Vehicle Modeling and Parameter Estimation Using Adaptive Limited Memory Joint-State UKF

Changxi You[1] Panagiotis Tsiotras[2]

***Abstract*— Vehicle modeling is an essential part of controller design and validation. This is especially true for model-based control design approaches, such as model-predictive control (MPC), which require an accurate model for predicting the vehicle motion. In this paper we propose a new adaptive joint-state unscented Kalman filter (JUKF) to estimate the unknown vehicle parameters using experimentally collected data. We test the proposed algorithm using three nonlinear vehicle models of increased fidelity: a single-track model, a double-track model and a full 11-dof vehicle model. Simulation results validate the proposed algorithm.**

I. INTRODUCTION

Several vehicle models can be used to represent a vehicle's behavior and help with controller implementation and testing. Such vehicle models are important for the design of advanced control algorithms related to vehicle's active safety. Many models have been proposed for studying the vehicle dynamics and for investigating the handling characteristics of vehicles. Examples include the single track model, the double track model, and the full vehicle model [1]–[4].

The values of the parameters of each vehicle model are critical for accurately predicting the behavior of the vehicle under various operating regimes. The most commonly used techniques for parameter estimation include least-squares fitting [5], robust techniques [5], [6], and various Kalman filtering techniques [5], [7]. Among these, Kalman filtering is especially suitable for problems where the measurements are collected in a sequential manner.

The extended Kalman filter (EKF) and the unscented Kalman filter (UKF) are probably the most popular filters used for system identification of nonlinear systems. Nevertheless, they can only achieve good performance under some prior knowledge including: 1) an accurate system model, 2) complete information of noise statistics, and 3) properly selected initial conditions, all of which may be either not accurate or not available in practice [8]. A commonly used approach to solve these problems is to make the Kalman filters work adaptively, by dynamically modifying the filtering algorithm using various schemes [9], [10].

The adaptive limited memory filter (ALMF) in [10] estimates the process and observation noise statistics on-line, based on the past state estimations and observations. This algorithm improves the state estimation performance at little computational expense. Nonetheless, the ALMF was derived using a linear system model, and it has not been validated in parameter estimation applications, at least as far as the authors know. This paper builds on the work in [10], and develops a new adaptive limited memory unscented Kalman filter (ALM-UKF) for *nonlinear* applications. We apply the ALM-UKF to estimate the vehicle model parameters for three different vehicle models using both simulation data and experimental data.

[1]C. You is PhD candidate at the School of Aerospace Engineering, Georgia Institute of Technology, Atlanta. GA 30332-0150. USA. Email: cyou6@gatech.edu

[2]P. Tsiotras is a Professor at the School of Aerospace Engineering and the Institute for Robotics & Intelligent Machines, Georgia Institute of Technology, Atlanta. GA 30332-0150. USA. Email: tsiotras@gatech.edu

II. VEHICLE MODELING

In this section, we describe the three vehicle models used in this work, namely, the single-track model, the double-track model and a full 11-dof vehicle model [1]–[4].

A. Single-Track Model

The single-track model takes into consideration the longitudinal and lateral displacement, as well as the yaw motion of the vehicle, as shown in Fig. 1.

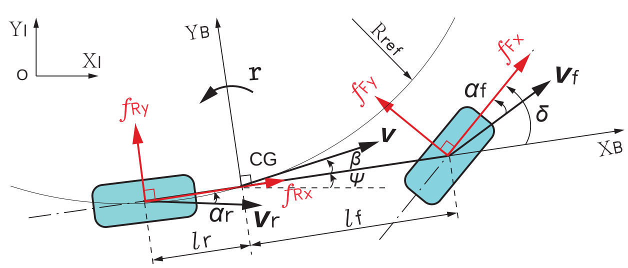

Fig. 1. Single-track vehicle model.

We use $X_{\rm I} - O - Y_{\rm I}$ and $X_{\rm B} - CG - Y_{\rm B}$ to denote the inertial frame fixed on the ground and the body frame fixed on the vehicle, respectively. The equations of motion of the model can be expressed in a body-fixed frame with the origin at CG as follows [1]:

$$\dot{V}_{\rm x} = (f_{\rm Fx}\cos\delta - f_{\rm Fy}\sin\delta + f_{\rm Rx})/m + V_{\rm y}\dot{\psi}, \tag{1a}$$
$$\dot{V}_{\rm y} = (f_{\rm Fx}\sin\delta + f_{\rm Fy}\cos\delta + f_{\rm Ry})/m - V_{\rm x}\dot{\psi}, \tag{1b}$$
$$\dot{r} = \big((f_{\rm Fy}\cos\delta + f_{\rm Fx}\sin\delta)\ell_{\rm f} - f_{\rm Ry}\ell_{\rm r}\big)/I_{\rm z}, \tag{1c}$$

where $V_{\rm x}$ and $V_{\rm y}$ are the components of V along the $X_{\rm B}$ and $Y_{\rm B}$ directions, respectively; m is the total mass, and $I_{\rm z}$ is the moment of inertia of the vehicle about the vertical axis. f_{ij} ($i = F, R$ and $j = x, y$) denote the longitudinal and lateral friction forces at the front and rear wheels, ψ denotes the yaw angle, and δ is the steering angle of the front wheel.

