^{+})$
KF
step
$\boldsymbol{x}_{\text{int}}$
$\hat{\boldsymbol{x}}_{k-M}^{+\text{cstr}}$
$\boldsymbol{x}_{\text{int}}^{\text{cstr}}$
$\hat{\boldsymbol{x}}_{k-M-1}^{-}$
$\hat{\boldsymbol{x}}_{k-M}^{+}$
t_{k-M-1}
t_{k-M}
t_k
t/T_s

Figure 12. Constraint violation detection and handling within the moving window.

3.2.2. Multiple Shooting Inspired Optimization Objective Extension

In the original MHE problem formulation (Equation (8) to Equation (10)), the discrete system states within the moving window $\xi_k = \left(x_{k-M}^T, x_{k-M+1}^T, \ldots, x_k^T\right)^T$ are not coupled with each other between the sample points (t_k). This implies that the optimization algorithm does not have any information about the dynamic behavior of the ESTM prediction model between the sample points within the estimation window. In the case of the ESTM, with a large sample time $T_s = 200$ ms, this may lead to a physically unfeasible set ξ_k, which however minimizes the optimization criteria. A consideration to overcome this weak point is the introduction of coupling penalty terms between the time instances in the minimization criterion in Equations (9) and (10). Figure 13 exemplifies the approach developed here, namely: the initial open loop integration from the time instance t_{k-M} to the current time instance t_k is denoted as x_{int}^0. The set of optimized state vectors ξ_k^j in the j-th NG descent step (cf. Algorithm 2) is marked with green circles.

The temporal evolution from these discrete states by means of the FMU yields a set of system states $x_{\text{int},i}^{\text{L}}$ denoted with a red circle, as follows:

$$x_{\text{int},i}^{\text{L}} = f_{i|i-1}(x_{i-1}, \boldsymbol{u}_{i-1}), (i = k - M, \ldots, k) \tag{12}$$

In the depicted qualitative example (cf. Figure 13), one can see, that through the evolution of the optimization process a displacement $x_{\text{int},i}^{\text{L}} \neq x_p$ is caused in the j-th iteration step of the NG algorithm (see Algorithm 2). To minimize this gap, the MHE optimization objective is changed to the following:

$$\begin{aligned} g(\xi_k) =& \| \boldsymbol{x}_{k-M} - \hat{\boldsymbol{x}}^{+}_{k-M} \|^2_{\boldsymbol{I}^{+}_{k-M}} \\ &+ \textstyle\sum_{i=k-M}^{k} \| \boldsymbol{y}^{\mathrm{m}}_i - \boldsymbol{h}(\boldsymbol{x}_i) \|^2_{\boldsymbol{R}^{-1}} \\ &+ \textstyle\sum_{i=k-M+1}^{k} \| \boldsymbol{x}_i - \boldsymbol{x}_{\mathrm{int},i} \|^2_{\boldsymbol{Q}^{-1}} \\ &\underbrace{+ \sum_{i=k-M}^{k} \| \boldsymbol{x}_i - \boldsymbol{x}^{\mathrm{L}}_{\mathrm{int},i} \|^2_{\boldsymbol{Q}_{\mathrm{MS}}}}_{\text{Additional penalty}} \end{aligned} \tag{13}$$

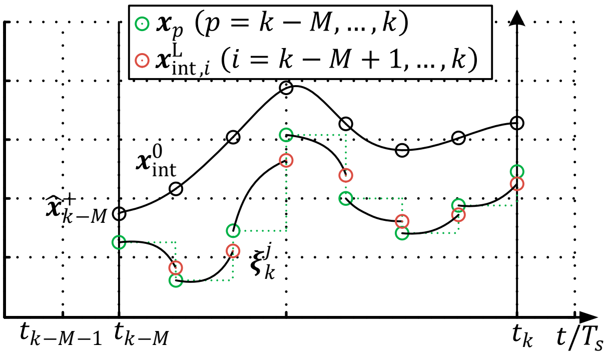

○ $\boldsymbol{x}_p$ $(p = k-M, \ldots, k)$
○ $\boldsymbol{x}^{\mathrm{L}}_{\mathrm{int},i}$ $(i = k-M+1, \ldots, k)$
$\boldsymbol{x}^{0}_{\mathrm{int}}$
$\hat{\boldsymbol{x}}^{+}_{k-M}$
$\boldsymbol{\xi}^{j}_{k}$
t_{k-M-1}
t_{k-M}
t_k
t/T_s

Figure 13. Schematic representation of a moving measurement window.

In comparison to the original formulation, the quality functional is extended with an additional weighted least squares expression so as to enforce a stronger coupling of the piecewise integration ($\boldsymbol{x}^{\mathrm{L}}_{\mathrm{int},i}$) and the optimized stated vector ($\boldsymbol{x}_i$) by means of the user tunable weighting matrix ($\boldsymbol{Q}_{\mathrm{MS}}$).

Besides performing the integration ($\boldsymbol{x}^{\mathrm{L}}_{\mathrm{int},i}$) by means of the FMU (Equation (12)), it is necessary to approximate the integration rule for the gradient $g(\xi_k)$ calculation. It is proposed that it is more important to generate a good approximated descent direction for the optimizer than the exact reproduction of the integration method $\boldsymbol{x}^{\mathrm{L}}_{\mathrm{int},i}$ used in the FMU.

In the simplest case, this is achieved by the consideration of the directional derivative at the past instance (later called V1). In the second version (V2), the integrator is approximated as an Euler 1 integration rule. The last optimization variable coupling approximation is formulated by a trapezoid integration rule (V3), as follows:

$$\begin{aligned} \mathrm{V2} \rightarrow \boldsymbol{x}^{\mathrm{L}}_{\mathrm{E1},i} &= \boldsymbol{f}_{i|i-1}(\boldsymbol{x}_{i-1}, \boldsymbol{u}_{i-1}) \approx\; \boldsymbol{x}_{i-1} + \boldsymbol{f}_{i-1} \cdot T_s \\ \mathrm{V3} \rightarrow \boldsymbol{x}^{\mathrm{L}}_{\mathrm{Tr},i} &= \boldsymbol{f}_{i|i-1}(\boldsymbol{x}_{i-1}, \boldsymbol{u}_{i-1}) \approx\; \boldsymbol{x}_{i-1} + \tfrac{1}{2} \cdot (\boldsymbol{f}_{i-1} + \boldsymbol{f}_i) \cdot T_s \end{aligned} \tag{14}$$

With these three versions, the complete extended MHE gradient computation is given in Algorithm 4. Comparing this gradient calculation to the original formulation in Algorithm 3, the main difference is the additional "for loop" with the index j, in which the operator ± denotes that all of the values are additively added to the existing entries from the earlier loop.

Algorithm 4. Multiple shooting MHE gradient calculation

```
Define: $\boldsymbol{R}^* = (\boldsymbol{R}\cdot\boldsymbol{R})^{-1}$; $\boldsymbol{Q}^* = (\boldsymbol{Q}\cdot\boldsymbol{Q})^{-1}$; $\boldsymbol{E} = \text{eye}(n)$
Define: $i_1(i) := 1 + (i-k+M)n : (i-k+M+1)n$
$\nabla g_{1:n} = \left(\boldsymbol{I}^+_{k-M} + (\boldsymbol{I}^+_{k-M})^T\right)\left(\boldsymbol{x}_{k-M} - \hat{\boldsymbol{x}}^+_{k-M}\right) - 2\frac{\partial \boldsymbol{h}(\boldsymbol{x}_{k-M})}{\partial \boldsymbol{x}_{k-M}}^T \boldsymbol{R}^*\left(\boldsymbol{y}^m_{k-M} - \boldsymbol{h}(\boldsymbol{x}_{k-M})\right)$
for $i = k-M+1$ to $k$ do
        $\nabla g_{i_1(i)} = \frac{\partial g(\xi_k)}{\partial \boldsymbol{x}_i} = 2\boldsymbol{Q}^*\left(\boldsymbol{x}_i - \boldsymbol{x}_{\text{int},i}\right) - 2\frac{\partial \boldsymbol{h}(\boldsymbol{x}_i)}{\partial \boldsymbol{x}_i}^T \boldsymbol{R}^*\left(\boldsymbol{y}^{\text{m}}_i - \boldsymbol{h}(\boldsymbol{x}_i)\right)$
        if version == V3 then
                $\nabla g_{i_1(i)} -= \left(\frac{1}{2}\cdot T_s \cdot \frac{\partial f_i}{\partial x_i}\right)^T \cdot 2 \cdot \boldsymbol{Q}_{\text{MS}} \cdot \left(\boldsymbol{x}_i - \boldsymbol{x}^{\text{L}}_{\text{int},i}\right)$
        end if
end for
Define: $j_1(j) := 1 + (j-k+M)n : (j-k+M+1)n$
for $j = k-M$ to $k-1$ do
    $\nabla g_{j_1(j)} += \frac{\partial g(\xi_k)}{\partial \boldsymbol{x}_j}$
    switch(version)
        case V1:
            $\nabla g_{j_1(j)} += -\frac{\partial f_j}{\partial \boldsymbol{x}_j} \cdot 2 \cdot \boldsymbol{Q}_{\text{MS}} \cdot \left(\boldsymbol{x}_{j+1} - \boldsymbol{x}^{\text{L}}_{\text{int},j+1}\right)$
        case V2:
            $\nabla g_{j_1(j)} += -\left(\frac{1}{2}\cdot T_s \cdot \frac{\partial f_j}{\partial x_j}\right)^T \cdot 2 \cdot \boldsymbol{Q}_{\text{MS}} \cdot \left(\boldsymbol{x}_{j+1} - \boldsymbol{x}^{\text{L}}_{\text{int},j+1}\right)$
        case V3:
            $\nabla g_{j_1(j)} -= \left(\boldsymbol{E} + \frac{1}{2}\cdot T_s \cdot \frac{\partial f_j}{\partial \boldsymbol{x}_j}\right)^T \cdot 2 \cdot \boldsymbol{Q}_{\text{MS}} \cdot \left(\boldsymbol{x}_{j+1} - \boldsymbol{x}^{\text{L}}_{\text{int},j+1}\right)$
    end switch
end for
```

