2013 European Control Conference (ECC)
July 17-19, 2013, Zürich, Switzerland.

Nonlinear Moving Horizon Estimation for Combined State and Friction Coefficient Estimation in Autonomous Driving

Mario Zanon, Janick V. Frasch and Moritz Diehl

Abstract— **Real-time autonomous driving requires a precise knowledge of the state and the ground parameters, especially in dangerous situations. In this paper, an accurate yet computationally efficient nonlinear multi-body vehicle model is developed, featuring a detailed Pacejka tire model, and a Moving Horizon Estimation (MHE) scheme is formulated. To meet the real-time requirements, an efficient algorithm based on the Real Time Iteration (RTI) scheme for the Direct Multiple Shooting method is exported through automatic C code generation. The exported plain C-code is tailored to the model dynamics, resulting in computation times in the range of a few milliseconds. In addition to state estimates, MHE provides friction coefficient estimates, allowing the controller to adapt to varying road conditions. Simulation results from an obstacle avoidance scenario on a low friction road are presented.**

Keywords **: Moving Horizon Estimation, code generation, autonomous driving, road friction estimation**

I. Introduction

Increasing computational power, improvements in the optimization algorithms and theoretical studies on the stability properties have been the major contributors to the success of optimization-based techniques for control and estimation. Thanks to those advances, the application of Model Predictive Control (MPC) to the control of autonomous vehicles has recently become attractive, as it allows one to deal with constrained nonlinear systems in a natural way, and to control the vehicle even in dangerous situations [10], [9], [6].

The design of such feedback control schemes typically assumes that all the current process states and parameters are known. While most of the parameters can be identified offline in many real applications, particularly for detailed models, the full state cannot be measured and some parameters might need to be estimated online. Popular estimation techniques are the Kalman filter with its many extensions, Moving Horizon Estimation (MHE), and Particle Filtering (PF). The Kalman filter is an exact maximum likelihood (ML) estimator for unconstrained, linear systems under the assumption of normally distributed state and measurement noise [15], [11]. In the nonlinear case, a variety of extensions of the Kalman filter have been proposed. Popular variants are the Extended Kalman Filter (EKF) [11], [1], which relies on a linearization of the nonlinear model at the current state, or the Unscented Kalman Filter (UKF) [14], which samples the nonlinear response at carefully chosen points instead of linearizing at a single point.

Unlike the previously mentioned methods, Particle Filtering (PF) does not need any assumption on the probability distribution of the noise, but instead approximates the distribution via Monte Carlo sampling. The major drawback of this last approach is the so called "curse of dimensionality", which bounds the application of such methods to relatively small state spaces. Unlike Kalman filters, PF is an ML estimator for arbitrary noise density functions, as it relies on Monte Carlo sampling to approximate it [22].

MHE considers the evolution of a possibly nonlinear model on a fixed time horizon, and minimizes the deviation of the model from the last N measurements subject to equality and inequality constraints [21]. For unconstrained, linear systems and Gaussian noise, MHE can be made equivalent to the Kalman filter. Kalman filters and MHE are ML estimators only for normal noise distribution, and loose this property when the actual distribution is different. Model nonlinearities generally lead to an asymmetric and potentially multi-modal probability distribution. By observing that MHE is a deterministic approach, it can be argued that MHE does not need to rely on any specific error distribution. The probabilistic insight is nevertheless still very valuable for the tuning of the weighting marices. Moreover, as constraints can be added to the optimization problem in a very natural way, MHE can handle constrained processes. For a survey on the mentioned estimation methods, excellent overviews can be found in [22], [3].

Several estimation methods have been successfully applied to mechanical systems. The comparison of different techniques or the assessment of which technique is the best suited for vehicle state and parameter estimation is beyond the scope of this paper.

As ground vehicles exhibit highly nonlinear dynamics, it is proposed to estimate the vehicle state and the road friction coefficient by means of nonlinear MHE. General purpose implementations of optimization-based estimation techniques for nonlinear systems typically require computation times too long for real-time applications. To tackle this issue, an auto-generated algorithm based on the real time iteration scheme (RTI) [4] for direct multiple shooting [2] is proposed.

M. Zanon and M. Diehl are with the Optimization in Engineering Center (OPTEC), K.U. Leuven, Kasteelpark Arenberg 10, B-3001 Leuven-Heverlee, Belgium. Janick Frasch is with the Interdisciplinary Center for Scientific Computing (IWR), University of Heidelberg. Im Neuenheimer Feld 368, D-69120 Heidelberg, Germany. mario.zanon@esat.kuleuven.be

* This research was supported by Research Council KUL: PFV/10/002 Optimization in Engineering Center OPTEC, GOA/10/09 MaNet and GOA/10/11 Global real- time optimal control of autonomous robots and mechatronic systems. Flemish Government: IOF / KP / SCORES4CHEM, FWO: PhD/postdoc grants and projects: G.0320.08 (convex MPC), G.0377.09 (Mechatronics MPC); IWT: PhD Grants, projects: SBO LeCoPro; Belgian Federal Science Policy Office: IUAP P7 (DYSCO, Dynamical systems, control and optimization, 2012-2017); EU: FP7- EMBOCON (ICT-248940), FP7-SADCO (MC ITN-264735), ERC ST HIGHWIND (259 166), Eurostars SMART, ACCM.