B. Double-Track Model

The double-track model takes into consideration the longitudinal, lateral and yaw motion of the vehicle, but considers the load difference between the left and right wheels arising from the lateral load transfer. We use $f_{i,j,k}$ ($i = L, R$, $j = L, R$ and $k = x, y$) to denote the longitudinal or lateral friction force for each wheel, respectively. The vehicle's equations of motion are then given by:

$$\dot{V}_{\rm x} = \big((f_{\rm LFx} + f_{\rm RFx})\cos\delta - (f_{\rm LFy} + f_{\rm RFy})\sin\delta + f_{\rm LRx} + f_{\rm RRx}\big)/m + V_{\rm y}\dot{\psi}, \tag{2a}$$

$$\dot{V}_{\rm y} = \big((f_{\rm LFx} + f_{\rm RFx})\sin\delta + (f_{\rm LFy} + f_{\rm RFy})\cos\delta + f_{\rm LRy} + f_{\rm RRy}\big)/m - V_{\rm x}\dot{\psi}, \tag{2b}$$

$$\dot{r} = \Big(\big((f_{\rm LFy} + f_{\rm RFy})\cos\delta + (f_{\rm LFx} + f_{\rm RFx})\sin\delta\big)\ell_{\rm f} - (f_{\rm LRy} + f_{\rm RRy})\ell_{\rm r}\Big)/I_{\rm z}. \tag{2c}$$

C. Full Vehicle Model

The full vehicle model considers the dynamics of the sprung and unsprung mass of the vehicle separately. The equations of motion for the total mass are the same as (2a)-(2c) for the double-track model. We also take the air resistance into account and modify (2a) as follows

$$\dot{V}_{\rm x} = \big((f_{\rm LFx} + f_{\rm RFx})\cos\delta - (f_{\rm LFy} + f_{\rm RFy})\sin\delta + f_{\rm LRx} + f_{\rm RRx}\big)/m + V_{\rm y}\dot{\psi} - C_{\rm D}\rho_{\rm air}AV_{\rm x}^2/2, \tag{3}$$

where $C_{\rm D}$ is the air resistance coefficient, ρ_{air} is the air density, and A is the frontal area of the vehicle. The vertical translation is accounted for by a riding model as shown in Fig. 2. The rolling and pitching model are given in Fig. 3.

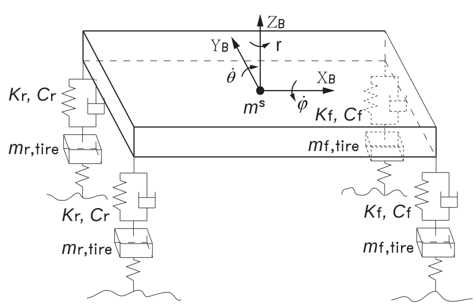

Fig. 2. Riding model.

In Fig. 2, K_i and C_i ($i = {\rm f}, {\rm r}$) denote the spring stiffness and the damping coefficient of the suspension system related to each wheel, $m_{i,{\rm tire}}$ ($i = {\rm f}, {\rm r}$) denotes the mass of the front and rear tire, respectively, $m^{\rm s}$ is the sprung mass, and $\dot{\phi}$ and $\dot{\theta}$ are the rolling and pitching rate, respectively.

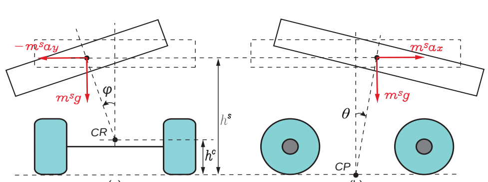

Fig. 3. Rolling and pitching model.

Fig. 3(a) shows the rolling motion arising from the lateral acceleration and the gravity center offset from the rolling center. The parameters $h^{\rm s}$ and $h^{\rm c}$ are the heights of the sprung mass center and the rolling center (CR), respectively. Fig. 3(b) shows the pitching motion arising from the longitudinal acceleration and the gravity center offset from the pitching center (CP) that is assumed to be on the ground. The dynamical equations of the vertical, rolling and pitching motion of the sprung mass are given by

$$\dot{V}_{\rm z}^{\rm s} = \Big(-2(K_{\rm f} + K_{\rm r})\theta - 2(C_{\rm f} + C_{\rm r})V_{\rm z}^{\rm s} + 2(\ell_{\rm f}K_{\rm f} - \ell_{\rm r}K_{\rm r})\phi + 2(\ell_{\rm f}C_{\rm f} - \ell_{\rm r}C_{\rm r})\dot{\theta}\Big)/m^{\rm s}, \tag{4a}$$

$$\ddot{\theta} = \Big(2(\ell_{\rm f}K_{\rm f} - \ell_{\rm r}K_{\rm r})z^{\rm s} + 2(\ell_{\rm f}C_{\rm f} - \ell_{\rm r}C_{\rm r})V_{\rm z}^{\rm s} - 2(\ell_{\rm f}^2K_{\rm f} + \ell_{\rm r}^2K_{\rm r})\theta - 2(\ell_{\rm f}^2C_{\rm f} + \ell_{\rm r}^2C_{\rm r})\dot{\theta} + m^{\rm s}gh^{\rm s}\sin\theta + m^{\rm s}a_{\rm x}^{\rm s}h_s\cos\theta\Big)/I_{\rm y}^{\rm P}, \tag{4b}$$

$$\ddot{\phi} = \Big(-w_{\rm f}^2K_{\rm f}\phi/2 - w_{\rm f}^2C_{\rm f}\dot{\phi}/2 - w_{\rm r}^2K_{\rm r}\phi/2 - w_{\rm r}^2C_{\rm r}\dot{\phi}/2 + m^{\rm s}g(h^{\rm s} - h^{\rm c})\sin\phi + m^{\rm s}a_{\rm y}^{\rm s}(h^{\rm s} - h^{\rm c})\cos\phi\Big)/I_{\rm x}^{\rm R}, \tag{4c}$$

where $w_{\rm i}$ ($i = {\rm f}, {\rm r}$) denote the front and rear track , respectively; $a_{\rm x}^{\rm s}$ and $a_{\rm y}^{\rm s}$ are the longitudinal and lateral acceleration of the sprung mass center in the body-fixed frame, and $I_{\rm x}^{\rm R}$ and $I_{\rm y}^{\rm P}$ are the moments of inertia of the sprung mass about the rolling axis and the pitching axis, respectively.

