Vehicle Mass Estimation Using a Total Least-Squares Approach*

Stephan Rhode and Frank Gauterin[1]

Abstract— We introduce an incremental total least-squares vehicle mass estimation algorithm, based on a vehicle longitudinal dynamics model. Available control area network signals are used as model inputs and output. In contrast to common vehicle mass estimation schemes, where noise is only considered at the model output, our algorithm uses an errors-in-variables formulation and considers noise at the model inputs as well. A robust outliner treatment is realized as batch total least-squares routine and hence, the proposed algorithm works in a superior way on a broad range of vehicle acceleration. The results of six test runs on various vehicle masses show highly accurate mass estimation results on high and low dynamics of vehicular operation.

I. INTRODUCTION

The vehicle mass is a vital parameter to determine and forecast the driving resistance. Hence, a robust, fast and accurate vehicle mass estimation method is required for several driver assistance, driving strategy and range management systems.

Over the last decade, numerous model-based vehicle mass estimation methods were presented. The major approach is to describe the vehicle's longitudinal dynamics through a linear state-space or input-output model. Available control area network (CAN) signals are used as model inputs and output. Additional methods use suspension dynamics, lateral dynamics or drive-train dynamics. A detailed overview is given in [1] and [2].

In [1], a recursive least-squares (RLS) vehicle mass estimator is used in conjunction with a fuzzy supervisor to determine driving states where the vehicle's motion is predominantly longitudinal. A band-pass filter extracts low-frequent road grade and rolling resistance forces from higher-frequent acceleration forces. The proposed supervisor thresholds seem to be too restrictive for energy-efficient driving strategies. For instance, the vehicle acceleration should exceed one meter per square second. A simultaneous mass and road grade RLS estimation scheme is presented in [3]. For an accurate estimation, sufficiently rich vehicle dynamics were ensured during the test runs. However, this assumption may be invalid for energy-efficient driving strategies. To prevent the ambitious simultaneous estimation of time-varying road grade and constant vehicle mass, McIntyre et al. [4] developed a two-stage estimation strategy. The vehicle mass is estimated by an adaptive least-squares (LS) scheme, followed by a

nonlinear road grade estimator. Bae, Ryu, and Gerdes [5] introduce an aerodynamic drag and vehicle mass estimator using road grade measurements from a global positioning system (GPS). The parameter estimation was conducted during repeated vehicle acceleration and deceleration through coasting. Therefore, the presented velocity profile shows a sawtooth analog shape. As indicated above, the restriction to rich dynamics can cause this approach to fail in energy-efficient driving states on low dynamics.

Common model-based RLS vehicle mass estimation approaches generate low computational effort, but use a simplified LS problem formulation. The model output is treated as the sum of true data and unknown measurement noise. In contrast, all inputs are assumed to be noise-free and accurate. Moonen [6] pointed out that the simplified LS approximation results in biased estimates when this assumption is not met and noise affects the input as well.

The total least-squares (TLS) regression (Section II) considers unknown noise at the model output and at the model input as well. The proposed incremental total least-squares algorithm (Section III) is applied in a vehicle mass estimation scheme (Section IV) and is based on a vehicle longitudinal dynamics model (Section IV-A). Outliners are treated in a batch total least-squares approach (Section IV-D), thus a very weak data preselection (Section IV-C) can be used. Accurate mass estimates (Section V) were obtained in six test runs on high and low vehicle acceleration, which equates to sportive drive and comfort or energy-efficient drive, respectively.

II. TOTAL LEAST-SQUARES

Markovsky and Van Huffel [7] illustrate the basic TLS algorithm as a maximum-likelihood estimator for the parameters X in the overdetermined linear system of equations (1).

$$AX = B, \quad A = \overline{A} + \widetilde{A}, \quad B = \overline{B} + \widetilde{B}. \tag{1}$$

The input data $A \in \mathbb{R}^{m \times n}$ and output data $B \in \mathbb{R}^{m \times d}$ are given. From now on, we focus on multi-input single-output (MISO) systems, where $d = 1$. The matrices A and B are sums of noise-free data $\overline{A}$ and $\overline{B}$ and measurement noise $\widetilde{A}$ and $\widetilde{B}$, respectively. The noise is assumed to be uncorrelated and normally distributed with zero mean. Extensions of the basic TLS can deal with column-wise or row-wise correlated noise. A hierarchical overview is given in [7]. The model (1) is known as errors-in-variables (EIV) model. Apart from TLS, there are various EIV identification methods studied in [8], such as instrumental variables (IV). In contrast, simplified LS corrects the system of equations optimally when $\widetilde{A}$ is zero (noise-free input).