This paper is organized as follows. The vehicle model is detailed in Section II. Section III presents the proposed MHE scheme. Simulation results are presented in Section IV and Section V provides conclusions and proposes future work perspectives.

Contributions of the paper: an MHE algorithm is formulated based on an accurate vehicle model taking the most important dynamic effects into account. The proposed observer is tested in simulations considering an icy road. The proposed scheme provides friction coefficient estimates under observability assumptions.

II. System Model

In [10] a 6 degrees of freedom (DoF) vehicle model was used for the obstacle avoidance problem. This model is extended in the following to account for some of the unmodeled dynamics, namely wheel dynamics and load transfer. The model proposed here is also used in [9]. The time dependent dynamic system is then transformed into a track progress dependent one in a similar fashion as presented for the obstacle avoidance problem in [10].

The vehicle chassis is modeled as a rigid body, described by its global position in the X-Y plane, its global orientation, and the corresponding velocities in a local x-y-z frame. The four wheels are modeled as independent bodies with only spinning inertia. Roll, pitch and heave (vertical displacement) motions of the car are neglected and a rigid suspension model has been assumed; the effect of these motions on the vehicle load transfer is however included, see Section II-D. A car with front steering and rear wheel drive is considered. The control inputs are the steering rate $\dot{\delta}$, the accelerating engine torque T^{a} and four braking torques $T^{\mathrm{b}}_{\mathrm{fl}}$, $T^{\mathrm{b}}_{\mathrm{fr}}$, $T^{\mathrm{b}}_{\mathrm{rl}}$, $T^{\mathrm{b}}_{\mathrm{rr}}$. Throughout this paper, subscripts $\mathrm{fl}, \mathrm{fr}, \mathrm{rl}, \mathrm{rr}$ denote quantities corresponding to the front left, front right, rear left and rear right wheel, respectively. For clarity of notation let $\mathcal{F} := \{\mathrm{f}, \mathrm{r}\}$, $\mathcal{S} := \{\mathrm{l}, \mathrm{r}\}$ and $\mathcal{F} \times \mathcal{S} = \{\mathrm{fl}, \mathrm{fr}, \mathrm{rl}, \mathrm{rr}\}$.

A. Chassis Dynamics

Quantities are described in an orthonormal reference frame in the vehicle's center of gravity (CoG) with the z-direction pointing upwards. The chassis dynamic equations used in this paper are

$$m\dot{v^x} = mv^y\dot{\psi} + F^x_{\mathrm{fr}} + F^x_{\mathrm{fl}} + F^x_{\mathrm{rr}} + F^x_{\mathrm{rl}} + F_D, \tag{1a}$$
$$m\dot{v^y} = -mv^x\dot{\psi} + F^y_{\mathrm{fr}} + F^y_{\mathrm{fl}} + F^y_{\mathrm{rr}} + F^y_{\mathrm{rl}}, \tag{1b}$$
$$I^z\ddot{\psi} = a(F^y_{\mathrm{fl}} + F^y_{\mathrm{fr}}) - b(F^y_{\mathrm{rl}} + F^y_{\mathrm{rr}}) \tag{1c}$$
$$+ c(F^x_{\mathrm{fr}} - F^x_{\mathrm{fl}} + F^x_{\mathrm{rr}} - F^x_{\mathrm{rl}}), \tag{1d}$$
$$\dot{X} = v^x\cos\psi - v^y\sin\psi, \tag{1e}$$
$$\dot{Y} = v^x\sin\psi + v^y\cos\psi, \tag{1f}$$

where m denotes the mass and I^z the moment of inertia of the car. The distances of the tires from the vehicle's CoG are characterized by a, b and c, cf. Figure 1. Note the CoG is assumed to be located halfway between the left and right side of the car. The vehicle's yaw angle ψ is obtained by direct integration of $\dot{\psi}$ as is the steering angle δ from input

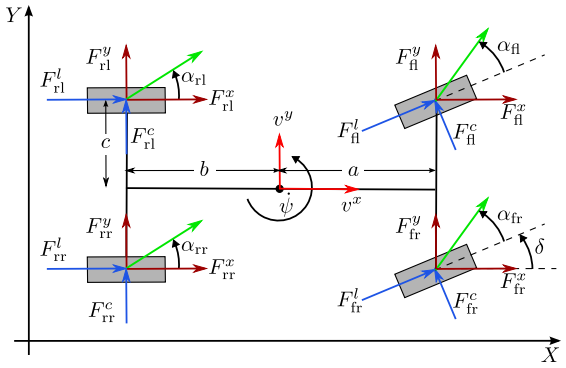


Fig. 1. Tire forces and slip angles of the 4-wheel vehicle model in inertial coordinates. The tires' directions of movement are indicated by green vectors.

