2008 IEEE/RSJ International Conference on Intelligent Robots and Systems
Acropolis Convention Center
Nice, France, Sept, 22-26, 2008

On-Line Estimation of Inertial Parameters Using a Recursive Total Least-Squares Approach

Daniel Kubus, Torsten Kröger, and Friedrich M. Wahl

***Abstract*— The estimation of the ten inertial parameters of rigid loads, which are attached to manipulators, may benefit several robotics applications, e.g.: force control, object recognition, and pose estimation. These applications require sufficiently accurate, robust, and fast estimation of the inertial parameters. Existing approaches, however, do not allow for *robust on-line* estimation, since they use standard batch least-squares techniques, which ignore noise in the data matrix. The proposed approach, however, estimates the inertial parameters on-line and very fast (approx. 1.5s), while explicitly considering noise in the data matrix by a *total* least-squares approach. Apart from estimation equations and estimation approaches, the design of estimation trajectories is addressed in this paper. The performance of the proposed estimation approach is compared with the recursive ordinary least-squares (RLS) and the recursive instrumental variables (RIV) method. Experimental results clearly recommend the proposed recursive total least-squares approach (RTLS).**

I. Introduction

Various challenging handling, assembly, and service robotics applications may benefit from the estimation of the inertial parameters (the mass, the coordinates of the center of mass, and the elements of the inertia matrix) of a manipulator load, i.e., a tool or a gripped object. Regarding force-guided and force-guarded motions, estimates of the inertial parameters of the load facilitate the elimination of all non-contact forces and torques caused by the inertia of the load, thus enabling highly-dynamic force-guided or force-guarded motions [1]–[3].
Common object recognition approaches most often rely on visual information. However, inertial parameters may also serve as recognition features. Moreover, inertial parameters can be used to estimate gripping poses of manipulated objects [4].
Since suitable CAD data is rarely provided for complex tools or gripped objects in robotics applications, the inertial parameters of the load have to be estimated experimentally. To estimate these parameters using a wrist-mounted force-torque sensor, the load is attached to the manipulator and moved on an estimation trajectory. During the motion (on-line) or afterwards (off-line), the forces and torques acting upon the sensor as well as accelerations and angular velocities of the load are fused to obtain estimates of the inertial parameters.

Aiming particularly at applications in industrial environments, the estimation approach should fulfill the following requirements: the estimates have to be provided on-line; offsets affecting the force-torque sensor values have to be compensated; sensor noise and disturbances have to be considered appropriately; workspace limitations have to be incorporated in estimation trajectory design. Since execution times and down-times are crucial in industrial assembly and handling, the estimation process should be as short as possible while maintaining sufficient quality of the estimates. Common force-torque sensors use strain gauges and hence exhibit noise, offsets, and drift effects that have to be counteracted. Apart from the measured forces and torques, the accelerations and angular velocities contained in the data matrix are subjected to considerable disturbance effects and noise, but the popular ordinary least-squares (LS) method ignores errors in the data matrix. The herein-proposed recursive total least-squares approach, however, overcomes this flaw. Moreover, in many industrial environments, the workspace of the manipulator is limited. Therefore, estimation trajectories have to be adaptable to the current environment.

The off-line estimation of inertial parameters of rigid loads attached to industrial manipulators using a wrist-mounted force-torque sensor was addressed early by An et al. [5]. Their approach employed ordinary least-squares estimation and manually-designed estimation trajectories. Kozlowski [6] extended this approach by employing trajectories that were optimized w.r.t. parameter excitation. Niebergall [7] developed a robot, which was solely dedicated to the off-line estimation of inertial parameters. Recently, Winkler [8] proposed an approach to off-line estimation using a 6D force-torque and 6D acceleration sensor. These approaches did neither consider time-efficient on-line estimation nor did they incorporate sensor offset handling. The on-line estimation approach in [4] and [3] employed a recursive instrumental variables method – effectively combining acceleration signals from different sources – and considered sensor offsets. Nevertheless, all these approaches have the same shortcoming: the employed estimation approach is based on least-squares (LS) techniques, thus *ignoring errors* in the data matrix, i.e., noise and disturbance affecting acceleration and angular velocity signals. However, our experimental results suggest not to neglect these errors, regardless of the source of the signals employed in the data matrix.