The solution of the basic TLS requires the singular value decomposition (SVD) in (2) of the data matrix $Z = [A \ B]$, where $Z \in \mathbb{R}^{m \times n+d}$. The matrices $U \in \mathbb{R}^{m \times m}$ and $V \in \mathbb{R}^{n+d \times n+d}$ are orthogonal and their columns are called the left and right singular vectors, respectively. The positive diagonal matrix $S \in \mathbb{R}^{m \times n+d}$ contains the singular values of Z in decreasing order.

$$Z = USV^\top, \quad U^\top ZV = S, \quad S = \text{diag}(\sigma_1, \ldots, \sigma_{n+d}). \quad (2)$$

The TLS parameter estimate solution becomes with

$$V := \begin{pmatrix} \overset{n}{V_{11}} & \overset{d}{V_{12}} \\ V_{21} & V_{22} \end{pmatrix} \begin{matrix} n \\ d \end{matrix}, \quad (3)$$

if $\sigma_n \neq \sigma_{n+1}$ and V_{22} is non-singular,

$$\widehat{X} = -V_{12}V_{22}^{-1}. \quad (4)$$

If one thinks of online applications, the major drawback of this batch algorithm is the computational effort. The size of Z, U and S increases with each sample update. Hence, the SVD computation becomes slower with time. Previous data have to be stored (buffer size limit) and re-calculated in each iteration. For this reason, the batch computation of the SVD is only useful if the data size does not increase or a size limit is ensured by a downdate scheme.

III. INCREMENTAL TOTAL LEAST-SQUARES

Online system identification requires fast algorithms with limited computations per update. This may be the main reason why RLS is the most applied algorithm in vehicle mass estimation. Additionally, RLS offers weightings, parameter bounds and forgetting factors. The latter allow to estimate slow time-varying parameters as well. Recently, Kubus, Kroger, and Wahl [9] showed an encouraging example that demonstrates the superior accuracy of recursive total least-squares (RTLS) algorithms. They compared the RTLS, RLS and recursive instrumental variables (RIV) estimation of ten parameters in a robotics application. The results attest the predominant accuracy of RTLS. The proposed RTLS algorithm is based on the work of Brand [10], who derived a fast and accurate SVD update algorithm with truncation. In [11] this algorithm is extended to an update, downdate, revise and recenter functionality.

The proposed incremental TLS algorithm herein is an extended version of that presented by Gu and Eisenstat [12]. The basic idea is to take advantage of the previous SVD when new data arrives and the data matrix Z is updated. As only V is needed in (4) to compute the parameter estimate, a full SVD update of $USV^\top$ is not required. Presuming a known data matrix Z_i, the new data matrix Z_{i+1} consists of

$$Z_{i+1} = \begin{pmatrix} Z_i \\ z^\top \end{pmatrix}, \quad (5)$$

where $Z_{i+1} \in \mathbb{R}^{m+1 \times n+d}$ and $z^\top$ is one new data row. We focus here on appending a single data row, but the algorithm

works as well when a batch of new samples arrives. We compute

$$a = V_i^\top z, \quad (6)$$

where V_i is known from the previous SVD and gain $D \in \mathbb{R}^{m-2 \times n+d}$ as submatrix from the known $S_i \in \mathbb{R}^{m \times n+d}$. Now, we implement the truncation approach of [10] to size the following SVD to an efficient rank. The QR decomposition yields

$$JK = \text{qr}(z - V_i a) \quad \text{and} \quad (7)$$

$$v = \sqrt{\det(K^\top K)}, \quad (8)$$

where v is required to decide whether a truncation is performed. Brand [10] defines v as volume of $z^\top$ which is orthogonal to V_i. Note that J is not needed further on. Now let

$$PNQ^\top = \text{svd}\begin{pmatrix} D \\ a^\top \end{pmatrix} \quad (9)$$

be a common batch SVD, then

$$Z_{i+1} = U_{i+1}\begin{pmatrix} N \\ 0 \end{pmatrix}\begin{pmatrix} V_i & Q \end{pmatrix}^\top = U_{i+1}S_{i+1}V_{i+1}^\top \quad (10)$$

is the desired SVD update. Note that P and U_{i+1} are not required to gain parameter estimates, hence the update of U_{i+1} is not derived in this paper. The interested reader is referred to [12] for a full SVD update procedure.