3.2.3. Multi-Rate, Triggered, or Delayed Measurements in MHE

For the nonlinear MHE approach of Section 3, it is proposed to keep the lagged sensor data in a ring buffer, to interconnect them with past measurements of non-lagged sensors, and to index the active sensors at the particular time instance. Comparing this with the formulation of Equation (9), all of the expressions that are connected with the measured output (the middle part) have to be calculated separately in each time step so as to incorporate the changing amount of available sensor information at the dedicated time step.

$$
\begin{aligned}
g(\xi_k) &= \cdots + \sum_{i=k-M}^{k} \parallel \boldsymbol{y}^{\text{m}}_{i,\text{a}} - \boldsymbol{h}_{\text{a}}(\boldsymbol{x}_i) \parallel^2_{\boldsymbol{R}_k^{-1}} + \cdots \\
\overline{\sigma}_{r_\text{a}} &= \left\{\sigma_{r_i} \middle| i \in \boldsymbol{y}_k\right\} \\
\boldsymbol{R}_k &= \text{diag}(\overline{\sigma}_{r_\text{a}}) \\
\boldsymbol{h}_{\text{a}} &= \left\{\boldsymbol{h}_i \middle| i \in \boldsymbol{y}_k\right\} \\
\boldsymbol{y}^{\text{m}}_{\text{a}} &= \begin{bmatrix} y^{\text{m}}_{k-M,1} & \cdots & y^{\text{m}}_{k,1} \\ \vdots & \ldots & \vdots \\ y^{\text{m}}_{k-M,n_y} & \cdots & y^{\text{m}}_{k,n_y} \end{bmatrix}
\end{aligned}
\qquad (15)
$$

In Equation (15), all of the active measurements are denoted with an additional subscript $((\cdot)_\text{a})$ for the active sensors. In the construction of the active measurement matrix ($\boldsymbol{y}^{\text{m}}_{\text{a}}$), the time-delays of the single sensors are already considered. To cope with the varying dimensions in the optimization problem, the implementation could utilize the vector indexing feature in the Modelica language. The same technique is used in the Kalman filter propagation (cf. Algorithm 1) for the reduced indexed measurements. This procedure is valid according to the Kalman theory, and can be seen in an analogy to sequential Kalman filtering (see [10] – Section 6.1). In these implementations, the matrix vector notation of the Kalman filter algorithm is replaced by sequentially solving a scalar problem for each measurement. Therefore, more sensor information can improve the estimation in the sense of minimizing the covariance, $\boldsymbol{P}^+_{k-M}$. In the case that less sensor information is available, larger values of $\det(\boldsymbol{P}^+_{k-M})$ occur, but the validity of the Kalman theory still holds. In the same manner, the dimension of the available measurements at t_{k-M} for the calculation of the information matrix $\boldsymbol{I}^+_{k-M}$ and the initial guess $\hat{\boldsymbol{x}}^+_{k-M}$ also needs to be considered in the Kalman step.

3.2.4. Adaptive Initial Reference Refreshing for Delayed Measurements

In Section 3.2.3, a theory extension to MHE is given that enables the assignment of measurements to their particular time instance by intelligent measurement storage and temporal activation indexing. In Figure 14, the measurement signal $\boldsymbol{y}^{\mathrm{m}}_{\mathrm{a}}$ is schematically sketched, which is only available at the time instances highlighted with a yellow flash. For example, between time instances t_{k-M+1} and t_{k-M+3}, no new measurement information is available.

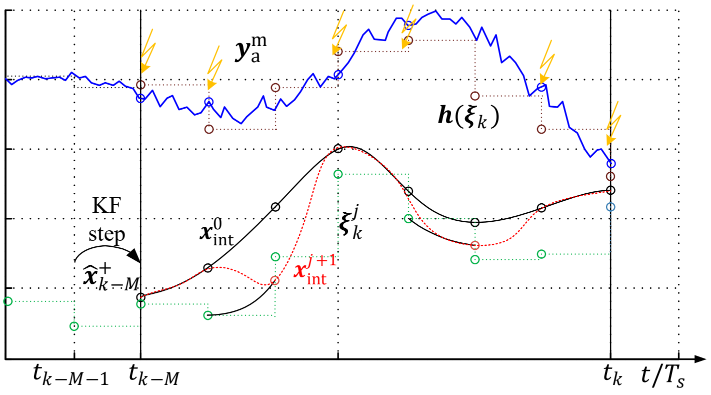

$\boldsymbol{y}^{\mathrm{m}}_{\mathrm{a}}$
$\boldsymbol{h}(\boldsymbol{\xi}_k)$
KF
step
$\boldsymbol{x}^0_{\mathrm{int}}$
$\boldsymbol{\xi}^j_k$
$\hat{\boldsymbol{x}}^+_{k-M}$
$\boldsymbol{x}^{j+1}_{\mathrm{int}}$
t_{k-M-1}
t_{k-M}
t_k
t/T_s

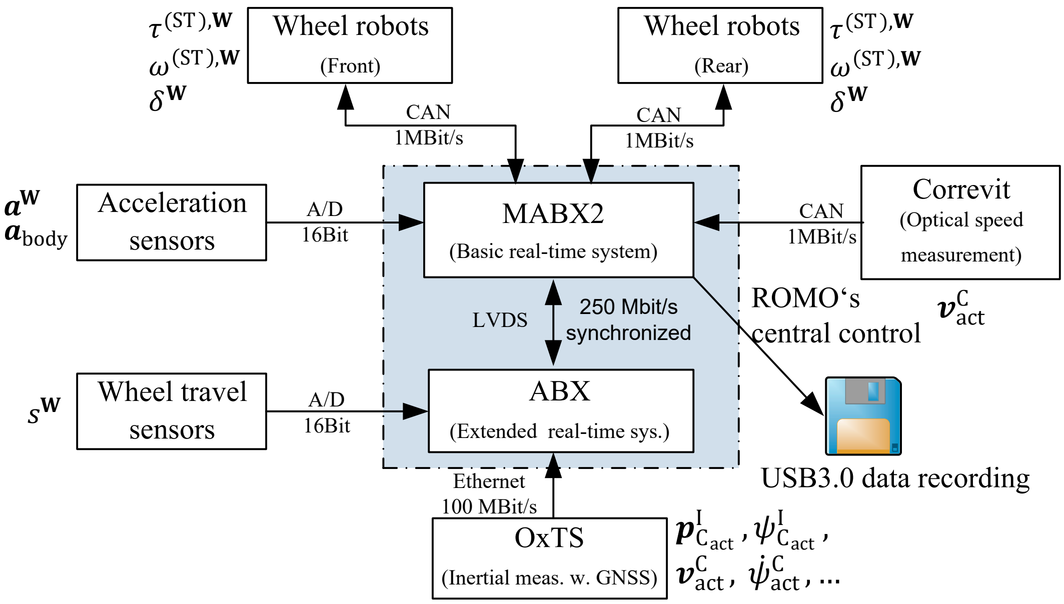
Figure 14. Moving horizon window with fragmentary measurements.