$\dot{\delta}$. The drag force due to air resistance is denoted by F_D, while $F^x_{\cdot\cdot}$ and $F^y_{\cdot\cdot}$ denote the components of the tire contact forces along the vehicle's local x and y axis. For the rear wheels it holds

$$F^x_{\mathrm{r}\star} = F^l_{\mathrm{r}\star}, \qquad F^y_{\mathrm{r}\star} = F^c_{\mathrm{r}\star} \qquad \forall\, \star \in \mathcal{S},$$

while the forces on the front wheels are computed as

$$F^x_{\mathrm{f}\star} = F^l_{\mathrm{f}\star}\cos\delta - F^c_{\mathrm{f}\star}\sin\delta \qquad \forall\, \star \in \mathcal{S},$$
$$F^y_{\mathrm{f}\star} = F^l_{\mathrm{f}\star}\sin\delta - F^c_{\mathrm{f}\star}\cos\delta \qquad \forall\, \star \in \mathcal{S}.$$

The forces $F^l_{\cdot\cdot}$ and $F^c_{\cdot\cdot}$ denote lateral and cornering forces, respectively, for each tire. These forces are computed from a Pacejka-type tire model, see Section II-B.

B. The Pacejka-type Tire Force Model

Pacejka's so-called Magic Formula (cf. [19]) is a semi-empirical tire model to calculate steady-state tire force and moment characteristics. It includes a combined slip model which couples the vertical load $F^z_{\cdot\cdot}$ to the longitudinal force $F^l_{\cdot\cdot}$ and the cornering force $F^c_{\cdot\cdot}$. In this paper the self-aligning moment $M^z_{\cdot\cdot}$ is neglected, as its impact on the dynamics is only significant at slow speeds [19]. Longitudinal and cornering forces are given by

$$F^l_{\diamond\star} = f^l_{\mathrm{P}}(\alpha_{\diamond\star}, \kappa_{\diamond\star}, \mu, F^z_{\diamond\star}) \qquad \forall\ \diamond\star \in \mathcal{F}\times\mathcal{S},$$
$$F^c_{\diamond\star} = f^c_{\mathrm{P}}(\alpha_{\diamond\star}, \kappa_{\diamond\star}, \mu, F^z_{\diamond\star}) \qquad \forall\ \diamond\star \in \mathcal{F}\times\mathcal{S},$$

where $f^l_{\mathrm{P}}(\cdot)$ and $f^c_{\mathrm{P}}(\cdot)$ denote longitudinal and cornering components of Pacejka's tire model.

For each tire $\diamond\star \in \mathcal{F}\times\mathcal{S}$ the main parameters and inputs to the model are the side slip angles $\alpha_{\diamond\star}$, defined as the angle between the wheel's orientation and the actual direction of movement (cf. Figure 1), the slip ratio $\kappa_{\diamond\star} = \frac{\omega_{\diamond\star}R_{\mathrm{e}} - v_{\diamond\star}}{v_{\diamond\star}}$ representing the longitudinal slip while accelerating and braking, and the normal load $F^z_{\diamond\star}$. Here, $v_{\diamond\star}$ denotes the wheel speed with respect to the ground, and $\omega_{\diamond\star}$ denotes the rotational speed of the weel, while R_{e} is taken to be the tire radius. The friction coefficient μ models the road friction in the tire model. More details on the computation of slip angles and Pacejka forces can be found in [17], [19], [16]; the precise model implementation used for this paper, including all parameters, can be found in [12].

C. Wheel Dynamics

The wheels' rotational velocities $\omega_{\cdot\cdot}$ are computed from a first order model in the accelerating torques $T^{\mathrm{a}}_{\cdot\cdot}$ and the braking torques $T^{\mathrm{b}}_{\cdot\cdot}$, taking the wheel moment of inertia I^{w} into account. Note that the consideration of wheel dynamics enhances the predictive quality of the model but also renders the underlying ODE system stiff, making its solution computationally significantly more challenging. The dynamic equations are given by

$$\dot{\omega}_{\diamond\star} = \frac{1}{I^{\mathrm{w}}}(T^{\mathrm{a}}_{\diamond\star} + T^{\mathrm{b}}_{\diamond\star} - R_{\mathrm{e}} F^{l}_{\diamond\star}), \quad \forall\ \diamond\star \in \mathcal{F}\times\mathcal{S} \tag{2}$$

where R_{e} is taken to be the tire radius and individual wheel braking is considered. A differential model is included for the total acceleration torque T^{a} (w.l.o.g. assuming a rear-wheel driven car), yielding

$$\begin{aligned} T^{\mathrm{a}}_{\mathrm{f}\star} &= 0 && \forall\ \star \in \mathcal{S},\\ T^{\mathrm{a}}_{\mathrm{r}\star} &= T^{\mathrm{a}}\left(1 - \frac{\omega_{\mathrm{r}\star}}{\omega_{\mathrm{rl}} + \omega_{\mathrm{rr}}}\right) && \forall\ \star \in \mathcal{S}. \end{aligned}$$