If $v < \mathfrak{v}$, where $\mathfrak{v}$ is near the machine precision, we downsize S_{i+1} to $S_{i+1} \in \mathbb{R}^{n+d \times n+d}$. This means, that the number of singular values in S_{i+1} remains at $n + d$ and the computational effort is notably reduced. Hence, the algorithm is suitable for online applications. Finally, the parameter estimate $\widehat{X}_{i+1}$ is obtained from (4).

A simple linear two-input one-output channel model was simulated to compare the proposed incremental TLS algorithm and the classic batch TLS procedure. Basic linear functions were used as input signals. White Gaussian noise is added on both input channels and the output. Fig. 1 shows the relative parameter estimation error over samples, where each sample corresponds to one new data row. The estimates of parameters $\widehat{X_1}$ and $\widehat{X_2}$ are approximately similar. The reduction in computing time through the incremental TLS method is exceptional. Fig. 2 shows the parameter estimates for the same model. In contrast to Fig. 1, RLS give slightly biased estimates even for this basic model.

IV. VEHICLE MASS ESTIMATION SCHEME

A. Vehicle longitudinal dynamics model

Vehicle longitudinal dynamics can be expressed through a basic force equilibrium of tractive force and total driving resistance.

$$F = m \cdot g \cdot f_r \cdot \cos\alpha + m \cdot g \cdot \sin\alpha + (m + m_{\text{rot}})a_x + c_x \frac{\rho}{2}A \cdot v^2, \quad (11)$$

where F is the tractive force, m is the vehicle mass, f_r is the coefficient of rolling resistance, α is the road-grade, a_x is the longitudinal acceleration, c_x is the drag coefficient, ρ
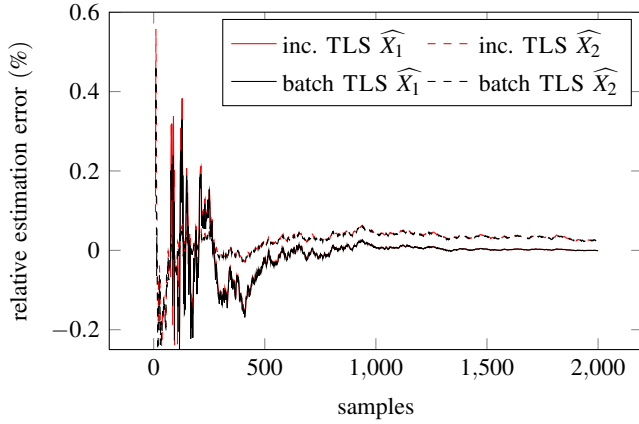
Fig. 1: Relative parameter estimate errors of the proposed incremental TLS update algorithm with $\mathfrak{v} = 10^{-13}$ in comparison with the default batch TLS. The incremental procedure took 1.75 seconds, while the batch TLS required 77 seconds on an Intel P8600 CPU.
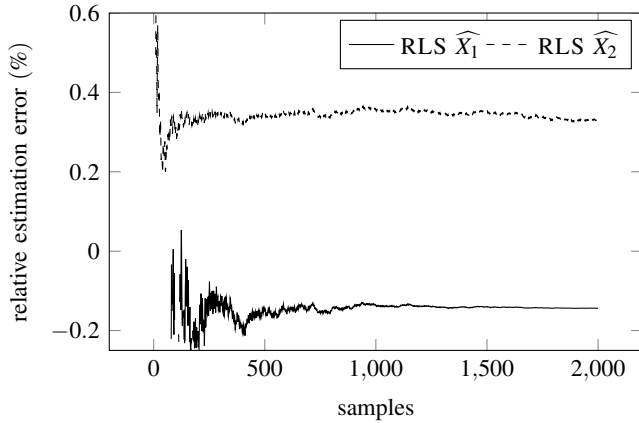