D. Tire Force Model

The tire slip is defined by the non-dimensional relative velocity of each tire with respect to the road, as follows

$$s_{ijx} = \frac{V_{ijx} - \omega_{ijx}R_j}{\omega_{ijx}R_j}, \qquad s_{ijy} = \frac{V_{ijy}}{\omega_{ijx}R_j}, \tag{5}$$

where $i = L, R$ and $j = F, R$. V_{ijk} ($k = x, y$) is the tire frame component of the vehicle velocity of each tire. The total slip of each tire is defined by $s_{ij} = \sqrt{s_{ijx}^2 + s_{ijy}^2}$. The total friction coefficient related to each tire is calculated using Pacejka's "magic formula" (MF) as follows [1], [3]:

$$\mu_{ij} = D\sin\Big(C{\rm atan}\Big(BS_{\rm E} - E\big(BS_{\rm E} - {\rm atan}S_{\rm E}\big)\Big)\Big) + S_{\rm v} \tag{6}$$

where B, C, D, E are the stiffness, shape, peak and curvature factors, respectively; $S_{\rm E} = s_{ij} - S_{\rm h}$, where $S_{\rm h}$ is the horizontal shift. $S_{\rm v}$ is the vertical shift. The tire friction force components are given by

$$f_{ijk} = -\frac{s_{ijk}}{s_{ij}}\mu_{ij}f_{ijz}, \quad i = L, R;\; j = F, R;\; k = x, y. \tag{7}$$

where f_{ijz} is the normal load on the corresponding tire and can be calculated following [1]. We do not show the details on the calculation for f_{ijz} due to lack of space.

III. Unscented Kalman Filter

The joint-state UKF includes the unknown parameters into the original state vector and estimates the new augmented state. The state and the noise are assumed to be Gaussian random variables. Recall that for a system given by

$$x_{k+1} = f(x_k, u_k) + w_k, \qquad y_k = h(x_k, u_k) + v_k, \tag{8}$$

where $w_k \sim N(q, Q)$ and $v_k \sim N(r, R)$ are Gaussian process and measurement noise, respectively, the EKF propagates the Gaussian random variable x_k by linearizing the nonlinear state transition (observation) function $f : \mathbb{R}^n \times$

$\mathcal{U} \mapsto \mathbb{R}^n$ ($h : \mathbb{R}^n \times \mathcal{U} \mapsto \mathbb{R}^m$) with the Jacobian matrix at each time step k [11]. Instead of an EKF, this paper adopts a UKF filter since: a) the UKF propagates the Gaussian random variable through a nonlinear function more accurately than the EKF; and b) The UKF avoids calculating the Jacobians that may be too cumbersome for highly nonlinear systems.

A. Standard UKF

The UKF is based on the unscented transformation (UT), and avoids calculating the Jacobian matrices at each time step. Assuming an L-dimensional Gaussian random variable x with mean $\hat{x}$ and covariance P_x, to calculate the statistics of $y = g(x)$, we select $2L+1$ discrete sample points $\{\mathcal{X}_i\}_{i=0}^{2L}$ which are propagated through the system dynamics. The UKF redefines the state vector as $x_k^a = [x_k^{\mathrm{T}}, w_k^{\mathrm{T}}, v_k^{\mathrm{T}}]^{\mathrm{T}}$, which concatenates the original state and noise variables, and then estimates x_k^a recursively [12].

B. Adaptive Limited Memory UKF

We propose a new estimation algorithm for nonlinear systems called Adaptive Limited Memory UKF (ALM-UKF). First, recall that the adaptive Kalman filter algorithm [10] adjusts the mean and the covariance of the noise online, which is expected to compensate for time-varying modeling errors. Define the set of unknown time-varying hyperparameters for the Kalman filter corresponding to the noise statistics at the i^{th} time step, as $\mathcal{S}_i \triangleq \{q_i, Q_i, r_i, R_i\}$. $\mathcal{S}_i$ is estimated simultaneously with the system state and parameters. Since an optimal estimator for $\mathcal{S}_i$ does not exist, and many suboptimal schemes are either too restrictive for nonlinear applications or too computationally demanding [8], [13], this paper adopts the adaptive limited memory algorithm in [10], with the following two extensions: a) the algorithm is developed for a nonlinear application (i.e., UKF); b) we wish to estimate the unknown parameters of the system along with the state, instead of just the system state. In the following, we assume that $\mathcal{S}_i$ is constant and is denoted by $\mathcal{S} = \{q, Q, r, R\}$.