D. Vertical Forces

The vertical loads $F^{z}_{\cdot\cdot}$ acting on each wheel are obtained from the equilibrium of the main body w.r.t. the vertical forces, and roll and pitch torques. For a height h between the vehicle's CoG and its projection onto the wheelbase longitudinal and lateral load transfer, $\tilde{\Delta}^{z}_{\mathrm{lon}}$ and $\tilde{\Delta}^{z}_{\mathrm{lat}}$, are given by

$$\begin{aligned} 2\,\tilde{\Delta}^{z}_{\mathrm{lon}} &= (F^{x}_{\mathrm{fl}} + F^{x}_{\mathrm{fr}} + F^{x}_{\mathrm{rl}} + F^{x}_{\mathrm{rr}})\,\frac{h}{a+b},\\ 2\,\tilde{\Delta}^{z}_{\mathrm{lat}} &= (F^{y}_{\mathrm{fl}} + F^{y}_{\mathrm{fr}} + F^{y}_{\mathrm{rl}} + F^{y}_{\mathrm{rr}})\,\frac{h}{2\,c}. \end{aligned}$$

Note that this formulation of the load transfer implies an algebraic loop through the tire model, which can be relaxed by introducing a first order model with time constant τ_{LT}, yielding

$$\dot{\Delta}^{z}_{\bullet} = \frac{1}{\tau_{\mathrm{LT}}}(\tilde{\Delta}^{z}_{\bullet} - \Delta^{z}_{\bullet}) \qquad \forall\ \bullet \in \{\mathrm{lon}, \mathrm{lat}\}, \tag{3}$$

where $\Delta^{z}_{\bullet}$ is added to the model as a state.

Denoting the rest normal loads by $\bar{F}^{z}_{\cdot\cdot}$, the vertical forces $F^{z}_{\cdot\cdot}$ are given by

$$\begin{aligned} F^{z}_{\mathrm{fl}} &= \bar{F}^{z}_{\mathrm{fl}} - \Delta^{z}_{\mathrm{lon}} - \Delta^{z}_{\mathrm{lat}},\\ F^{z}_{\mathrm{fr}} &= \bar{F}^{z}_{\mathrm{fr}} - \Delta^{z}_{\mathrm{lon}} + \Delta^{z}_{\mathrm{lat}},\\ F^{z}_{\mathrm{rl}} &= \bar{F}^{z}_{\mathrm{rl}} + \Delta^{z}_{\mathrm{lon}} - \Delta^{z}_{\mathrm{lat}},\\ F^{z}_{\mathrm{rr}} &= \bar{F}^{z}_{\mathrm{rr}} + \Delta^{z}_{\mathrm{lon}} + \Delta^{z}_{\mathrm{lat}}. \end{aligned}$$

Future work will aim at including the suspension in the model, to improve the predictive qualities of the model. Note that, in the presence of a suspension, no algebraic loop is present.

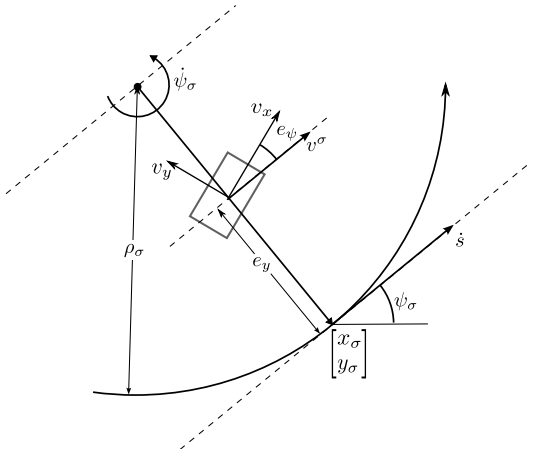


Fig. 2. The curvilinear coordinate system. The dynamics are derived about a curve defining the center-line of a track. The coordinate s defines the arc-length along the track. The relative spatial coordinates e_y and e_ψ are shown.

E. Spatial Reformulation of Dynamics

A model transformation from time-dependent vehicle dynamics to track-dependent (spatial) dynamics is proposed, in a similar manner as presented in [9]. This allows a natural formulation of obstacles and general road bounds under varying vehicle speed.

The X-Y coordinates are projected on a curve σ, taken e.g. as the center-line of a road. The ODE model is then stated w.r.t. the independent variable s, the parametrization of σ by its arc-length. The states X, Y, ψ are replaced by $e_y := \left\| [X,Y]^T - [X_\sigma, Y_\sigma]^T \right\|_2$ and $e_\psi := \psi - \psi_\sigma$, where $[X_\sigma, Y_\sigma]^T$ and ψ_σ denote the position and orientation of the current reference point on the path given by s. Figure 2 details the states in the new curvilinear coordinate system. The full system of states ξ and control inputs ν for this vehicle model are listed in Table I.

The spatial dynamics of the state vector ξ in relation to the time dependent dynamics are

$$\xi' := \frac{d\xi}{ds} = \frac{d\xi}{dt}\,\frac{dt}{ds}\,.$$

If $\dot{s} \neq 0$ is assumed at any time, by the inverse function theorem $\frac{dt}{ds} = \frac{1}{\dot{s}}$ holds. The coordinate transformation is therefore written as a division of the state vector ξ by $\dot{s}$. The spatial dynamics thus read

$$\xi' = \frac{1}{\dot{s}}\,\dot{\xi},$$

where $\dot{\xi}$ is defined in Equations (1), (2), and (3). In the following of the paper, the right-hand side function of the system dynamics will be denoted as $f(\xi, \nu, \mu)$, where ξ and ν denote the state and control vector respectively.