Fig. 2: RLS parameter estimates using the rarx command of MATLAB®. The computing time was 7.9 seconds on the same machine as in Fig. 1.

is the air density and v is the velocity. The vehicle cross-sectional area A and the gravity acceleration g are known absolute terms. The inertia loss $m_{rot}a_x$ through rotating parts is obtained from

$$m_{rot} = \frac{J_{red,i}}{r_{dyn}^2} \tag{12}$$

with the known reduced mass moment of inertia J_{red}, the gear i, and the dynamic rolling radius r_{dyn}.

In order to separate unknown parameters, one can rewrite (11) in a MISO system form.

$$\underbrace{B}_{output} = \underbrace{\begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_n \end{bmatrix}}_{parameters}^{\top} \underbrace{\begin{bmatrix} A_1 \\ A_2 \\ \vdots \\ A_n \end{bmatrix}}_{inputs} \tag{13}$$

Our model is based on an extended form of (11) and consid-

ers rolling resistance, climbing resistance, acceleration resistance and aerodynamic drag. The longitudinal acceleration a_x is taken from a vehicle acceleration sensor. This CAN signal comprises the required acceleration caused by longitudinal movement and the acceleration caused by gravity. Hence, the correction

$$a_x = a_{x,CAN} - g \cdot \sin\alpha \tag{14}$$

is considered according to [13]. Furthermore, F is gained from the CAN clutch torque signal and a vehicle-specific drive-train and internal loss model. The road-grade α is obtained from highly resolved map data (0.01 meters height increments) and available in real time during the test runs, using a GPS map matching method. The aerodynamic drag is assumed to increase linearly with the square of velocity v. The air density ρ is obtained from

$$\rho = \frac{p}{R_{s,dry}T} \quad \text{with} \quad R_{s,dry} = 2.871 \cdot 10^2 \, \text{J/kg K}, \tag{15}$$

where T is the measured air temperature and p is the ambient pressure, assumed with $1.013 \cdot 10^5$ pascal.

B. Low-pass filter

The CAN signals are available with 100 hertz frequency. To extract high-frequent noise that does not represent lower-frequent vehicle longitudinal dynamics, all CAN signals pass through a third-order Butterworth filter with 1 hertz pass frequency and 10 hertz stop frequency.

C. Basic driving state observer

A Boolean logic ensures that only samples are considered in the subsequent mass estimation scheme that fulfill the following driving states:

1) The vehicle is in operation, so the absolute value of velocity exceeds a threshold v_{min};
2) The absolute value of longitudinal acceleration exceeds a threshold $a_{x,min}$;
3) Wheel slip ratios do not exceed a threshold s_{max};
4) No braking state;
5) No coast-down with open clutch.

The thresholds above are chosen very low and can be interpreted as basic constraints. As subject for future investigation, the third condition can be replaced by a slip tire force function in the longitudinal model (13). After this data preselection, a downsample from 100 hertz to 10 hertz is performed to save storage.

In addition to this, cornering resistance, variable wind and high-frequent tractive force changes caused by shifting are not included in (13). One could extend the Boolean logic above and enhance the quantities to exclude these influences as well. However, extensive testing would be required to adjust each threshold in any imaginable driving state and boundary condition for each novel vehicle. Hence, we seek in an automatic approach to exclude driving states that are not treated within the vehicle longitudinal model. From now on, we call these states outliners.

D. Outliner treatment

Now, we introduce an unsophisticated approach to detect and exclude outliners. First, a certain number of samples is buffered to create an initial data matrix

$$Z = [A\ B] \quad \text{where}$$

$$A = \begin{bmatrix} A_{11} & A_{12} & \dots & A_{1n} \\ \vdots & \vdots & \vdots & \vdots \\ A_{m1} & A_{m2} & \dots & A_{mn} \end{bmatrix}, \quad (16)$$

$$B = \begin{bmatrix} B_1 \\ \vdots \\ B_m \end{bmatrix}.$$

Initial parameter estimates are computed with the basic TLS batch algorithm (2) and (4). These parameter estimates $\widehat{X}$ are used to compute the model error

$$E = B - \widehat{B} = B - \widehat{X}A. \quad (17)$$

Further on, the percentage of output explained by the estimated model is computed with

$$w = 100 \left(1 - \frac{\|E\|}{\|B - \overline{B}\|} \right), \quad (18)$$

here $\overline{B} = \frac{1}{k} \sum_{j=1}^{k} B_j$, and this is the mean of B, [14, p. 8-15]. From now on, we call w the model fit.