Different algorithm experiments have shown that this may force the NG optimizer to tune the variable $\xi_{k,3}$ towards the initial guess of the open loop state trajectory (x_{int}). To overcome this very conservative solution, a heuristic method is introduced in Equation (16) to refresh $\boldsymbol{x}^{j+1}_{\mathrm{int}}$ after the optimization step (line 10 in Algorithm 2). It determines the gaps in the logical vector indexing matrix of the active measurements $\boldsymbol{y}^{\mathrm{m}}_{\mathrm{index_a}}$ and integrates from the last time instance where all of the measurements are available, as follows:

$$\boldsymbol{x}^{j+1}_{\mathrm{int},i} = \begin{cases} \boldsymbol{x}^{0}_{\mathrm{int},i}, \ \forall \left\{ i \middle| \left(i \in \boldsymbol{y}^{\mathrm{m}}_{\mathrm{a}}\right) \wedge \left(i+1 \in \boldsymbol{y}^{\mathrm{m}}_{\mathrm{a}}\right) \right\} \\ \boldsymbol{f}_{i|i-1}\left(\boldsymbol{\xi}^{j}_{k,i-1}, \boldsymbol{u}_{i-1}\right), \ \forall \left\{ i \middle| \left(i \in \boldsymbol{y}^{\mathrm{m}}_{\mathrm{a}}\right) \wedge \left(i+1 \notin \boldsymbol{y}^{\mathrm{m}}_{\mathrm{a}}\right) \right\}, \ (i = k-M, \ldots, k) \\ \boldsymbol{f}_{i|i-1}\left(\boldsymbol{x}^{j+1}_{\mathrm{int},i-1}, \boldsymbol{u}_{i-1}\right), \forall \left\{ i \middle| \left(i \notin \boldsymbol{y}^{\mathrm{m}}_{\mathrm{a}}\right) \wedge \left(i+1 \notin \boldsymbol{y}^{\mathrm{m}}_{\mathrm{a}}\right) \right\} \end{cases} \tag{16}$$

4. Experimental Results

The following experimental results rely on the test measurement data that were recorded during the driving experiments using ROboMObil, as described in Section 2. Two real-time rapid prototyping systems were used to record all of the variables from the wheel robot controllers as well as the senor system of ROboMObil. The arrangement, shown in Figure 15, is designed to meet the robotics inspired central control architecture with two synchronized rapid prototyping controllers (dSPACE MABX2 and ABX), which represent ROboMObil's central control unit. The wheel robots with the traction and steering drive torques ($\tau^{(\mathrm{ST}),\mathbf{W}}$), angular velocities ($\omega^{(\mathrm{ST}),\mathbf{W}}$), and steering angles ($\delta^{\mathbf{W}}$) quantities, as well as the Correvit optical measurement unit (used to measure the vehicle's longitudinal and lateral speed $v^{\mathrm{C}}_{\mathrm{act}} = \left\{v^{\mathrm{C}}_{\mathrm{act}_x}, v^{\mathrm{C}}_{\mathrm{act}_y}\right\}$), are connected via high-speed controller area network (CAN) busses. The inertial measurement system with a differential corrected Global Navigation Satellite System (GNSS) to measure the vehicle states, such as the positon $\boldsymbol{p}^{\mathrm{I}}_{\mathrm{C}_{\mathrm{act}}}$, yaw angle $\psi^{\mathrm{I}}_{\mathrm{C}_{\mathrm{act}}}$ and yaw rate $\dot{\psi}^{\mathrm{I}}_{\mathrm{C}_{\mathrm{act}}}$, or the velocity $v^{\mathrm{C}}_{\mathrm{act}}$, is wired to the central control by an Ethernet bus. Additionally, the wheel ($\boldsymbol{a}^{\mathbf{W}}$) and chassis body ($\boldsymbol{a}_{\mathrm{body}}$) accelerations, as well as wheel travel sensors (s^{W}) are tethered via analog inputs. Both the inertial measurement unit from Oxford Technical Solutions Ltd. (OxTS) measurement unit and the optical odometry sensor are suited for the experimental evaluation of the driving experiments.



$\tau^{(\mathrm{ST}),\mathbf{W}}$
$\omega^{(\mathrm{ST}),\mathbf{W}}$
$\delta^{\mathbf{W}}$
Wheel robots
(Front)
Wheel robots
(Rear)
$\tau^{(\mathrm{ST}),\mathbf{W}}$
$\omega^{(\mathrm{ST}),\mathbf{W}}$
$\delta^{\mathbf{W}}$
CAN
1MBit/s
CAN
1MBit/s
$\boldsymbol{a}^{\mathbf{W}}$
$\boldsymbol{a}_{\mathrm{body}}$
Acceleration sensors
A/D
16Bit
MABX2
(Basic real-time system)
CAN
1MBit/s
Correvit
(Optical speed measurement)
LVDS
250 Mbit/s
synchronized
ROMO's central control
$\boldsymbol{v}^{\mathrm{C}}_{\mathrm{act}}$
$s^{\mathbf{W}}$
Wheel travel sensors
A/D
16Bit
ABX
(Extended real-time sys.)
USB3.0 data recording
Ethernet
100 MBit/s
OxTS
(Inertial meas. w. GNSS)
$\boldsymbol{p}^{\mathrm{I}}_{\mathrm{C}_{\mathrm{act}}}, \psi^{\mathrm{I}}_{\mathrm{C}_{\mathrm{act}}},$
$\boldsymbol{v}^{\mathrm{C}}_{\mathrm{act}}, \dot{\psi}^{\mathrm{C}}_{\mathrm{act}}, \ldots$

Figure 15. ROboMObil's vehicle dynamics sensors and actuators architecture. OxTS—inertial measurement unit; ABX—AutoBox (rapid prototyping real-time controller); MABX2—MicroAutoBox II (embedded rapid prototyping real-time controller).

4.1. Observer Configuration

The preliminary observer case studies showed that with the modeling approach chosen here (cf. Section 2.1), in combination with a Runge–Kutta 4 integrator, the vehicle position estimator can be run with a cycle time of $T_s = 200$ ms. Moreover, the usage of the vehicle yaw rate ($\dot{\psi}^{\mathrm{C}}$) can actively contribute to improve the measurement, whereas the incorporation of the vehicle lateral acceleration (a^{C}_y) downgrades the observer performance. The output equation, $a^{\mathrm{C}}_y = v^{\mathrm{C}} \cdot (\dot{\psi}^{\mathrm{C}} + \dot{\beta}^{\mathrm{C}})$, shows clearly that the lateral acceleration is algebraically cross-coupled to the vehicle yaw rate ($\dot{\psi}^{\mathrm{C}}$). In fact, this causes the aforementioned observations, as it is, to the best knowledge of the author, impossible to find a covariance configuration—even by optimization—which makes reasonable use of both the sensor information and overcomes the negative effects of, for example, minimal jittering relative delays between both signals.

In Figure 16, different simulation results of the same vehicle position estimator are given. Again, the road boundaries are marked in a dotted-orange color. The vehicle completed three rounds through the circuit. The red trajectory denotes the open loop result—the ESTM model is simulated with the actual system inputs $\boldsymbol{u}$—and the position is strongly drifting away from the planned trajectory, caused by the evolving position integration error (compare equations for $\mathrm{d}x_{\mathrm{C}}/\mathrm{d}t$, $\mathrm{d}y_{\mathrm{C}}/\mathrm{d}t$—in Equation (A1)). The green and the blue curves denote the results with a single step Kalman filter, under the assumption that no delays are in the measurements.

The best observation results could be achieved by the use of a square root extended Kalman filter (SR-EKF) algorithm (cf. e.g., [2]), whereas the square root unscented Kalman filter (SR-UKF) (cf. e.g., [2]) was less robust. The reason for this is the sigma point propagation through the Pacejka tire model, which evidently leads to a wrong state propagation caused by its high sensitivity around the actual state estimation point.

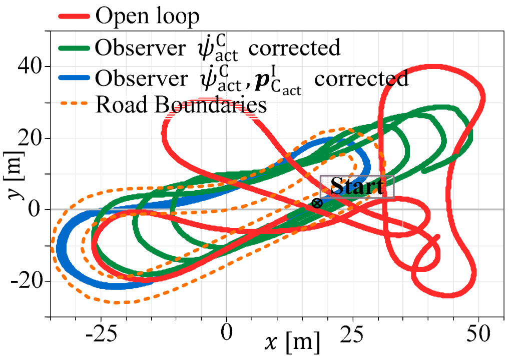


Figure 16. Comparison of different measurement incorporations.

4.2. Moving Horizon Estimation Algorithm Assessment

In this last subSection, the complete MHE ESTM algorithm with its extensions to couple the optimization variables (cf. Section 3.2), refreshment of the initial guess, and efficient road boundary constraint handling are demonstrated (cf. Figure 7). In Table 1, the outcome of a comprehensive simulation case study is summarized. The window length has been set to $M = 4$ and the GPS time delay varies between $n_{\mathrm{d}} = 0 \ldots 3$ steps. All of the parameters for the respective observer configuration have been optimized with the DLR multi-objective parameter synthesis (MOPS) optimizer framework [27] to guarantee comparable results.