If necessary, time information may be recovered by integrating $\frac{dt}{ds}$ along σ:

$$t(s) = \int_{s_0}^{s} \frac{1}{\dot{s}(\tau)}\,\mathrm{d}\tau\,.$$

State ξ	Unit	Description
v^x	$\frac{\mathrm{m}}{\mathrm{s}}$	Longitudinal velocity of vehicle
v^y	$\frac{\mathrm{m}}{\mathrm{s}}$	Lateral velocity of vehicle
$\dot{\psi}$	$\frac{\mathrm{rad}}{\mathrm{s}}$	Vehicle's yaw rate
e_ψ	rad	Yaw angle relative to path
e_y	m	Deviation from center-line
δ	rad	Steering angle
$\omega_{..}$	$\frac{\mathrm{rad}}{\mathrm{s}}$	Rotational wheel speeds
$\Delta^z_{.}$	N	Vertical load transfer

Control ν	Unit	Description
$\dot{\delta}$	rad	Front steering rate
T_a	Nm	Engine torque
$T_{b..}$	Nm	Brake torques

TABLE I

STATES AND CONTROL INPUTS OF THE SPATIAL VEHICLE MODEL.

Inertial coordinates may be recovered by a transformation from the spatial coordinates to the global coordinates:

$$\begin{aligned} X &= X_\sigma - e_y \sin(\psi_\sigma) \\ Y &= Y_\sigma + e_y \cos(\psi_\sigma) \\ \psi &= \psi_\sigma + e_\psi \,. \end{aligned}$$

F. Available Measurements

In this paper it is assumed that the vehicle is equipped with a) a GPS providing position measurements, b) an inertial measurement unit (IMU) providing accelerations and rotational velocity, c) encoders providing the rotational speed of the wheels and the steering angle and d) force sensors on the suspensions providing vertical force measurements. All measurements are summarized in Table II. In the following all measurements are lumped together in the function $y(\xi, \mu)$.

G. Estimation problem

The observer is formulated as a MHE scheme using a least squares (LSQ) function penalizing the deviation of the process outputs from the measurements. The MHE is based on repeatedly ($k = 0, 1, \ldots$) solving the following dynamic optimization problem:

$$\begin{aligned} \min_{\xi(\cdot),\nu(\cdot),\mu} \quad & \int_{s_k - S_\mathrm{E}}^{s_k} \Big(\|y(\xi(\tau),\mu) - \bar{y}(\tau)\|^2_{\Sigma_y} + \\ & \qquad \|\nu(\tau) - \bar{\nu}(\tau)\|^2_{\Sigma_\nu} \Big) d\tau, \\ s.t. \quad & \xi' = f\left(\xi, \nu, \mu\right), \end{aligned} \tag{4}$$

where s_k is the current pseudo-time, S_E the MHE estimation horizon. Here $\|\cdot\|_{\{\Sigma_y,\Sigma_\nu\}}$ denotes the Euclidian norm with diagonal positive-definite matrices Σ_y and Σ_ν. Note that the controls are also included in the dynamic optimization problem, to account for the fact that the controls applied to the system may differ from the controls computed by the controller $\bar{\nu}$ due to actuator noise and inaccuracy.

Sensor	Measurements	Standard deviation
IMU	Linear acceleration	$0.1\ \mathrm{m/s^2}$
IMU	Angular velocity	$0.1\ \mathrm{rad/s}$
GPS	Position	$0.1\ \mathrm{m}$
Force sensor	Vertical forces	$5 \cdot 10^3\ \mathrm{N}$
Encoder	Wheel rotational velocity	$10^{-3}\ \mathrm{rad/s}$
Encoder	Steering angle	$10^{-2}\ \mathrm{rad}$

TABLE II

AVAILABLE MEASUREMENTS

III. NUMERICAL SOLUTION METHOD

The dynamic optimization problem (4) can be generalized to the following form:

$$\begin{aligned} \min_{\xi(\cdot),\nu(\cdot),p} \quad & \|\xi(s_k - S_\mathrm{E}) - \bar{\xi}(s_k - S_\mathrm{E})\|^2_{\Sigma_\xi} + \\ & \int_{s_k - S_\mathrm{E}}^{s_k} \|y(\xi(\tau),\nu(\tau),p) - \bar{y}(\tau)\|^2_{\Sigma_y} d\tau, \\ \text{s.t.} \quad & \xi'(s) = f\left(\xi(s), \nu(s), p\right), \ \forall s \in [s_k - S_\mathrm{E}, S_\mathrm{E}], \\ & h\left(\xi(s), \nu(s), p\right) \le 0, \qquad \forall s \in [s_k - S_\mathrm{E}, S_\mathrm{E}]. \end{aligned} \tag{5}$$

As in the previous formulation, the nonlinear right-hand side of the model equations is defined as $f\left(\xi(s), \nu(s), p\right)$, where ξ are the states, ν the control inputs and p the model parameters to be estimated online. The output function $y(\xi(\tau), \nu(\tau), p)$ is denoted here in a general form, depending on the state, control inputs and parameters. The arrival cost penalizes the deviation of the initial state $\xi(s_k - S_\mathrm{E})$ from its estimate $\bar{\xi}(s_k - S_\mathrm{E})$ with weighting matrix Σ_ξ. Constraints are summarized in the generic form $h\left(\xi(s), \nu(s), p\right)$.