The function

$$g = E < \max(E)d \quad \text{AND} \quad E > \min(E)d \quad (19)$$

determines samples that are within an allowed error range, where $\{d \in \mathbb{R} \mid 0 < d < 1\}$ is a constant. Only samples remain in the buffer where $g = \text{TRUE}$, thus the buffer size decreases.

While $w < \mathfrak{w}$, where $\mathfrak{w}$ is a chosen threshold, a loop yields back to generate a new reduced data matrix (16) out of the buffer. Again, the parameter estimates (2), (4), error (17), and fit (18) are calculated. As (19) deletes samples with high error, the fit w increases continuously with each iteration. This approach assumes that the model (13) represents most driving states that occur. Through extensive testing, it is ensured that this while loop does not lead to a local optimum with unlikely parameter estimates.

Now, the data matrix Z can be seen as "clean" and the second entry of

$$H = \text{diag}[(A^\top A)^{-1}] \quad (20)$$

is observed.

If $H_2 < \mathfrak{H}_2$, where $\mathfrak{H}_2$ is a chosen threshold, Z forms the update rows $z^\top$ in (5) and the proposed incremental TLS algorithm is performed. H is regarded to be coherent with the estimated parameter variance, hence $H_2 \sim \sigma^2(\widehat{m})$. Finally, a certain amount of old data leaves the buffer to perform a sliding window.

Otherwise, if $H_2 > \mathfrak{H}_2$ accordingly more data is buffered and the algorithm goes back to the outliner treatment (16)–(19).

TABLE I: Overview of the performed test runs.

#	day	vehicle mass	dynamic mode
1			low automatic
2	a	2038 kg	high manual
3			low automatic
4			low automatic
5	b	2340 kg	high manual
6			low automatic

E. Mass estimation scheme as pseudo-code

The full vehicle mass estimation scheme is shown once again clearly with the following compact pseudo-code section:

1) Pass CAN signals through low-pass filter;
2) Buffer data that fulfill the quantities of the basic driving state observer;
3) Outliner treatment:

> While $H_2 > \mathfrak{H}_2$ {;
> Add data out of the buffer to Z (16);
> While $w < \mathfrak{w}$ {;
> Estimate parameters with (2), (4);
> Compute the model error (17) and the fit w (18);
> Delete data with high error (19) };
> Compute H };

4) Compute the vehicle mass estimate with the incremental TLS (5)–(10);
5) Delete a certain amount of old data in the buffer and go back to 1).

The costliest part is step three, which includes two nested loops and is done in comparatively slow batch mode. A subject for further investigation is a faster outliner treatment, using methods from the field of robust fit.

V. RESULTS WITH EXPERIMENTAL DATA

A. Test runs

Six test runs on a grand touring sports car were conducted with two sets of vehicle mass. The vehicle is equipped with an automatic longitudinal controller, thus the driver only operates manually the steering when the adaptive cruise control (ACC) mode is activated. Four test runs were performed with automatic cruise control on low dynamics and two runs were manually driven on higher dynamics. The track is a 22.9 kilometer long public road with rich variation in the road grade. Table I provides an overview of all test runs. Fig. 3 shows different dynamic levels on track kilometers 12–16 of measurement #1 and #2. It can be seen that the acceleration on low dynamic mode in measurement #1 rarely exceeds one meter per square second. The manually driven test run of measurement #2 shows a rich longitudinal dynamics.