Table 1. Comparison of the vehicle position observer performance indicators. SR-EKF—square-root extended Kalman filter; MHE—moving horizon estimation/estimator; MS—multiple shooting; SS—single shooting.

Algorithm	$\overline{T}_{\mathrm{mean}}$	$x_{\mathrm{C_{Fit}}}$	$y_{\mathrm{C_{Fit}}}$	$\psi_{\mathrm{C_{Fit}}}$	$\dot{\psi}^{\mathrm{C}}_{\mathrm{Fit}}$	$v^{\mathrm{C}}_{\mathrm{Fit}}$	$\beta^{\mathrm{C}}_{\mathrm{Fit}}$
$n_{\mathrm{d}} = -$							
Open Loop	6.4	75.5	62.2	96.4	80.2	77.7	81.0
Open Loop Cstr.	6.4	87.7	87.5	96.4	80.2	77.7	81.0
$n_{\mathrm{d}} = 0$							
SR-EKF $\dot{\psi}^{\mathrm{C}}_{\mathrm{act}}$	4.4	78.9	95.2	99.1	95.5	77.3	87.4
SR-EKF $\dot{\psi}^{\mathrm{C}}_{\mathrm{act}}$ Cstr.	4.8	88.2	89.2	99.1	95.5	77.3	87.4
SR-EKF	6.4	97.9	97.2	99.2	93.3	82.4	85.2
MHE SS	67.2	97.3	96.5	99.4	95.5	83.1	81.3
MHE MS V1	46.4	98.5	97.8	99.3	86.9	86.4	80.4
$n_{\mathrm{d}} = 1$							
MHE SS up	78.8	98.2	97.8	99.4	98.4	84.6	80.9
MHE MS up V1	93.6	98.3	97.8	99.4	86.8	85.3	81.0
$n_{\mathrm{d}} = 2$							
MHE SS fix	77.6	98.4	97.8	99.2	98.9	85.9	80.1
MHE SS up	75.2	98.7	98.1	99.2	98.2	85.9	80.3
MHE MS up V1	61.6	98.8	98.2	99.3	86.9	86.4	80.4
MHE MS fix V2	64.8	98.5	98.1	99.2	86.3	86.7	80.0
MHE MS up V2	110.0	98.7	98.2	99.2	86.8	86.7	80.0
MHE MS fix V3	67.2	98.5	98.0	99.2	86.5	86.6	80.3
MHE MS up V3	136.8	98.7	98.3	99.2	86.6	86.6	80.3
$n_{\mathrm{d}} = 3$							
MHE SS up	56.8	97.7	97.1	98.9	97.3	86.8	79.6
MHE MS up V1	79.6	97.7	97.1	99.1	87.9	87.3	80.1

The results in Table 1 are sorted with respect to the number of delayed samples (n_{d}) of the GPS signal. The results that differ tremendously (good or bad) in a group of the same number of delays are highlighted in green or red, respectively. The first column denotes the used observer setup, in which the acronyms for SS (single shooting; standard formulation), MS (multiple shooting), fix (constant initial guess), and up (anti optimization freezing) are used. The second column gives the average execution time per estimation step $\overline{T}_{\text{mean}}$ in [ms] of the experiment executed on a standard 64-bit Windows-based system (Intel i7-4600U 2,7 GHz, 8 GB Ram, SSD), and can be interpreted as a measure for the increase of computational complexity in comparison to the improvement of the estimation. It can be seen that even in the most computationally demanding configuration (MHE MS up V3 $\overline{T}_{\text{mean}} = 136.8$ ms), the algorithm can be executed faster than in real-time ($\overline{T}_{\text{mean}} \ll T_s$). In the third to eighth columns, the percentage goodness of fit (normalized root mean square error between to time series) in comparison to the reference measures of the experiment (cf. Section 2) are summarized.

The first two rows in Table 1 give reference of the ESTM without any observer correction, but in the second row has the roadway constraint incorporation. The next two rows are the first results using an observer that only incorporates the measured vehicle yaw rate ($\dot{\psi}^{\text{C}}_{\text{act}}$). Both still have a large positional deviation, albeit the estimation of the vehicle yaw rate, and the angle and side slip angle are improved in comparison with the open loop tests. The next section highlights three versions of the ESTM observer, with all measurements $\dot{\psi}^{\text{C}}_{\text{act}}$, $\boldsymbol{p}^{\text{I}}_{\text{C}_{\text{act}}}$ available and not delayed. The best results can be achieved with the multiple shooting objective V1. The following two experiments incorporate a delay of one sample step.

Unfortunately, the SR-EKF algorithm could not be stabilized to give a feasible estimate for all of the measures, especially the side slip angle (β^{C}) is heavily oscillating (cf. Figure 17—bottom left). Here, for the first time, the delayed measurement compensation in the MHE formulation can show its advantage in the single shooting as well as in the multiple shooting objective formulations.

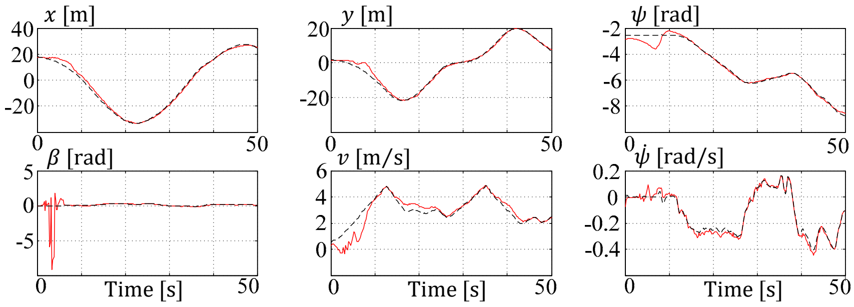


Figure 17. Extended single track model (ESTM) square-root extended Kalman filter (SR-EKF) ($n_{\text{d}} = 1$) setup state estimations (red) vs. reference (black).

The section with $n_{\text{d}} = 2$ is the largest section, which correlates to a delay of 400 ms. All of the modifications introduced in Section 3.2 are tested here for their performance. Even though all of the results are very close to each other, the multiple shooting V1 with reference updating gives the best performance in positioning accuracy and computational reliability (cf. state plots in Figure 18).

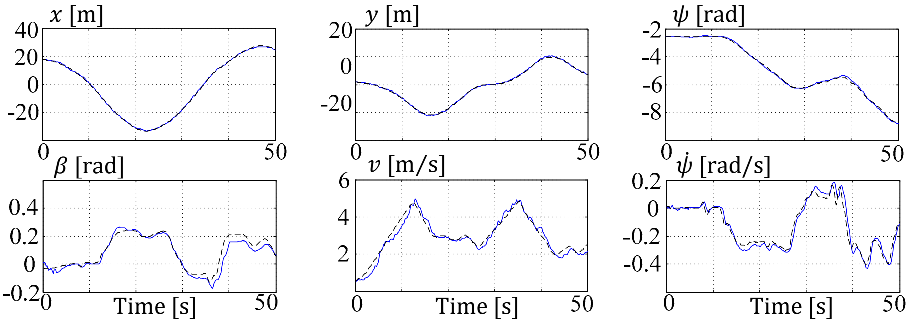


Figure 18. ESTM moving horizon estimation/estimator multiple shooting (MHE MS) up V1 ($n_{\text{d}} = 2$) state estimations (blue) vs. reference (black).

In the last two rows of the simulation study, two configurations with $n_\mathrm{d} = 3$ are assessed. The single shooting (see Figure 19) and multiple shooting do benefit from the proposed update mechanism, and even if the GPS signal is delayed by 600 ms, the results are still reliable.

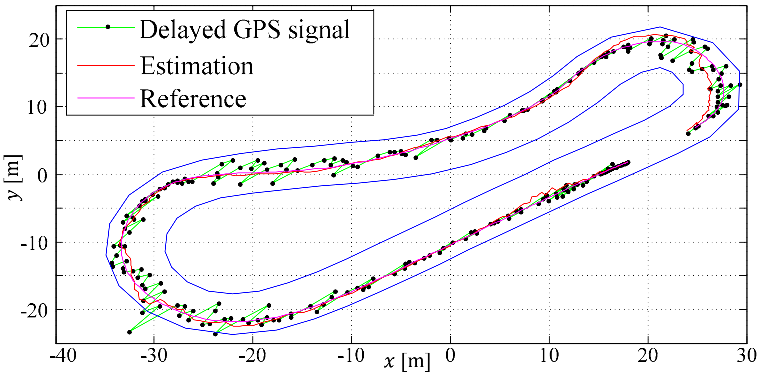


Figure 19. ESTM MHE single shooting (SS) up ($n_\mathrm{d} = 3$): x–y position estimations results.