For the observation noise statistics r and R, we consider the nonlinear observation at time k, which is given by $y_k = h(x_k, u_k) + v_k$. Since the true value of x_k is unknown, v_k is approximated by

$$r_k = y_k - \hat{h}(x_k, u_k), \tag{9}$$

where r_k represents a sample of the observation noise v at time k, and

$$\hat{h}(x_k, u_k) = \sum_{i=0}^{2L} W_i^{(m)} h(\mathcal{X}_{i,k}^x, u_k) \triangleq \hat{h}_k. \tag{10}$$

We define a new random variable $\xi \sim (r, C_r)$, and assume that there are N samples r_k ($k = 1, \ldots, N$), such that the r_k's are N empirical measurements for ξ. An unbiased estimator for r can be given by the sample mean

$$\hat{r} = \frac{1}{N} \sum_{k=1}^{N} r_k, \tag{11}$$

where the term "unbiased" implies that $\mathbb{E}[\hat{r}] = \mathbb{E}[\xi] = r$. An unbiased estimator for the covariance of ξ can be given by

$$\hat{C}_r = \frac{1}{N-1} \sum_{k=1}^{N} (r_k - \hat{r})(r_k - \hat{r})^{\mathrm{T}}, \tag{12}$$

where the term "unbiased" implies $\mathbb{E}[\hat{C}_r] = \mathbb{E}[(\xi - r)(*)^{\mathrm{T}}]$. For simplicity, in this paper we use $(*)$ to represent repeated terms when necessary. Since $y_k = h(x_k, u_k) + v_k$, it follows from (9) that

$$r_k = h(x_k, u_k) - \hat{h}_k + v_k. \tag{13}$$

We can therefore calculate the covariance of ξ as follows

$$\begin{aligned} \mathbb{E}[(\xi - r)(*)^{\mathrm{T}}] &= \frac{1}{N} \sum_{k=1}^{N} \mathbb{E}[(r_k - r)(*)^{\mathrm{T}}] \\ &= \frac{1}{N} \sum_{k=1}^{N} \mathbb{E}[(h(x_k, u_k) - \hat{h}_k + v_k - r)(*)^{\mathrm{T}}] \\ &= \frac{1}{N} \sum_{k=1}^{N} \left(\mathbb{E}\big[(h(x_k, u_k))(*)^{\mathrm{T}}\big] - (\hat{h}_k)(*)^{\mathrm{T}}\right) + R. \end{aligned} \tag{14}$$

where

$$\mathbb{E}[(h(x_k, u_k))(*)^{\mathrm{T}}] = \sum_{i=0}^{2L} W_i^{(m)} (h(\mathcal{X}_{i,k}^x, u_k))(*)^{\mathrm{T}}. \tag{15}$$

Note that we assume that x_k and v_k are independent in (14). An unbiased estimate of R is given following (12) and (14):

$$\hat{R} = \frac{1}{N-1} \sum_{k=1}^{N} \Big((r_k - \hat{r})(*)^{\mathrm{T}} - \frac{N-1}{N} \Big(\mathbb{E}[(h(x_k, u_k)) (*)^{\mathrm{T}}] - (\hat{h}_k)(*)^{\mathrm{T}} \Big)\Big). \tag{16}$$

For the process noise statistics q and Q, we consider the nonlinear state propagation at time k, which is given by $x_k = f(x_{k-1}, u_{k-1}) + w_{k-1}$. Since the true values of x_k and x_{k-1} are unknown, w_{k-1} is approximated by

$$q_k = \hat{x}_k - \hat{f}(x_{k-1}, u_{k-1}), \tag{17}$$

where q_k represents a sample of the process noise w at time step $k-1$, and

$$\hat{f}(x_{k-1}, u_{k-1}) = \sum_{i=0}^{2L} W_i^{(m)} f(\mathcal{X}_{i,k-1}^x, u_{k-1}) \triangleq \hat{f}_{k-1}. \tag{18}$$

We define a new random variable $\zeta \sim (q, C_q)$, and assume that there are M samples q_k ($k = 1, \ldots, M$), where the q_k's are M empirical measurements for ζ. An unbiased estimator for the mean value of ζ is given by the sample mean

$$\hat{q} = \frac{1}{M} \sum_{k=1}^{M} q_k. \tag{19}$$

An unbiased estimator for the covariance of ζ is given by

$$\hat{C}_q = \frac{1}{M-1} \sum_{k=1}^{M} (q_k - \hat{q})(q_k - \hat{q})^{\mathrm{T}}, \tag{20}$$

such that $\mathbb{E}[\hat{C}_q] = \mathbb{E}[(\zeta - q)(*)^{\mathrm{T}}]$. We then calculate the covariance Q by the following equation

$$\mathbb{E}[(w_{k-1} - q)(*)^{\mathrm{T}}] = \mathbb{E}[(w_{k-1} - q_k + q_k - q)(*)^{\mathrm{T}}]$$

$$= \frac{1}{M}\sum_{k=1}^{M}\mathbb{E}\Big[\Big((x_k - \hat{x}_k) - (f(x_{k-1}, u_{k-1}) - \hat{f}_{k-1}) + (q_k - q)\Big)\Big(*\Big)^{\mathrm{T}}\Big]$$

$$= \frac{1}{M}\sum_{k=1}^{M}\Big(\mathbb{E}[(f(x_{k-1}, u_{k-1}))(*)^{\mathrm{T}}] - (\hat{f}_{k-1})(*)^{\mathrm{T}} - P_k\Big) + \mathbb{E}[\hat{C}_q], \tag{21}$$

where

$$\mathbb{E}[(f(x_{k-1}, u_{k-1}))(*)^{\mathrm{T}}] = \sum_{i=0}^{2L} W_i^{(m)}(f(\mathcal{X}^x_{i,k-1}, u_{k-1}))(*)^{\mathrm{T}}. \tag{22}$$

Then Q can be estimated unbiasedly following the equations (20)-(21),

$$\hat{Q} = \frac{1}{M-1}\sum_{k=1}^{M}\Big((q_k - \hat{q})(*)^{\mathrm{T}} + \frac{M-1}{M}\Big(\mathbb{E}[(f(x_{k-1}, u_{k-1}))(*)^{\mathrm{T}}] - (\hat{f}_{k-1})(*)^{\mathrm{T}} - P_k\Big)\Big). \tag{23}$$

Equations (11), (16), (19) and (23) provide unbiased estimates for r, R, q and Q, which are based on N observation noise samples and M process noise samples, respectively. All samples r_k and q_k are assumed to be statistically independent and identically distributed. We summarize the algorithm of the Adaptive Limited Memory UKF based on equations (9)-(23) in Algorithm I, where α, β, κ and λ are the UT parameters [12].