A full set of CAN signals was recorded during the test runs and used in the vehicle mass estimation scheme afterwards. The online implementation of the proposed algorithm is subject for future work.
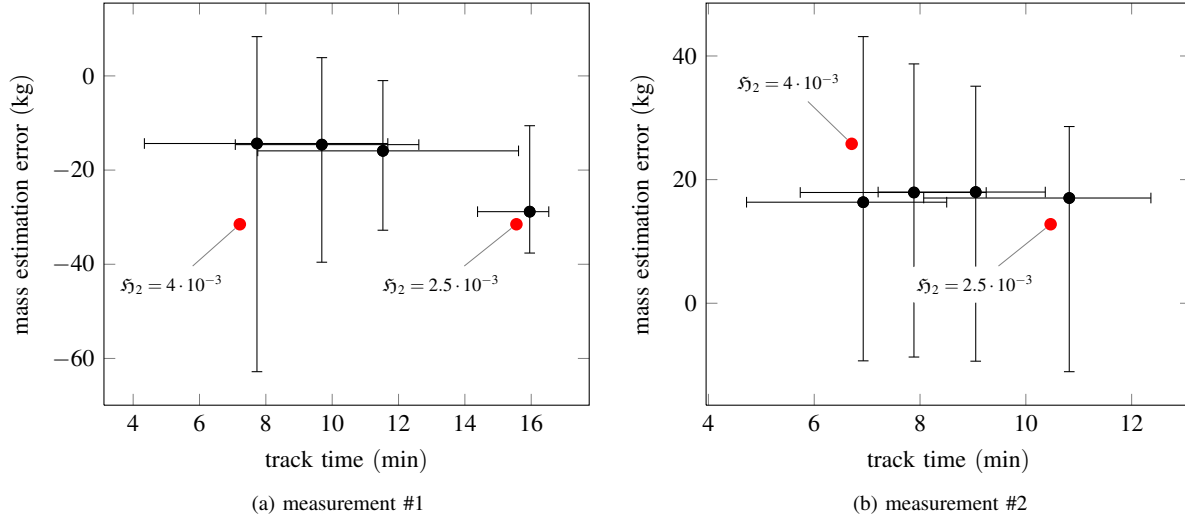
Fig. 4: Absolute error of initial mass estimates right after the outliner treatment. The measured vehicle mass is 2038 kilograms. The threshold $\mathfrak{H}_2$ is varied between $\mathfrak{H}_2 = [4 \cdot 10^{-3}, \ldots, 2.5 \cdot 10^{-3}]$, from leftmost to rightmost error bar, respectively.
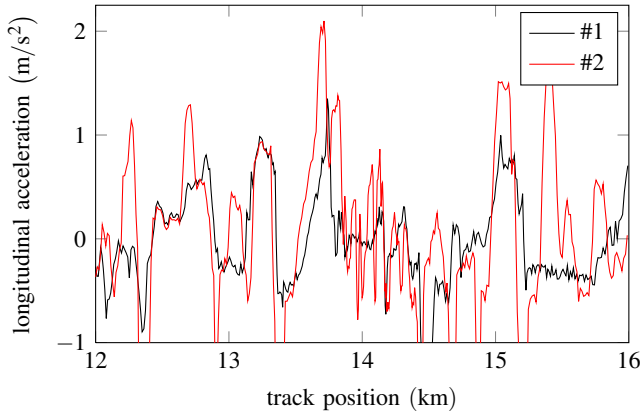

Fig. 3: Longitudinal acceleration of measurement #1 and #2. Measurement #1 is driven with automatic ACC mode on low dynamics. Measurement #2 is manually driven on higher dynamics. Large negative acceleration in measurement #2 due to braking is not shown.

B. Accuracy of initial mass estimates

The effectiveness of the proposed outliner treatment is studied in Fig. 4. For now, the algorithm is performed without incremental TLS, where the start time is varied at $0, 5, 10, \ldots, 375$ seconds in a loop. Thus, each error bar consists of 76 initial estimates and represents the mass estimation error and required time on the track. The threshold $\mathfrak{H}_2$ is decreasing from left to right. The black dots are the mean values of the mass estimation error and the required time on track, respectively.