In total, it can be stated that the time delay incorporation in a MHE is very effective and gives good stability, especially when the sample steps are large. The competitive SR-EKF algorithm failed already with a delay of $n_\mathrm{d} = 1$, although its computational time is up to 20 times lower. The influence of the extension of the objective function with multiple shooting penalties needs to be analyzed from case to case, whereas the anti-freezing feature in the optimizer can be seen as a good improvement to the solution at time instance t_k (cf. Figure 20).

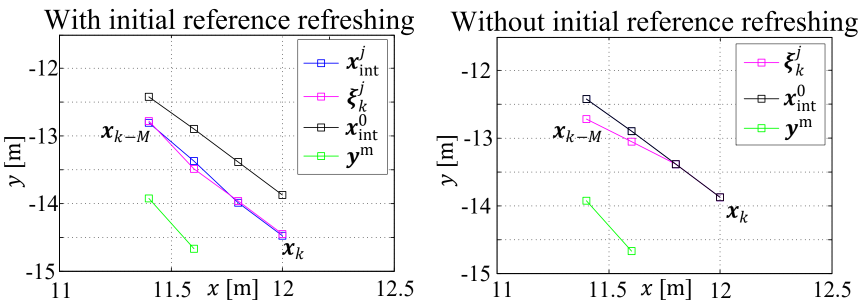


Figure 20. Effectiveness of the anti-freezing heuristic within the optimization window.

Future investigations will be performed with methods other than the NG solver methods, as recently proposed in the literature [28], which are capable of handling nonlinear (in-)equality constraints in real-time. With this extension, it is likely that the multiple shooting approaches might be even more effective.

5. Conclusions

The outcome of the ESTM MHE observer can be summarized as follows:

- A continuous-time Modelica vehicle model with event handling for vehicle standstill (cf. Section 2.1) could be derived (see estimation experiment in Figure 21, starting from standstill and coming back to standstill for about 20 s) and automatically discretized via Dymola, the extended FMU 2.0 for co-simulation technology and the model-based observer framework.
- The derivation of the roadway limit constraint has been designed by extending the principle of the path interpolation introduced in Section 2.2.
- The nominal MHE algorithm of Section 3 was augmented with a multiple shooting formulation in the objective function, a heuristic optimization freezing prevention, a multi-rate model, and a constraint calculation splitting methodology.

- For the experimental investigations, real test data from ROMO was selected, and additionally, the GPS position measures were delayed for the experimental setup.
- All together with the technique for delayed measurements in the MHE application (Section 3.2.3), a comprehensive simulative assessment with the different objective configuration and anti-freezing features, as well as varying delays in comparison to a standard Kalman filter, have been given.
- The proposed estimation approach could achieve a position $x_{\mathrm{C}}, y_{\mathrm{C}}$ estimate fit of about 98%, and by mean of the delay compensation technique, this value is, even with a delay of $n_{\mathrm{d}} = 3$, only reduced to about 97%. The estimate of the vehicle velocity is even 4% improved by the use of the MHE technique in comparison to the EKF. The yaw angle estimate (ψ_{C}) quality for all of the configurations is very high, whereas the yaw rate ($\dot{\psi}^{\mathrm{C}}$) and side slip angle (β^{C}) do vary about 5%, but still have reasonable estimates.

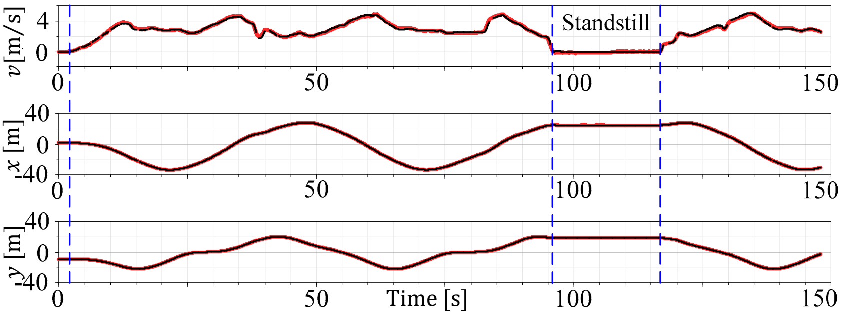


Figure 21. Experiment with vehicle standstill estimation (red) vs. reference (black).

Acknowledgments: The author's thanks goes to the DLR ROboMObil team for supporting the ROboMObil test drives and data acquisition. Moreover, the author's special thanks go to BMBF for supporting the research on the Modelica-based observer framework in the ITEA project MODRIO (BMBF support code 01IS12022G), and the current research on the embedded FMI realization and standardization in the ITEA project EMPHYSIS (BMBF support code 01IS17023B).

Conflicts of Interest: The author declares no conflict of interest.

Appendix A. The Extended Nonlinear Single Track Model Equations

The set of ordinary differential equations of the ordinary differential equation form $x^{\mathrm{C}} = \left\{\beta^{\mathrm{C}}, v^{\mathrm{C}}, \dot{\psi}^{\mathrm{C}}, \psi_{\mathrm{C}}, x_{\mathrm{C}}, y_{\mathrm{C}}\right\}$ can be given as follows:

$$
\begin{aligned}
\frac{\mathrm{d}\beta^{\mathrm{C}}}{\mathrm{d}t} &= \frac{-\sin\left(\beta^{\mathrm{C}}\right)F_x^{\mathrm{C}} + \cos\left(\beta^{\mathrm{C}}\right)F_y^{\mathrm{C}}}{m \cdot v_{\mathrm{mod}}^{\mathrm{C}}} - \dot{\psi}^{\mathrm{C}} \\
\frac{\mathrm{d}v^{\mathrm{C}}}{\mathrm{d}t} &= \frac{\cos\left(\beta^{\mathrm{C}}\right)F_x^{\mathrm{C}} + \sin\left(\beta^{\mathrm{C}}\right)F_y^{\mathrm{C}}}{m} \\
\frac{\mathrm{d}\dot{\psi}^{\mathrm{C}}}{\mathrm{d}t} &= M_Z^{\mathrm{C}} / J_Z^{\mathrm{C}} \\
\frac{\mathrm{d}\psi_{\mathrm{C}}}{\mathrm{d}t} &= \dot{\psi}^{\mathrm{C}} \\
\frac{\mathrm{d}x_{\mathrm{C}}}{\mathrm{d}t} &= v^{\mathrm{C}} \cdot \cos\left(\psi_{\mathrm{C}} + \beta^{\mathrm{C}}\right) \\
\frac{\mathrm{d}y_{\mathrm{C}}}{\mathrm{d}t} &= v^{\mathrm{C}} \cdot \sin\left(\psi_{\mathrm{C}} + \beta^{\mathrm{C}}\right)
\end{aligned}
\tag{A1}
$$

The vehicle side slip (β^{C}) is the angle between the vehicle origin coordinate system and the actual vehicle velocity vector (v^{C}). The yaw angle (ψ_{C}) and yaw angle rate ($\dot{\psi}^{\mathrm{C}}$) describe the rotation with respect to a fixed inertial system ($x^{\mathrm{I}}, y^{\mathrm{I}}$), in which the vehicle positions $x_{\mathrm{C}}, y_{\mathrm{C}}$ are also expressed. Additional quantities are the vehicle mass (m), the yaw moment around the center of gravity (M_Z^{C}), and the corresponding vehicle yaw inertia (J_Z^{C}). To prevent division by zero, the denominator of the vehicle side slip state ($\dot{\beta}^{\mathrm{C}}$) is calculated by use of Equation (A2), which limits the minimum velocity to $v_{\mathrm{min}}^{\mathrm{C}}$ (only valid for small values of $v_{\mathrm{min}}^{\mathrm{C}}$).

$$v_{\mathrm{mod}}^{\mathrm{C}} = \frac{\sqrt{v^{\mathrm{C}} \cdot v^{\mathrm{C}} + 4 \cdot v_{\mathrm{min}}^{\mathrm{C}} \cdot v_{\mathrm{min}}^{\mathrm{C}}} + v^{\mathrm{C}}}{2} \tag{A2}$$