A. Direct Multiple Shooting Method

To solve problem (4), it is proposed to use the direct multiple shooting method, a "first discretize, then optimize" approach, first introduced by Bock and Plitt [2]. The space of feasible controls is reduced to a finite dimensional one, defined on a time grid $\mathcal{T} = \{t_k \in \mathbb{R},\ k = 1, \ldots, N \mid t_k < t_j,\ \forall k < j\}$, and the system dynamics are integrated on each time interval $[t_k, t_{k+1}]$ independently. The path constraints are also relaxed and only evaluated on the time grid $\mathcal{T}$.

The discretized problem is a least-squares nonlinear programming problem (NLP) and can be efficiently solved using a sequential quadratic programming (SQP) approach with Gauss-Newton approximation of the Hessian matrix. In order to reduce the size of each quadratic programming problem (QP), a condensing technique is used (cf. [18] for more details).

B. Real Time Iteration Scheme with Shifting

The computational effort to solve problem (4) exactly can be too demanding for real-time feedback control purposes, especially for mechanical systems with fast-evolving dynamics, as they require faster feedback. The real time iteration (RTI) scheme overcomes this issue by computing only a single Newton-type iteration. Feedback is obtained already during the optimization process and optimality is achieved while the process is alredy being controlled. The

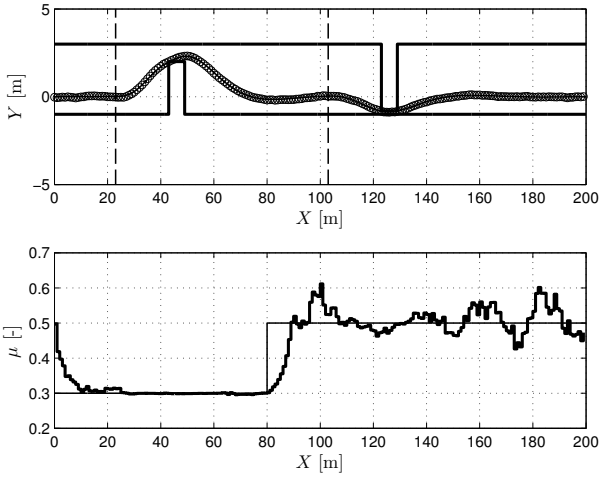


Fig. 3. Vehicle trajectory for a straight reference (thin line) and obstacles (thick line). The MHE estimates are shown in circles. The estimated friction coefficient is displayed in the bottom figure (thick line), together with the actual friction coefficient (thin line).

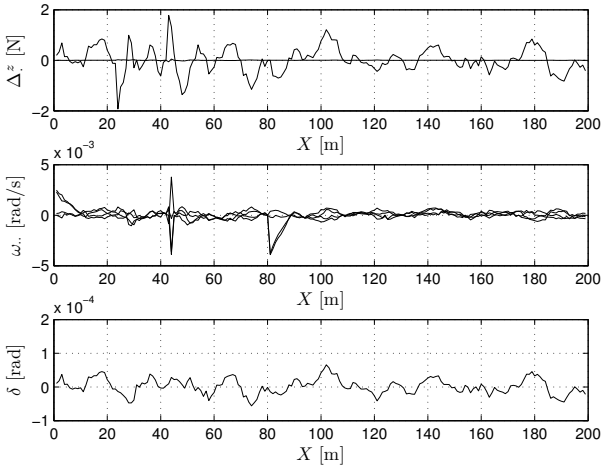


Fig. 4. Estimation error for the load transfer $\Delta\dot{z}$, the rotational velocities $\omega_{\cdot\cdot}$ and the steering angle δ.

RTI scheme also reduces the latency between the instant when the measurements are available and the instant when feedback is available by preparing the computations as much as possible even before the measurements are available. See [5], [4] for a detailed description of the RTI scheme.

Stability proofs for the RTI scheme assume a good initial guess. Starting from the solution of the previous problem, the state and control vectors are first shifted back in time by eliminating the oldest element, then a new element is introduced as guess for the most recent state and control. The guess for the new controls is selected as the inputs sent to the system, while the new state is taken to be the one obtained by applying those inputs and integrating the system. Note that, in the proposed framework, the initial guess for the state is readily available from the controller prediction. In the ideal case, without measurement noise and without model mismatch, the proposed initial guess would directly be the solution to problem (4). In the perturbed case, the guess built on the controller prediction may not be optimal due to noise and model mismatch, but it is always feasible.

C. Code Generation

Once the problem is defined, all dimensions are fixed, dynamic memory allocation can be avoided, and the problem-specific structure can be exploited. Based on a symbolic representation of the dynamic optimization problem, tailored code can be automatically generated, which limits the computations to the essential ones. Automatically generated nonlinear MPC and MHE algorithms have been found to significantly outperform their counterparts implemented in a generic way on several examples [7].