C. Experimental Platform

In the following, we use data collected from a fifth-scale Auto-Rally vehicle (see Fig. 4) and estimate the unknown parameters of all three of the vehicle models. Table I (bottom) summarizes the unknown parameters to be estimated for the three different vehicle models.

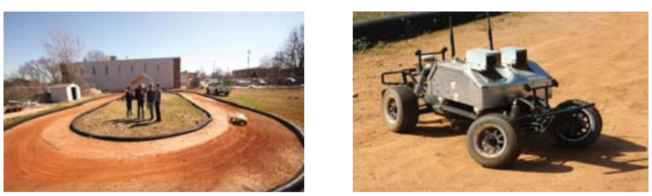

Fig. 4. The test track (left) and the Auto-Rally vehicle model (right).

The fifth-scale Auto-Rally vehicle is driven by two rear wheels. The known parameters of the Auto-Rally vehicle model are given in Table I (top).

D. Parameter Estimation

We use a joint-state UKF to estimate the unknown parameters of the system. For the system given in (8), we introduce the following dynamics for the parameter vector p,

$$p_{k+1} = p_k + w_k^{\mathrm{p}}, \tag{24}$$

where $w_k^{\mathrm{p}} \sim N(q^{\mathrm{p}}, Q^{\mathrm{p}})$ is Gaussian process noise. We define the augmented state as $x^{\mathrm{a}} = [x^{\mathrm{T}}, p^{\mathrm{T}}]^{\mathrm{T}}$. It then follows from (8) and (24) that

$$x^{\mathrm{a}}_{k+1} = F(x^{\mathrm{a}}_k, u_k) + w^{\mathrm{a}}_k, \qquad y_k = H(x^{\mathrm{a}}_k, u_k) + v_k, \tag{25}$$

where $w^{\mathrm{a}}_k = [w_k^{\mathrm{T}}, (w_k^{\mathrm{p}})^{\mathrm{T}}]^{\mathrm{T}}$.

Algorithm I: Adaptive Limited Memory UKF

0: UT parameters setup:

$\lambda = \alpha^2(L+\kappa) - L$

$W_0^{(m)} = \lambda/(L+\lambda)$

$W_0^{(c)} = \lambda/(L+\lambda) + 1 - \alpha^2 + \beta$

$W_i^{(m)} = W_i^{(c)} = 0.5/(L+\lambda), \qquad i = 1, \ldots, 2L$

$\gamma = \sqrt{L+\lambda}$

1: Initialize with:

$\hat{x}_0 = \mathbb{E}[x_0],\ \hat{q}_0 = \mathbb{E}[q_0],\ \hat{r}_0 = \mathbb{E}[r_0]$

$P_0 = \mathbb{E}[(x_0 - \hat{x}_0)(*)^{\mathrm{T}}]$

$Q_0 = \mathbb{E}[(q_0 - \hat{q}_0)(*)^{\mathrm{T}}]$

$R_0 = \mathbb{E}[(r_0 - \hat{r}_0)(*)^{\mathrm{T}}]$

$\hat{x}_0^a = \mathbb{E}[x_0^a] = [\hat{x}_0^{\mathrm{T}}\ \ \hat{q}_0^{\mathrm{T}}\ \ \hat{r}_0^{\mathrm{T}}]^{\mathrm{T}}$

$P_0^a = \mathbb{E}[(x_0^a - \hat{x}_0^a)(*)^{\mathrm{T}}] = \begin{bmatrix} P_0 & 0 & 0 \\ 0 & Q_0 & 0 \\ 0 & 0 & R_0 \end{bmatrix}$

2: Sigma-point calculation and prediction:

$\mathcal{X}^a_{k-1} = [\hat{x}^a_{k-1} \quad \hat{x}^a_{k-1} + \gamma\sqrt{P^a_{k-1}} \quad \hat{x}^a_{k-1} - \gamma\sqrt{P^a_{k-1}}]$

$\mathcal{X}^x_{k|k-1} = f(\mathcal{X}^x_{k-1}, u_{k-1}) + \mathcal{X}^w_{k-1}$

$\hat{x}_k^- = \sum_{i=0}^{2L} W_i^{(m)} \mathcal{X}^x_{i,k|k-1}$

$P_k^- = \sum_{i=0}^{2L} W_i^{(c)} (\mathcal{X}^x_{i,k|k-1} - \hat{x}_k^-)(*)^{\mathrm{T}}$

$\mathcal{Y}_{k|k-1} = h(\mathcal{X}^x_{k|k-1}, u_{k-1}) + \mathcal{X}^v_{k|k-1}$

$\hat{y}_k^- = \sum_{i=0}^{2L} W_i^{(m)} \mathcal{Y}_{i,k|k-1}$

3: Observation noise estimation ($k \geq N$):