Fig. 4(a) shows measurement #1 on low dynamics, driven in the automatic ACC mode with 2038 kilograms measured vehicle mass. The mass estimation error spread decreases with decreasing $\mathfrak{H}_2$. Apart from $\mathfrak{H}_2 = 2.5 \cdot 10^{-3}$, where the mean value of the mass estimation shows a large negative

bias, the batch TLS estimator computes estimates with slightly negative bias. However, $\mathfrak{H}_2$ and the required time on track are inversely proportionally correlated. Hence, larger values of $\mathfrak{H}_2$ reduce the needed time and yield to faster estimates.

Fig. 4(b) shows measurement #2 on high dynamics, manually driven with the same vehicle mass. Here, the mass estimation error spread decreases with decreasing $\mathfrak{H}_2$ as well. The estimator computes estimates with slightly positive bias. Once again, the inversely proportional correlation between the threshold $\mathfrak{H}_2$ and required time on track is visible. On the other hand, the time range is on a notably smaller level than in Fig. 4(a). This can be explained by the richer dynamics in measurement #2. More suitable samples can be used in the estimator on the same track distance as in Fig. 4(a).

C. Results with incremental TLS

The full proposed incremental TLS algorithm is performed every five seconds from the start of each test run. The mass estimation error over the number of SVD updates is shown in Fig. 5. The threshold $\mathfrak{H}_2$, which is used in the outliner treatment is set to $4 \cdot 10^{-4}$, while the truncation threshold $\mathfrak{v}$ is fixed to $1 \cdot 10^{-8}$. The mass estimates of measurement #1–3 in Fig. 5(a) are in a range of ± 20 kilograms after 40 SVD updates. This is equivalent to approximately ± 1 % accuracy. Due to a larger vehicle mass in Fig. 5(b), even a better accuracy of roughly $+10 - 20$ kilograms or $+0.43 - 0.85$ % is achieved from 40 SVD updates on.

The remaining parameters of (13) are not shown, since the incremental TLS algorithm herein performs updates only. From there, this scheme provides time-averaged estimates. The vehicle mass can be treated as time-constant parameter, but the coefficients of rolling resistance vary slowly in time. A downdate or forgetting factor scheme is required to extend the incremental TLS algorithm to time-varying parameters. Furthermore, a synchronous measurement of the true rolling
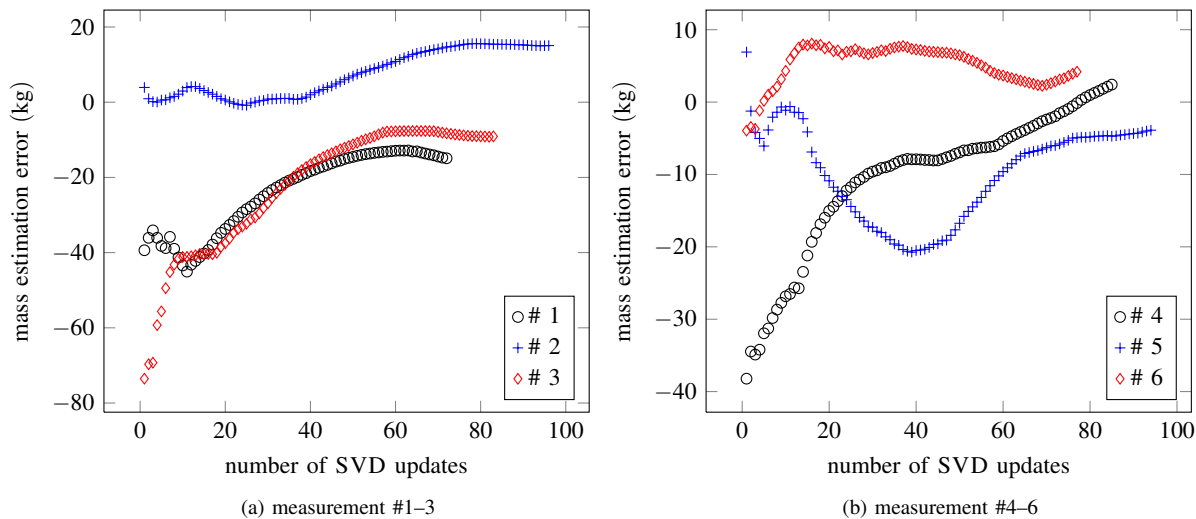
(a) measurement #1–3

(b) measurement #4–6

Fig. 5: Mass estimation error of the proposed incremental TLS algorithm. The vehicle mass is 2038 kilograms in Fig. 5(a) and 2340 kilograms in Fig. 5(b).

resistance is required to evaluate the respective parameters. In order to obtain highly accurate drag coefficient estimates, the velocity range needs to be enlarged. All test runs were performed up to at most 80 kilometers per hour.