The open-source ACADO Code Generaion tool [8] has been used to export the algorithm into plain C code. The core components of the generated code are: a) a simplified representation of the model right-hand side and its first order derivatives with respect to states and controls, b) a tailored constant stepsize integration method, c) a multiple-shooting discretization scheme, featuring a condensing algorithm and d) a Gauss-Newton method based on the RTI scheme.

D. MHE Tuning

A piecewise-constant parametrization of the process control input ν with $N = 10$ elements of uniform duration $S_{\mathrm{c}} = S_{\mathrm{E}}/N$ is used. The chosen MHE estimation horizon was set to $S_{\mathrm{E}} = 10\,\mathrm{m}$.

The system dynamics f are discretized over the shooting intervals via the implicit Runge-Kutta integration method of order 2 proposed in [20]. The MHE sampling time $S_{\mathrm{s}} = s_k - s_{k-1}$ was chosen equal to the length of the shooting intervals, i.e. $S_{\mathrm{s}} = S_{\mathrm{c}}$. Matrix Σ_y was chosen diagonal. All elements of Σ_y match the square of the inverse of the standard deviation σ_i of the noise they correspond to, i.e. $\{\Sigma_y\}_{i,i} = (\sigma_i)^{-2}$. The standard deviation of the noise is given in Table II.

IV. Simulation results

The proposed MHE scheme has been tested in combination with the NMPC scheme proposed in [9]. All simulations have been run using the code generation tool of ACADO [8], [13], featuring all the aforementioned techniques for the efficient real-time solution of dynamic optimization problems. The generated C code has been compiled in a MEX function and all simulations have been run in Matlab.

In the considered scenario, no information is available about the friction coefficient μ. The reference is a straight trajectory, the vehicle is travelling at $10\ \mathrm{m/s}$ and has to avoid obstacles which become visible only $20\ \mathrm{m}$ in advance [9]. In the proposed scenario, the correct friction coefficient is not known a priori, thus the guess $\mu = 0.5$ is chosen as first initialization. After $80\ \mathrm{m}$, the friction coefficient jumps from the initial value $\mu = 0.3$ to $\mu = 0.5$.

The process measurement noise is summarized in Table II. The trajectory resulting from the control scheme and the estimates on the position are displayed in the top part of Fig. 3. Even in the presence of measurement noise and without prior knowledge of the friction coefficient, the vehicle is able to avoid the obstacles. The friction coefficient estimate is displayed in the bottom part of Fig. 3. It can be noticed that the estimate is good when the vehicle is steering,

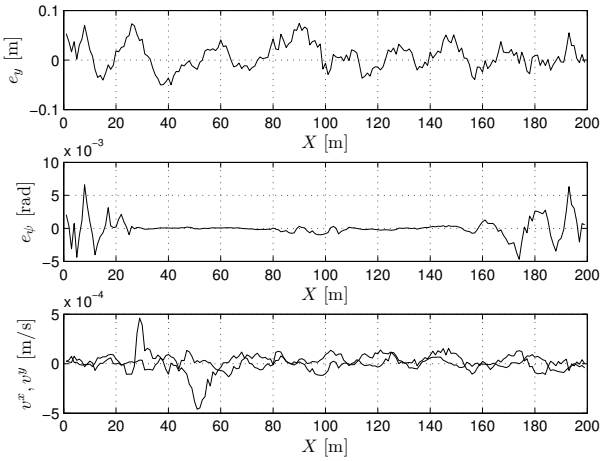


Fig. 5. Estimation error for the distance from the reference e_y and e_ψ, and the longitudinal and lateral velocities v^x and v^y.

while it becomes less accurate on straight trajectories. This is due to the sensitivity of the estimate to the encoder noise and to the loss of observability of the friction coefficient. The simulation results suggest that the friction coefficient is observable only when certain conditions are met. In particular, when no torques are applied to the wheels and the vehicle is not steering, no slip occurs and observability of the friction coefficient is lost. Ongoing research is aimed at finding ways to cope with this observability loss.

The state estimation error is displayed in Fig. 4 and 5. It can be seen that the distance from the reference e_y is sensitive to the measurement noise, while the error for all other state estimates is small.

Simulations have been run on a standard PC, powered by an Intel i7 mobile CPU at 2.8GHz and 8GB RAM. Computational times are displayed in Fig. 6 and are consistently less than 50 ms, thus allowing for a real-time implementation. For each iteration, feedback times take about 10% of the total computational time.

V. Conclusion & Further developments

This paper proposes an estimation algorithm based on a detailed vehicle model which includes wheel dynamics, a Pacejka tire model and load transfer.

Even when dealing with the complexity of a highly nonlinear model, the proposed scheme achieves computational times in the range of milliseconds, with feedback latencies below 10 ms, while successfully estimating the system state and parameters.

The friction coefficient estimation is an involved problem and a more rigorous observability analysis is subject of ongoing research.

The addition of a suspension model to the system dynamics is envisaged to increase the descriptiveness of the model and a parallel implementation of the algorithms to further speedup the computations is being investigated.

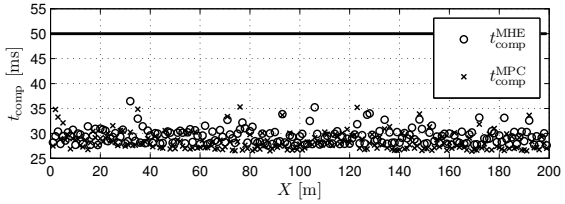


Fig. 6. Computational times t_{comp} for the MHE and NMPC schemes.