$r_k = y_{k-1} - \hat{h}_{k-1}$

$\Gamma_k = \sum_{i=0}^{2L} W_i^{(m)} (h(\mathcal{X}^x_{i,k-1}, u_{k-1}))(*)^{\mathrm{T}} - (\hat{h}_{k-1})(*)^{\mathrm{T}}$

$\hat{r}_k = \hat{r}_{k-1} + \frac{1}{N}(r_k - r_{k-N})$

$R_k = R_{k-1} + \frac{1}{N-1}\Big((r_k - \hat{r}_k)(*)^{\mathrm{T}} - (r_{k-N} - \hat{r}_k)(*)^{\mathrm{T}} + \frac{1}{N}(r_k - r_{k-N})(*)^{\mathrm{T}} + \frac{N-1}{N}(\Gamma_{k-N} - \Gamma_k)\Big)$

4: Measurement update:

$P_{y_k y_k} = \sum_{i=0}^{2L} W_i^{(c)} (\mathcal{Y}_{i,k|k-1} - \hat{y}_k^-)(*)^{\mathrm{T}}$

$P_{x_k y_k} = \sum_{i=0}^{2L} W_i^{(c)} (\mathcal{X}^x_{i,k|k-1} - \hat{x}_k^-)(\mathcal{Y}_{i,k|k-1} - \hat{y}_k^-)^{\mathrm{T}}$

$\mathcal{K} = P_{x_k y_k} P_{y_k y_k}^{-1}$

$\hat{x}_k = \hat{x}_k^- + \mathcal{K}(y_k - \hat{y}_k^-)$

$P_k = P_k^- - \mathcal{K} P_{y_k y_k} \mathcal{K}^{\mathrm{T}}$

5: Process noise estimation ($k \geq M$):

$q_k = \hat{x}_k - \hat{f}_{k-1}$

$\Pi_k = \sum_{i=0}^{2L} W_i^{(m)} \big(f(\mathcal{X}^x_{i,k-1}, u_{k-1})\big)\big(*\big)^{\mathrm{T}} - (\hat{f}_{k-1})(*)^{\mathrm{T}} - P_k$

$\hat{q}_k = \hat{q}_{k-1} + \frac{1}{M}(q_k - q_{k-M})$

$Q_k = Q_{k-1} + \frac{1}{M-1}\Big((q_k - \hat{q}_k)(*)^{\mathrm{T}} - (q_{k-N} - \hat{q}_k)(*)^{\mathrm{T}} + \frac{1}{M}(q_k - q_{k-N})(*)^{\mathrm{T}} - \frac{M-1}{M}(\Pi_{k-M} - \Pi_k)\Big)$

Note: $x^a = [x^{\mathrm{T}}\ w^{\mathrm{T}}\ v^{\mathrm{T}}]^{\mathrm{T}}$, $\mathcal{X}^a = [(\mathcal{X}^x)^{\mathrm{T}}\ (\mathcal{X}^w)^{\mathrm{T}}\ (\mathcal{X}^v)^{\mathrm{T}}]^{\mathrm{T}}$.

It is noticed that $\hat{R}$ and $\hat{Q}$ in (16) and (23) may become negative definite during the process of the implementation (this is also mentioned in [10]). This paper calculates the nearest positive definite matrices of $\hat{R}$ or $\hat{Q}$ when negative eigenvalues of $\hat{R}$ or $\hat{Q}$ are observed, such that a symmetric positive definite matrix nearest to $\hat{R}$ or $\hat{Q}$ in terms of the Frobenius norm can be obtained [14].

It is worth mentioning that the artificial Gaussian process noise w_k^{p} in (24) is used to change the parameter p when the UKF is running. However, if the value of w_k^{p} is large, the parameter p will be changed by a large amount at each time step. This condition may further cause the filter to diverge, since the parameterized vehicle models in Section II are sensitive to p and may thus become unstable for unreasonable values of p. We addressed this problem by rescaling the diagonal entries of Q^{p} to be some small positive values at each time step. Other discussions on the numerical instability problems of the UKF can be found in [10], [14].

IV. Results and Discussion

In this section we implement the proposed filter, show and validate the results from the parameter estimations using the standard UKF and the ALM-UKF, respectively.

A. Standard UKF

We first implemented the standard joint-state UKF using the three different vehicle models in Section II, respectively. The hyperparameters of the filter are critical for the filter design, especially the process noise covariance Q [15]. In this section, we tune the diagonal elements of these matrices recursively, until the parameterized vehicle model shows satisfactory simulation results.

We selected 113 seconds of experimental data of the Auto-Rally vehicle. The first 100 seconds data were used to tune the hyperparameters and estimate the vehicle parameters, and the remaining 13 seconds (a complete cycle around the testing track) were used to validate the results. Fig. 5 shows the estimates for several selected states of the system for the single-track model. It can be seen that the estimates of the states agree well with the data. The results for the other vehicle models were similar and hence are omitted.

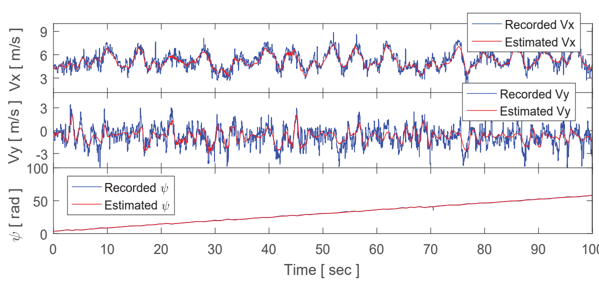

Fig. 5. State estimation for the single-track model using JUKF.