VI. CONCLUSIONS

The presented algorithm gains vehicle mass estimates within an accuracy of ± 1 % in realistic driving conditions on low and high longitudinal vehicle dynamics. Outliners are reliably eliminated by an automatic scheme that is tunable in speed and accuracy through thresholds, but causes currently the most computational effort. By reason of this, a faster outliner treatment is required in the future. An alternative is to particularize the vehicle longitudinal model for cornering resistance and to include a gear change model. Thus, less data has to be eliminated in the outliner treatment. The incremental TLS algorithm works with superior speed via an SVD update. An analogical SVD downdate method is essential to gain estimates of parameters that vary slowly in time. In general, the proposed method is suitable for online applications, which needs to be confirmed through an applied implementation in the test vehicle.

REFERENCES

[1] H.K. Fathy, D. Kang, and J.L. Stein. "Online vehicle mass estimation using recursive least squares and supervisory data extraction." In: *American Control Conference, 2008*. 2008, pp. 1842–1848.

[2] B.L. Pence. "Recursive Parameter Estimation using Polynomial Chaos Theory Applied to Vehicle Mass Estimation for Rough Terrain." PhD thesis. University of Michigan, 2011.

[3] A. Vahidi, A. Stefanopoulou, and H. Peng. "Experiments for online estimation of heavy vehicle's mass and time-varying road grade." In: *Proceedings of the 2003 ASME Dynamic Systems and Control Division, International Mechanical Engineering Congress and Exposition*. 2003, pp. 451–458.

[4] M.L. McIntyre et al. "A Two-Stage Lyapunov-Based Estimator for Estimation of Vehicle Mass and Road Grade." In: *Vehicular Technology, IEEE Transactions on* 58.7 (Sept. 2009), pp. 3177–3185. ISSN: 0018-9545.

[5] H.S. Bae, J. Ryu, and J.C. Gerdes. "Road grade and vehicle parameter estimation for longitudinal control using GPS." In: *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*. 2001, pp. 166–171.

[6] M. Moonen. "Systolic algorithms for recursive total least squares parameter estimation and mixed RLS/RTLS problems." In: *International Journal of High Speed Electronics* 4 (1993), pp. 55–55.

[7] I. Markovsky and S. Van Huffel. "Overview of total least-squares methods." In: *Signal Processing* 87.10 (2007), pp. 2283–2302.

[8] T. Söderström. "Errors-in-variables methods in system identification." In: *Automatica* 43.6 (2007), pp. 939–958.

[9] D. Kubus, T. Kroger, and F.M. Wahl. "On-line estimation of inertial parameters using a recursive total least-squares approach." In: *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*. Sept. 2008, pp. 3845–3852.

[10] M. Brand. "Incremental Singular Value Decomposition of Uncertain Data with Missing Values." In: *Computer Vision ECCV 2002*. Ed. by Anders Heyden et al. Vol. 2350. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2002, pp. 707–720. ISBN: 978-3-540-43745-1.

[11] M. Brand. "Fast low-rank modifications of the thin singular value decomposition." In: *Linear algebra and its applications* 415.1 (2006), pp. 20–30. ISSN: 0024-3795.

[12] M. Gu and S.C. Eisenstat. "A stable and fast algorithm for updating the singular value decomposition." In: *New Haven: Yale University Department of Computer Science. RR-939* (1993).

[13] Z. Yu et al. "Vehicle Mass Estimation for Four In-Wheel-Motor Drive Vehicle." In: *Electrical Engineering and Control*. Ed. by Min Zhu. Vol. 98. Lecture Notes in Electrical Engineering. Springer Berlin Heidelberg, 2011, pp. 117–125. ISBN: 978-3-642-21765-4.

[14] L. Ljung. *System Identification Toolbox^{TM}: User's Guide*. Mathworks. 2012.