References

[1] V.M. Becerra, P.D. Roberts, and G.W. Griffiths. Applying the extended Kalman Filter to systems described by nonlinear differential-algebraic equations. *Control Engineering Practice*, 9:267–281, 2001.

[2] H.G. Bock and K.J. Plitt. A multiple shooting algorithm for direct solution of optimal control problems. In *Proceedings 9th IFAC World Congress Budapest*, pages 243–247. Pergamon Press, 1984.

[3] F. Daum. Nonlinear Filters: Beyond the Kalman Filter. *IEEE A & E Systems Magazine*, 20(8):57–69, 2005.

[4] M. Diehl, H.G. Bock, and J.P. Schlöder. A real-time iteration scheme for nonlinear optimization in optimal feedback control. *SIAM Journal on Control and Optimization*, 43(5):1714–1736, 2005.

[5] M. Diehl, H.G. Bock, J.P. Schlöder, R. Findeisen, Z. Nagy, and F. Allgöwer. Real-time optimization and Nonlinear Model Predictive Control of Processes governed by differential-algebraic equations. *Journal of Process Control*, 12(4):577–585, 2002.

[6] P. Falcone, F. Borrelli, J. Asgari, H.E. Tseng, and D. Hrovat. Low complexity MPC schemes for integrated vehicle dynamics control problems. *9th International Symposium on Advanced Vehicle Control*, 2008.

[7] H.J. Ferreau. *Model Predictive Control Algorithms for Applications with Millisecond Timescales*. PhD thesis, K.U. Leuven, 2011.

[8] H.J. Ferreau, T. Kraus, M. Vukov, W. Saeys, and M. Diehl. High-speed moving horizon estimation based on automatic code generation. In *Proceedings of the 51th IEEE Conference on Decision and Control (CDC 2012), Hawaii*, 2012.

[9] J. V. Frasch, A. J. Gray, M. Zanon, H. J. Ferreau, S. Sager, F. Borrelli, and M. Diehl. An Auto-generated Nonlinear MPC Algorithm for Real-Time Obstacle Avoidance of Ground Vehicles. In *Proceedings of the European Control Conference*, 2013.

[10] Y. Gao, A. Gray, J. V. Frasch, T. Lin, E. Tseng, J.K. Hedrick, and F. Borrelli. Spatial predictive control for agile semi-autonomous ground vehicles. In *Proceedings of the 11th International Symposium on Advanced Vehicle Control*, 2012.

[11] A. Gelb. *Applied Optimal Estimation*. MIT Press, 1974.

[12] A. Gray, M. Zanon, and J. Frasch. Parameters for a Jaguar X-Type. http://www.mathopt.de/RESEARCH/obstacleAvoidance.php, 2012.

[13] B. Houska, H.J. Ferreau, and M. Diehl. An Auto-Generated Real-Time Iteration Algorithm for Nonlinear MPC in the Microsecond Range. *Automatica*, 47(10):2279–2285, 2011.

[14] S.J. Julier, J.K. Uhlmann, and H.F. Durrant-Whyte. A new method for the nonlinear transformation of means and covariances in filters and estimators. *IEEE Transactions on Automatic Control*, 45:477–482, 2000.

[15] R.E. Kalman. A New Approach to Linear Filtering and Prediction Problems. *Transactions of the ASME–Journal of Basic Engineering*, 82:35–45, 1960.

[16] F. Kehrle, J. V. Frasch, C. Kirches, and S. Sager. Optimal control of formula 1 race cars in a VDrift based virtual environment. In S. Bittanti, A. Cenedese, and S. Zampieri, editors, *Proceedings of the 18th IFAC World Congress*, pages 11907–11912, 2011.

[17] U. Kiencke and L. Nielsen. *Automotive Control Systems*. Springer, 2005.

[18] D.B. Leineweber, I. Bauer, A.A.S. Schäfer, H.G. Bock, and J.P. Schlöder. An Efficient Multiple Shooting Based Reduced SQP Strategy for Large-Scale Dynamic Process Optimization (Parts I and II). *Computers and Chemical Engineering*, 27:157–174, 2003.

[19] H. B. Pacejka. *Tyre and Vehicle Dynamics*. Elsevier, 2006.

[20] R. Quirynen, M. Vukov, and M. Diehl. Auto Generation of Implicit Integrators for Embedded NMPC with Microsecond Sampling Times. In Mircea Lazar and Frank Allgöwer, editors, *Proceedings of the 4th IFAC Nonlinear Model Predictive Control Conference*, 2012.

[21] C.V. Rao and J.B. Rawlings. Nonlinear Moving Horizon State Estimation. In F. Allgöwer and A. Zheng, editors, *Nonlinear Predictive Control*, pages 45–69, Basel Boston Berlin, 2000. Birkhäuser.

[22] J.B. Rawlings and B.R. Bakshi. Particle filtering and moving horizon estimation. *Computers and Chemical Engineering*, 30:1529–1541, 2006.