TABLE I

Known / Unknown Vehicle model parameters.

$m[kg]$	21.5	total mass	$m^{\mathrm{s}}[kg]$	18.03	sprung mass
$m_{\mathrm{f}}[kg]$	0.84	front wheel mass	$m_{\mathrm{r}}[kg]$	0.89	rear wheel mass
$w_{\mathrm{f}}[m]$	0.44	front track	$w_{\mathrm{r}}[m]$	0.46	rear track
$L[m]$	0.57	wheel base	$R[m]$	0.095	wheel radius

Tire forces model	$B, C, D, E, S_{\mathrm{h}}, S_{\mathrm{v}}$
Single/Double-track model	$I_{\mathrm{z}}, \ell_{\mathrm{f}}, h, g_{\mathrm{s}}^{*}$
Full vehicle model	$I_{\mathrm{z}}, \ell_{\mathrm{f}}, h, g_{\mathrm{s}}^{*}, C_{\mathrm{D}}, I_{\mathrm{x}}^{R}, I_{\mathrm{y}}^{P}, K_{\mathrm{f}}, K_{\mathrm{r}}, C_{\mathrm{f}}, C_{\mathrm{r}}, h^{\mathrm{c}}$

* g_{s} is the gear ratio defined by the steering command divided by δ

Next, we validated the estimated parameters in simulation. This was done in order to ensure that the parameters we obtained were able to satisfactorily reproduce the data, hence accurately predicting the vehicle's motion in practical applications. Fig. 6 shows the simulated trajectories for different vehicle models configured with the estimated parameters. The results in Fig. 6 indicate that the larger the number of degrees of freedom (DoF) of the model, the more accurate the results and the better the agreement with data.

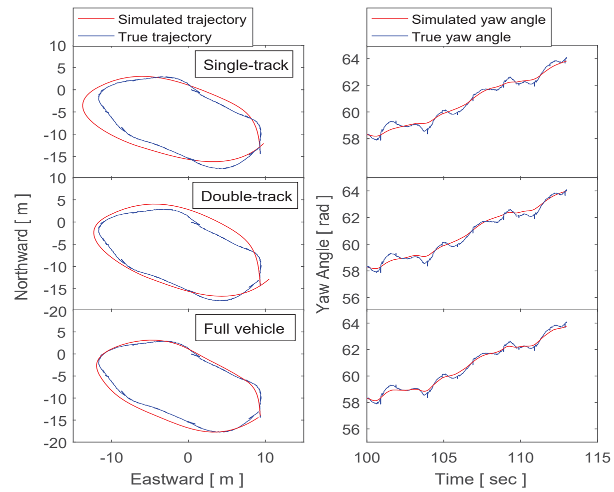

Fig. 6. Simulation results of the estimated vehicle models using standard UKF.

The process of manually tuning the hyperparameters of the UKF one-by-one until we achieve good performance is time consuming, and can be done only off-line.

B. Adaptive Limited Memory UKF

Instead of tuning the noise, we implemented the ALM-UKF to find the suboptimal estimation of the noise statistics on-line, during which the augmented-state and the noise are estimated simultaneously. Both simulation data collected using CarSim and experimental data collected with the Auto-Rally vehicle were used to validate Algorithm I. The noise samples at each time step k are from the estimation based on the last 10 seconds of data (defined by N and M in Algorithm I).

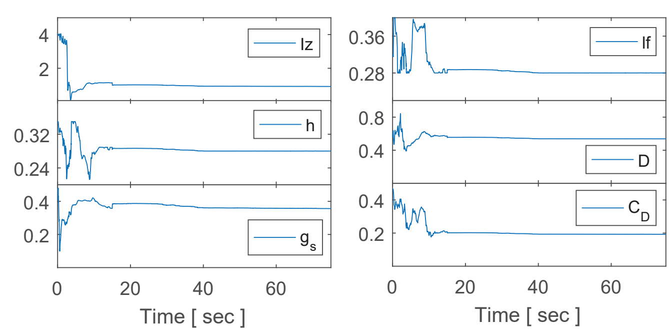

Fig. 7. Convergence of the vehicle parameters along with the estimation process.

The estimation of the states (i.e., the velocities, yaw angle and positions) is not difficult. We thus only show the estimation results of the unknown vehicle parameters. We implemented the adaptive limited memory joint-state UKF (ALM-JUKF) to estimate a full vehicle model's parameters using the Auto-Rally experimental data. Fig. 7 shows the time trajectories of several parameters (see Table I) during the estimation process, where all the parameters converge fast and get stabilized after about 20 seconds.

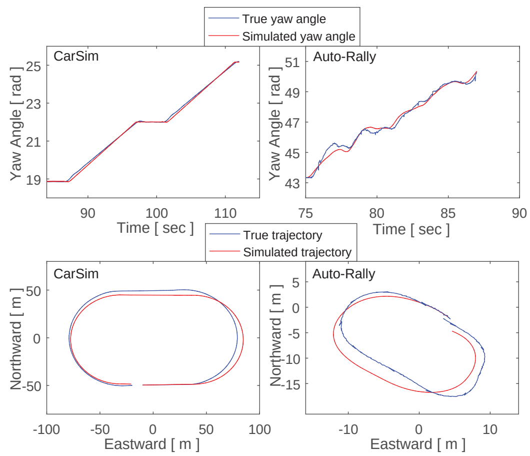

Fig. 8. Simulation results of the estimated vehicle models using ALM-JUKF.