The inputs to the model are the commanded torque set-points to the in-wheel motors and the per axle averaged steering angles $\boldsymbol{u}^{\mathrm{C}} = \left\{\tau^{W_1}, \tau^{W_2}, \tau^{W_3}, \tau^{W_4}, \delta^{W_f}, \delta^{W_r}\right\}$. The forces (marked blue in Figure 3) to the vehicle body center of gravity are calculated by considering the geometric dependencies to the instantaneous steering angles $\delta^{(\cdot)}$, as follows:

$$\begin{aligned}
F_x^{\mathrm{C}} &= -\sin\left(\delta^{\mathrm{W_f}}\right)F_{\mathrm{s}}^{\mathrm{W_f}} - \sin\left(\delta^{\mathrm{W_r}}\right)F_{\mathrm{s}}^{\mathrm{W_r}} + \cos\left(\delta^{\mathrm{W_f}}\right)F_{\mathrm{l}}^{\mathrm{W_f}} + \cos\left(\delta^{\mathrm{W_r}}\right)F_{\mathrm{l}}^{\mathrm{W_r}} - F_{\mathrm{Air}_x}^{\mathrm{C}} \\
F_y^{\mathrm{C}} &= \cos\left(\delta^{\mathrm{W_f}}\right)F_{\mathrm{s}}^{\mathrm{W_f}} + \cos\left(\delta^{\mathrm{W_r}}\right)F_{\mathrm{s}}^{\mathrm{W_r}} + \sin\left(\delta^{\mathrm{W_f}}\right)F_{\mathrm{l}}^{\mathrm{W_f}} + \sin\left(\delta^{\mathrm{W_r}}\right)F_{\mathrm{l}}^{\mathrm{W_r}} - F_{\mathrm{Air}_y}^{\mathrm{C}} \\
M_Z^{\mathrm{C}} &= l_{\mathrm{f}} \cdot \cos\left(\delta^{\mathrm{W_f}}\right)F_{\mathrm{s}}^{\mathrm{W_f}} \\
&-l_{\mathrm{r}} \cdot \cos\left(\delta^{\mathrm{W_r}}\right)F_{\mathrm{s}}^{\mathrm{W_r}} + l_{\mathrm{f}} \cdot \sin\left(\delta^{\mathrm{W_f}}\right)F_{\mathrm{l}}^{\mathrm{W_f}} \\
&-l_{\mathrm{r}} \cdot \sin\left(\delta^{\mathrm{W_r}}\right)F_{\mathrm{l}}^{\mathrm{W_r}} + e_{\mathrm{CoG}} \cdot F_{\mathrm{Air}_y}^{\mathrm{C}} \\
F_{\mathrm{Air}_y}^{\mathrm{C}} &= 0.5 \cdot c_{\mathrm{w}_y} \cdot \rho \cdot A_y \cdot {v_{\mathrm{Air}y}^{\mathrm{C}}}^2
\end{aligned} \tag{A3}$$

In this set of equations, $F_{\mathrm{s}}^{(\cdot)}$ denotes the wheel side forces and $F_{\mathrm{l}}^{(\cdot)}$ the corresponding longitudinal forces. The distances from the center of gravity (CoG) to the rear and front wheels are given by $l_{(\cdot)}$, and the external longitudinal $F_{\mathrm{Air}_x}^{\mathrm{C}}$ and lateral $F_{\mathrm{Air}_y}^{\mathrm{C}}$ air drag forces are applied to the vehicle body. The lateral forces of the tire are calculated by trigonometric functions based on Pacejka's tire model [18], whose input variable is the wheel's side slip angle $\alpha^{\mathrm{W}_{(\cdot)}}$, as follows:

$$\begin{aligned}
F_{\mathrm{s}}^{\mathrm{W_f}} &= D \cdot \sin\left(C \cdot \mathrm{atan}\left(B \cdot \alpha^{\mathrm{W_f}} - E \cdot \left(B \cdot \alpha^{\mathrm{W_f}} - \mathrm{atan}\left(B \cdot \alpha^{\mathrm{W_f}}\right)\right)\right)\right) \\
F_{\mathrm{s}}^{\mathrm{W_r}} &= D \cdot \sin\left(C \cdot \mathrm{atan}\left(B \cdot \alpha^{\mathrm{W_r}} - E \cdot \left(B \cdot \alpha^{\mathrm{W_r}} - \mathrm{atan}\left(B \cdot \alpha^{\mathrm{W_r}}\right)\right)\right)\right) \\
\alpha^{\mathrm{W_f}} &= \left(\delta^{\mathrm{W_f}}\right) - \mathrm{atan}\left(\frac{v_{\mathrm{mod}}^{\mathrm{C}} \cdot \sin\left(\beta^{\mathrm{C}}\right) + l_{\mathrm{f}} \cdot \dot{\psi}^{\mathrm{C}}}{v_{\mathrm{mod}}^{\mathrm{C}} \cdot \cos\left(\beta^{\mathrm{C}}\right)}\right) \\
\alpha^{\mathrm{W_r}} &= \left(\delta^{\mathrm{W_r}}\right) - \mathrm{atan}\left(\frac{v_{\mathrm{mod}}^{\mathrm{C}} \cdot \sin\left(\beta^{\mathrm{C}}\right) - l_{\mathrm{r}} \cdot \dot{\psi}^{\mathrm{C}}}{v_{\mathrm{mod}}^{\mathrm{C}} \cdot \cos\left(\beta^{\mathrm{C}}\right)}\right)
\end{aligned} \tag{A4}$$

Herein, the factor B denotes the tire's stiffness, C is an aspect ratio for the calculated force, D the maximum value of the curve, and E is the modulus of bending rupture. For the longitudinal tire dynamics, a simplified modeling without the consideration of the actual wheel slip was used, as experiments in combination with observer algorithms yielded poor and unstable results, and the vehicle's acceleration range of the maneuvers considered here is limited, as follows:

$$\begin{aligned}
F_{\mathrm{L}}^{\mathrm{W_f}} &= \frac{\tau^{\mathrm{W}_1}}{R} + \frac{\tau^{\mathrm{W}_2}}{R} - f_{\mathrm{rv}} \cdot \left(\frac{m \cdot l_{\mathrm{r}} \cdot g}{l_{\mathrm{f}} + l_{\mathrm{r}}}\right) \\
F_{\mathrm{L}}^{\mathrm{W_r}} &= \frac{\tau^{\mathrm{W}_3}}{R} + \frac{\tau^{\mathrm{W}_4}}{R} - f_{\mathrm{rv}} \cdot \left(\frac{m \cdot l_{\mathrm{f}} \cdot g}{l_{\mathrm{f}} + l_{\mathrm{r}}}\right)
\end{aligned} \tag{A5}$$

Finally, the speed dependent rolling resistance (f_{rv}(and the longitudinal air drag ($F_{\mathrm{Air}_x}^{\mathrm{C}}$) are calculated according to Equation (A6). By use of the "if-statement", the vehicle is prevented from rolling backwards in case of standstill on a planar surface, and no braking torque is applied, as follows:

$$
\begin{gathered}
\text{if } v^{\mathrm{C}} > v^{\mathrm{C}}_{\mathrm{min}} \\
f_{\mathrm{rv}} = f_{\mathrm{R0}} + \frac{f_{\mathrm{R1}} \cdot v^{\mathrm{C}}_{\mathrm{mod}}}{100} + f_{\mathrm{R4}} \cdot \left(\frac{v^{\mathrm{C}}_{\mathrm{mod}}}{100} \right)^4 \\
F^{\mathrm{C}}_{\mathrm{Air}_x} = \frac{1}{2} \cdot c_{\mathrm{w}_x} \cdot \rho \cdot A_x \cdot {v^{\mathrm{C}}_{\mathrm{Air}x}}^2 \\
\text{else} \\
f_{\mathrm{rv}} = 0 \\
F^{\mathrm{C}}_{\mathrm{Air}_x} = 0 \\
\text{end}
\end{gathered}
\tag{A6}
$$

In the following the result of the STM parameter, identification by means of a nonlinear model based optimization is given (cf. Table A1). The derived model is implemented in Modelica and later exported as an FMU for the observer, described in Section 3.2. For the optimization process, the DLR Optimization Library [29] has been utilized. As a tuning algorithm, the pattern search method was chosen. For training and validation, data real world experiments on vehicle test tracks with ROboMObil have been used. ROMO'S ESTM optimized parameters are given in Table A1.

Table A1. ROMO's ROboMObil's—(German Aerospace Center's (DLR's) robotic electric vehicle) extended single track model (ESTM) optimized parameters.

Parameter	Value	Description
$\mathrm{f_{R0}}$	0.009	First parameter of roll resistance
$\mathrm{f_{R1}}$	0.2811588	Second parameter of roll resistance
$\mathrm{f_{R4}}$	0.44906	Fourth parameter of roll resistance
B	5.1088547	Parameter of Pacjeka's MF
C	2.0280	Parameter of Pacjeka's MF
D	724.70	Parameter of Pacjeka's MF
E	0.8903703	Parameter of Pacjeka's MF
$\mathrm{c_{w_x}}$	0.3	Longitudinal air drag coefficient
ρ	$1.249512 \left[\mathrm{N/m^2}\right]$	Air density
$\mathrm{A_x}$	$1.2323485 \left[\mathrm{m^2}\right]$	Effective flow surface front
m	1013 [kg]	ROboMObil mass
$\mathrm{l_f}$	1.218 [m]	CoG to front wheel
$\mathrm{l_r}$	1.182 [m]	CoG to rear wheel
R	0.3722 [m]	Wheel radius
$\mathrm{J^C_Z}$	$1130 \left[\mathrm{kgm^2}\right]$	Chassis moment of yaw inertia

Appendix B. A Constraint Projection Method for Kalman Filters

In the literature [2], a survey on different methods for incorporating inequality constraints to Kalman filter algorithms is given, and three methods for nonlinear Kalman filters that fit for constraints of the type $\boldsymbol{c}(\boldsymbol{x}) \leq \boldsymbol{0}$, which are neither contradictory nor redundant, and only one constraint is active at the same time, are discussed. The algorithm applied here is based on a simplified Newton descent search (i.e., without considering the second order derivative), which projects the a posteriori estimation ($\hat{\boldsymbol{x}}^+_k$) of a Kalman filter on the constrained surface. The optimization objective is formulated as follows:

$$
\begin{gathered}
\hat{\boldsymbol{x}}^{\mathrm{P}}_k = \underset{\boldsymbol{x}}{\mathrm{argmin}} \, ||\boldsymbol{x} - \hat{\boldsymbol{x}}^+_k|| \\
\mathrm{s.t.}c(\boldsymbol{x}) <= 0
\end{gathered}
\tag{A7}
$$

The idea behind this algorithm is to perform a descent search along the gradient ($\nabla c\left(\hat{\boldsymbol{x}}^+_k\right)$) calculated at the point $\hat{\boldsymbol{x}}^+_k$, until the constraint equation $c\left(\hat{\boldsymbol{x}}^{\mathrm{P}}_k\right) \leq 0$ holds again. Graphically this can be interpreted as shown in Figure A1.

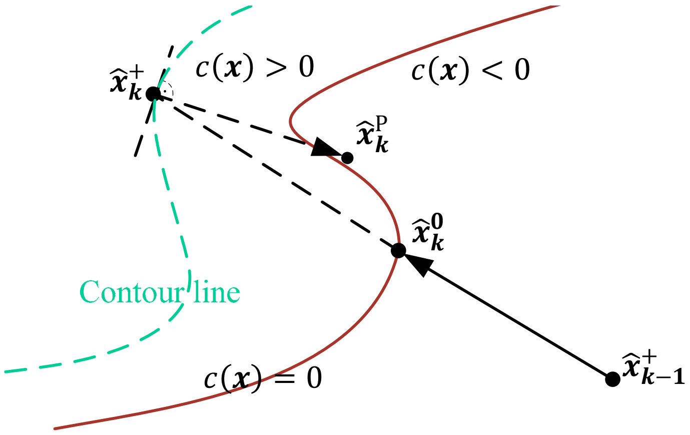

$\hat{\boldsymbol{x}}_{\boldsymbol{k}}^{+}$
$c(\boldsymbol{x}) > 0$
$c(\boldsymbol{x}) < 0$
$\hat{\boldsymbol{x}}_{\boldsymbol{k}}^{\mathrm{P}}$
$\hat{\boldsymbol{x}}_{\boldsymbol{k}}^{\boldsymbol{0}}$
Contour line
$c(\boldsymbol{x}) = 0$
$\hat{\boldsymbol{x}}_{\boldsymbol{k-1}}^{+}$

Figure A1. Principle of the Newton descent search for state constraint handling.

This optimization task can be formulated with the method of Lagrange multipliers. Equation (A8) denotes the above formulated optimization objective in a general description, as follows:

$$\min \boldsymbol{L}(\boldsymbol{x}, \mu) = f(\boldsymbol{x}) - \mu \cdot (g(\boldsymbol{x}) - d) \tag{A8}$$

Therefore, the optimal solution of this unrestricted minimization problem is formulated as follows:

$$\nabla_{\boldsymbol{x},\mu} \boldsymbol{L}(\boldsymbol{x}, \mu) = 0 \tag{A9}$$

This approach is now applied to the estimation projection optimization problem (Equation (A7)) and the searched restricted estimate ($\hat{x}_k^{\mathrm{P}}$) is calculated using the Lagrange multiplier (μ) and the gradient of the active constraint ($c(\boldsymbol{x})$), as follows:

$$\begin{aligned} \hat{x}_k^P &= \hat{x}_k^+ + \mu \cdot \nabla c(\boldsymbol{x}) \text{ s.t.} c(\boldsymbol{x}) = 0 \\ c\left(\hat{x}_k^+ + \mu \cdot \nabla c(\boldsymbol{x})\right) &= 0 \end{aligned} \tag{A10}$$

To fulfill Equation (A10), the approach is expressed with a simplified Newton descent search algorithm by means of a scalar zero search $\mathrm{F}(\mu) = 0$, with respect to the Lagrange variable μ, as follows:

$$\begin{aligned} \nabla c(\boldsymbol{x}) &\approx \nabla c\left(\hat{x}_k^+\right) \\ \overset{\text{yields}}{\rightarrow} \mathrm{F}(\mu) &= c\left(\hat{x}_k^+ + \mu \cdot \nabla c\left(\hat{x}_k^+\right)\right) \overset{!}{=} 0 \\ \frac{\partial \mathrm{F}}{\partial \mu} &= \nabla c\left(\hat{x}_k^+ + \mu \cdot \nabla c\right)^T \cdot \nabla c\left(\hat{x}_k^+\right) \\ \left.\frac{\partial \mathrm{F}}{\partial \mu}\right|_{\mu=0} &= \nabla c\left(\hat{x}_k^+\right)^T \nabla c\left(\hat{x}_k^+\right) \end{aligned} \tag{A11}$$

This can be implemented as an iterative search algorithm, which determines μ such that the condition $\mathrm{F}(\mu) = 0$ holds within a predefined maximum number of calculation steps. The pseudo code for the algorithm is given in Algorithm A1.

Algorithm A1. Simplified Newton descent search algorithm for constrained estimation projection.

```
        1. Set $\mu_o = 0$ and $k = -1$ and iter=1 and max_iter=30
Set $\alpha = \left(\nabla c(\hat{\boldsymbol{x}}_k^+)^T \cdot \nabla c(\hat{\boldsymbol{x}}_k^+)\right)^{-1}$
while $\left|\alpha \cdot \mathrm{c}\left(\hat{\boldsymbol{x}}_k^+ + \mathrm{\mu}_{\mathrm{k}+1} \cdot \nabla \mathrm{c}\left(\hat{x}_k^+\right)\right)\right| \geq \epsilon_{\mathrm{Newton}}$ and iter < max_iter do
    $k = k + 1$
    $\Delta\mu_{k+1} = -\alpha \cdot c\left(\hat{\boldsymbol{x}}_k^+ + \mu_k \cdot \nabla c\left(\hat{\boldsymbol{x}}_k^+\right)\right)$
    $\mu_{k+1} = \mu_k + \Delta\mu_{k+1}$
    $iter = iter + 1$
end while
Solution: $\hat{\boldsymbol{x}}_k^{\mathrm{P}} = \hat{\boldsymbol{x}}_k^+ + \mu_{k+1} \cdot \nabla c\left(\hat{\boldsymbol{x}}_k^+\right)$
```

Appendix C. Lists of Symbols, Nomenclatures, and Abbreviations

Formula symbol	Unit	Description
β^{C}	[rad]	Vehicle's side slip angle
v^{C}	[m/s]	Vehicle's velocity over ground
$\dot{\psi}^{\mathrm{C}}$	[rad/s]	Vehicle's yaw rate
ψ_{C}	[rad]	Vehicle's yaw angle
x_{C}	[m]	Vehicle's position in x direction
y_{C}	[m]	Vehicle's position in y direction

Nomenclature	Explanation
e	Lower case letter variable is a scalar
$\boldsymbol{e}$	Bold lower case letter variable is a vector
$\boldsymbol{E}$	Bold upper case letter variable is a matrix
$\boldsymbol{e} = \{e_1, e_2\}$ $= (e_1, e_2)^T$	Equivalent vector notations
$\boldsymbol{E} = \begin{bmatrix} e_{11} & e_{12} \\ e_{21} & e_{22} \end{bmatrix}$	Matrix notation
$(\cdot)^{\mathrm{W}_x}$	Quantity expressed in the x-th wheel robot coordinate system parallel to the car coordinate system
$(\cdot)^{\mathrm{C}}$	Quantity expressed in the car coordinate system with origin in CoG
$(\cdot)_{\mathrm{C}}^{\mathrm{P}}$	Car quantity expressed in the path coordinate system with origin in CoG
$(\cdot)_{\mathrm{C}}$	Car quantity expressed in the inertial coordinate system – short for $(\cdot)_{\mathrm{C}}^{\mathrm{I}}$
$(\cdot)_{\mathrm{P}}$	Quantity expressed in the path coordinate system
$\min_{\boldsymbol{x}} \|\boldsymbol{e}(\boldsymbol{x})\|_{\boldsymbol{W}}$ s.t.	With $\boldsymbol{W}$ weighted least squares minimization of function $\boldsymbol{e}$ with respect to $\boldsymbol{x}$"subject to" if the problem is restricted