Since CarSim provides a complete full-scale vehicle model and data from the simulation using CarSim show little irregular noise, the ALM-JUKF was also implemented using simulated data for validation purposes. In Fig. 8, we show the estimated vehicle parameters corresponding to the CarSim vehicle model and the Auto-Rally vehicle model from simulations. We simulated a full vehicle model using the estimated parameters and compared the model output with data. It can be seen that, as expected, the identified vehicle model can satisfactorily reproduce the data, especially when the system uses simulated data. Data collected using the Auto-Rally vehicle show obvious non-Gaussian noise which may have some effect on the estimation process.

Compared with the results in Fig. 6, the simulated trajectories of the Auto-Rally vehicle in Fig. 8 show larger deviation from the data. The reason may be that we tuned the estimation of the noise statistics of the standard UKF to be optimal (in some degree), but Algorithm I was using a suboptimal estimator for the noise statistics. The advantages of Algorithm I are, of course, that it is more efficient and can work on-line. We also expect that Algorithm I is especially useful for time-varying parameters estimation problems (i.e., estimation of a linear parameter varying (LPV) driver model).

V. Conclusions

In this paper we introduced three vehicle models, namely, a single-track model, a double-track model and a full vehicle model, and we estimated the model parameters using the joint-state UKF algorithms based on both simulation and experimental data. The design of the standard joint-state UKF is hindered by the lack of knowledge of the unknown noise statistics. By tuning the noise statistics of the standard joint-state UKF, satisfactory estimates of the model parameters can be obtained, but the tuning process is time consuming and hence can only be implemented off-line. We introduced an adaptive limited memory joint-state UKF algorithm (ALM-JUKF), which estimates the system state, model parameters and the Kalman filter hyperparameters related to the noise simultaneously, hence making possible to provide on-line estimates of the model parameters. The algorithm was validated with both CarSim simulation data and experimental data from a fifth-scale Auto-Rally vehicle.

Acknowledgment

This work was supported by National Science Foundation awards CMMI-1234286 and CPS-1544814. The authors would also like to thank G. Williams & B. Goldfain for providing the experimental data from the Auto-Rally platform.

References

[1] E. Velenis, E. Frazzoli, and P. Tsiotras, "Steady-state cornering equilibria and stabilisation for a vehicle during extreme operating conditions," *International Journal of Vehicle Autonomous Systems*, vol. 8, no. 2-4, pp. 217–241, 2010.

[2] K. Lundahl, J. Åslund, and L. Nielsen, "Investigating vehicle model detail for close to limit maneuvers aiming at optimal control," in *22nd International Symphosium of Dynamic on Vehicle*, Manchester Metropolitan University, Manchester, UK, Aug. 14-19 2011.

[3] M. S. Burhaumudin, P. M. Samin, H. Jamaluddin, R. A. Rahman, S. Sulaiman, *et al.*, "Integration of magic formula tire model with vehicle handling model," *International Journal of Research in Engineering and Technology*, vol. 1, no. 3, pp. 139–145, 2012.

[4] J. D. Setiawan, M. Safarudin, and A. Singh, "Modeling, simulation and validation of 14 dof full vehicle model," in *International Conference on Instrumentation, Communications, Information Technology, and Biomedical Engineering*, Nov. 23-25 2009, pp. 1–6.

[5] Z. Zhang, "Parameter estimation techniques: A tutorial with application to conic fitting," *Image and Vision Computing*, vol. 15, no. 1, pp. 59–76, 1997.

[6] D. A. Belsley, E. Kuh, and R. E. Welsch, *Regression diagnostics: Identifying influential data and sources of collinearity*. John Wiley & Sons, 2005, vol. 571.

[7] M. Wu and A. W. Smyth, "Application of the unscented Kalman filter for real-time nonlinear structural system identification," *Structural Control and Health Monitoring*, vol. 14, no. 7, pp. 971–990, 2007.

[8] J. Han, Q. Song, and Y. He, *Adaptive Unscented Kalman filter and its applications in nonlinear control*. INTECH Open Access Publisher, 2009.

[9] C. G. Hilborn and D. G. Lainiotis, "Optimal estimation in the presence of unknown parameters," *IEEE Transactions on Systems Science and Cybernetics*, vol. 5, no. 1, pp. 38–43, 1969.

[10] K. Myers and B. Tapley, "Adaptive sequential estimation with unknown noise statistics," *IEEE Transactions on Automatic Control*, vol. 21, no. 4, pp. 520–523, Aug. 1976.

[11] S. Hong, T. Smith, F. Borrelli, and J. K. Hedrick, "Vehicle inertial parameter identification using extended and unscented Kalman filters," in *IEEE Conference on Intelligent Transportation Systems*, 2013, pp. 1436–1441.

[12] E. A. Wan and R. Van Der Merwe, "The unscented Kalman filter for nonlinear estimation," in *Adaptive Systems for Signal Processing, Communications, and Control Symposium*, Alberta, Canada, October 1-4, 2000, pp. 153–158.

[13] A. H. Jazwinski, *Stochastic Processes and Filtering Theory*. Courier Corporation, 2007.

[14] J. Qi, K. Sun, J. Wang, and H. Liu, "Dynamic state estimation for multi-machine power system by unscented kalman filter with enhanced numerical stability," *arXiv preprint arXiv:1509.07394*, 2015.

[15] M. Saha, B. Goswami, and R. Ghosh, "Two novel costs for determining the tuning parameters of the Kalman filter," in *Proceedings of Advances in Control and Optimization of Dynamic Systems*, Bangalore, India, Feb. 16-18 2012, pp. 1–8.