Abbreviation	Explanation
ABX	AutoBox—rapid prototyping real-time controller from dSPACE GmbH
ACADO	Software environment and algorithm collection for automatic control and dynamic optimization [20]
ADAC	General German Automobile Club
BC	Boundary constraint
CAN	Controller area network—vehicle bus standard
CoG	Center of gravity
DLR	German Aerospace Center
Dymola	Modelica simulator from Dassault Systèmes

Abbreviation	Explanation
EKF	Extended Kalman filter
ESTM	Extended single track model
FIFO	First in first out (buffer)
FMI	Functional mockup interface
FMU	Functional mockup unit
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
IMU	Inertial measurement unit
INS	Inertial navigation system
LAPACK	Linear algebra package
MABX2	MicroAutoBox II—embedded rapid prototyping real-time controller from dSPACE GmbH
MHE	Moving horizon estimation/estimator
Modelica	Object oriented modeling language for multiphysical systems
NG	Nonlinear gradient (descent search)
OxTS	Inertial measurement unit from Oxford Technical Solutions Ltd.
PFC	Path following control
QP	Quadratic program
RObоMObil	see ROMO
ROMO	short for RObоMObil—DLR's robotic electric vehicle
RTI	Real-time iteration
SPKF	Sigma point Kalman filter
SQP	Sequential quadratic program
SR	Square-root
SR-EKF	Square-root extended Kalman filter
SR-UKF	Square-root unscented Kalman filter
TIPI	Time independent path interpolation
TM	Traction motor
UKF	Unscented Kalman filter
Wheel robot	Wheel unit that integrates propulsion, steering, and brake c.f. [1]

References

1. Brembeck, J.; Ho, L.M.; Schaub, A.; Satzger, C.; Tobolar, J.; Bals, J.; Hirzinger, G. ROMO—The Robotic Electric Vehicle. In Proceedings of the 22nd IAVSD International Symposium on Dynamics of Vehicle on Roads and Tracks, Manchester, UK, 14–19 August 2011.
2. Brembeck, J. Model Based Energy Management and State Estimation for the Robotic Electric Vehicle RObоMObil. Ph.D. Thesis, Technischen Universität München, Munich, Germany, 2018.
3. Ritzer, P.; Winter, C.; Brembeck, J. Advanced Path Following Control of an Overactuated Robotic Vehicle. In Proceedings of the IEEE Intelligent Vehicles Symposium, Seoul, Korea, 28 June–1 July 2015; pp. 1120–1125.
4. Winter, C.; Ritzer, P.; Brembeck, J. Experimental Investigation of Online Path Planning for Electric Vehicles. In Proceedings of the 19th the IEEE International Conference on Intelligent Transportation Systems (ITSC), Rio de Janeiro, Brasil, 1–4 November 2016; pp. 1403–1409.
5. Modelica Association. Modelica. Available online: http://www.modelica.org (accessed on 28 June 2018).
6. Modelica Association. Functional Mockup Interface. Available online: https://www.fmi-standard.org/ (accessed on 1 June 2018).
7. Grewal, M.S.; Andrews, A.P. *Kalman filtering: Theory and practice using MATLAB*, 4th ed.; John Wiley & Sons Inc.: Hoboken, NJ, USA, 2015.
8. Yoon, J.-W.; Kim, B.-W. Vehicle position estimation using nonlinear tire model for autonomous vehicle. *J. Mech. Sci. Technol.* **2016**, *30*, 3461–3468. [CrossRef]
9. Groves, P. *Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems, Second Edition*, 2nd ed.; Artech House: Lodon, UK, 2013.

10. Simon, D. *Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches*, 1st ed.; Wiley & Sons: Cleveland, OH, USA, 2006.
11. Merwe, R.v.d.; Wan, E.A.; Julier, S. Sigma-Point Kalman Filters for Nonlinear Estimation and Sensor-Fusion: Applications to Integrated Navigation. In Proceedings of the AIAA Guidance, Navigation & Control. Conference, Providence, IR, USA, 16–19 August 2004; pp. 16–19.
12. Hansen, J.M.; Fossen, T.I.; Johansen, T.A. Nonlinear observer for INS aided by time-delayed GNSS measurements: Implementation and UAV experiments. In Proceedings of the International Conference on Unmanned Aircraft Systems (ICUAS), Denver, CO, USA, 9–12 June 2015; pp. 157–166.
13. Girrbach, F.; Hol, J.D.; Bellusci, G.; Diehl, M. Optimization-Based Sensor Fusion of GNSS and IMU Using a Moving Horizon Approach. *Sensors* **2017**, *17*, 1159. [CrossRef] [PubMed]
14. Omar, H.M.; Yanzhong, Z.; Bo, Z.; Gul, H.U. Integration of GPS and dead reckoning navigation system using moving horizon estimation. In Proceedings of the IEEE Information Technology, Networking, Electronic and Automation Control. Conference, Chongqing, China, 20–22 May 2016; pp. 553–556.
15. Liu, A.; Zhang, W.; Yu, L.; Chen, J. Moving horizon estimation for multi-rate systems. In Proceedings of the 54th IEEE Conference on Decision and Control. (CDC), Osaka, Japan, 15–18 December 2015; pp. 6850–6855.
16. Huang, Y.; Zhang, Y.; Shi, P.; Wu, Z.; Qian, J.; Chambers, J.A. Robust Kalman Filters Based on Gaussian Scale Mixture Distributions With Application to Target Tracking. *IEEE Trans. Syst. Man Cybern. Syst. Hum. (Early Access)* **2018**, 1–15. [CrossRef]
17. Sun, Q.; Lim, C.; Shi, P.; Liu, F. Design and Stability of Moving Horizon Estimator for Markov Jump Linear Systems. *IEEE Trans. Autom. Control.* **2019**, *64*, 1109–1124. [CrossRef]
18. Pacejka, H. *Tire and Vehicle Dynamics*, 3rd ed.; Elsevier: Amsterdam, The Netherlands, 2012; p. 672.
19. Simon, D. Kalman filtering with state constraints: a survey of linear and nonlinear algorithms. *IET Control Theory Appl.* **2010**, *4*, 1303–1318. [CrossRef]
20. Vukov, M. Embedded Model. Predictive Control. and Moving Horizon Estimation for Mechatronics Applications. Ph.D. Thesis, KU Leuven – Faculty of Engineering Science, Leuven, Belgium, 2014.
21. Boegli, M. Real-Time Moving Horizon Estimation for Advanced Motion Control. Application to Friction State and Parameter Estimation. Ph.D. Thesis, KU Leuven, Leuven, Belgium, 2014.
22. Ferreau, H.J.; Kraus, T.; Vukov, M.; Saeys, W.; Diehl, M. High-speed moving horizon estimation based on automatic code generation. In Proceedings of the 51st Annual Conference on Decision and Control. (CDC), Maui, HI, USA, 10–13 December 2012; pp. 687–692.
23. Kühl, P.; Diehl, M.; Kraus, T.; Schlöder, J.P.; Bock, H.G. A real-time algorithm for moving horizon state and parameter estimation. *J. Comput. Chem. Eng.* **2011**, *35*, 71–83. [CrossRef]
24. Winter, C.; Brembeck, J.; Kennel, R. Online Energy Optimal Path Planner for Advanced Electric Vehicles. Master's Thesis, Technische Universität München, Munich, Germany, 2013.
25. Golub, G.H.; Van Loan, C.F. *Matrix Computations*, 4th ed.; Johns Hopkins University Press: Baltimore, MD, USA, 2013.
26. Brembeck, J.; Pfeiffer, A.; Fleps-Dezasse, M.; Otter, M.; Wernersson, K.; Elmqvist, H. Nonlinear State Estimation with an Extended FMI 2.0 Co-Simulation Interface. In Proceedings of the 10th International Modelica Conference, Lund, Sweden, 10–12 March 2014; pp. 53–62.
27. Joos, H.-D.; Bals, J.; Looye, G.; Schnepper, K.; Varga, A. Optimisation-based multi-objective design and assessment—MOPS—Multi-Objective Parameter Synthesis. Available online: https://www.dlr.de/rm/en/desktopdefault.aspx/tabid-3842/6343_read-9099/ (accessed on 10 November 2018).
28. Kouzoupis, D.; Quirynen, R.; Girrbach, F.; Diehl, M. An efficient SQP algorithm for Moving Horizon Estimation with Huber penalties and multi-rate measurements. In Proceedings of the Conference on Control. Applications (CCA), Buenos Aires, Argentina, 19–22 September 2016; pp. 1482–1487.
29. Pfeiffer, A. Optimization Library for Interactive Multi-Criteria Optimization Tasks. In Proceedings of the 9th International Modelica Conference, Munich, Germany, 3–5 September 2012; pp. 669–679.