The approach proposed herein takes into account sensor offsets and enables the on-line estimation of the inertial parameters. Furthermore, it explicitly considers errors

The authors are with the Department for Robotics and Process Control, Technische Universität Braunschweig, 38106 Braunschweig, Germany. {d.kubus, t.kroeger, f.wahl}@tu-bs.de

978-1-4244-2058-2/08/$25.00 ©2008 IEEE.

in the data matrix by applying a recursive total least-squares technique. The estimation may be performed using acceleration sensor data or deriving acceleration signals from joint setpoints or actual joint angles with a simple Kalman filter. To show the performance of the proposed approach, it is compared to the recursive ordinary least-squares technique (RLS) and the recursive instrumental variables (RIV) method.

The rest of this paper is structured as follows: Section II briefly reviews the equations used for parameter estimation and addresses the problem of incorporating force-torque sensor offsets into the estimation approach. Section III summarizes the pursued trajectory design approach and notes an important pitfall of this popular approach. Section IV addresses estimation techniques. The popular recursive least-squares approach as well as the recursive instrumental variables method are reviewed and our recursive total least-squares approach is introduced. In Section V, the mentioned estimation approaches are compared experimentally and their performances are evaluated. Moreover, the influence of different acceleration signals on the estimation result is addressed. Section VI concludes the paper.

II. Estimation Equations

To estimate the inertial parameters of a load attached to a manipulator, equations relating the dynamic state of the load and its inertial parameters (the mass m, the coordinates of the center of mass $\boldsymbol{c}$, and the elements of the inertia matrix $\boldsymbol{I}$) to the forces $^S\boldsymbol{f}$ and torques $^S\boldsymbol{\tau}$ acting upon the load have to be derived from the basic laws of dynamics. Based on the Newton-Euler approach, the motion of a rigid body due to external forces and torques is described by two vector equations [4], [6].

$$^S\boldsymbol{f} = m\,^S\boldsymbol{a} - m\,^S\boldsymbol{g} + {^S\boldsymbol{\alpha}} \times m\,^S\boldsymbol{c} + {^S\boldsymbol{\omega}} \times \left(^S\boldsymbol{\omega} \times m\,^S\boldsymbol{c}\right) \quad (1)$$

$$^S\boldsymbol{\tau} = {^S\boldsymbol{I}}\,^S\boldsymbol{\alpha} + {^S\boldsymbol{\omega}} \times ({^S\boldsymbol{I}}\,^S\boldsymbol{\omega}) + m\,^S\boldsymbol{c} \times {^S\boldsymbol{a}} - m\,^S\boldsymbol{c} \times {^S\boldsymbol{g}} \quad (2)$$

where

$$\boldsymbol{I} = \begin{pmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{xy} & I_{yy} & I_{yz} \\ I_{xz} & I_{yz} & I_{zz} \end{pmatrix} \quad (3)$$

To clarify the utilized coordinate systems and parameters, Fig. 1 depicts the force-torque sensor with a load (object) O rigidly attached to it. The measured forces $^S\boldsymbol{f}$, torques $^S\boldsymbol{\tau}$, the linear and angular acceleration vector $^S\boldsymbol{a}$ and $^S\boldsymbol{\alpha}$, the angular velocity vector $^S\boldsymbol{\omega}$, the gravity vector $^S\boldsymbol{g}$, the coordinates of the center of mass $^S\boldsymbol{c}$, and the inertia matrix $^S\boldsymbol{I}$ refer to the sensor frame $\boldsymbol{S}$. If one of the mentioned parameters is not measured w.r.t. the frame $\boldsymbol{S}$, it can be transformed to this frame [9]. Note that the axes of the frame $\boldsymbol{C}$, which is located at the center of mass (COM) of the object, are parallel to the axes of $\boldsymbol{S}$.

The vector equations 1 and 2 are converted to Eq. (4) [6], which is used for parameter estimation. This equation relates the measured forces/torques $^S\boldsymbol{f}$, $^S\boldsymbol{\tau}$ and the variables

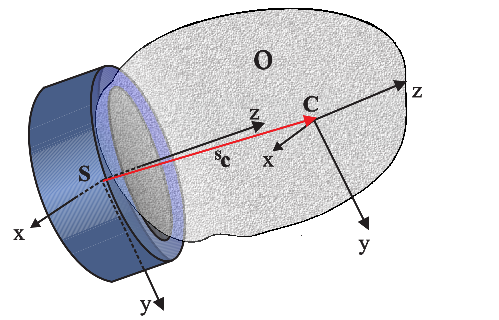

Fig. 1. Force-torque sensor with attached load; utilized coordinate systems.

$^S\boldsymbol{a}$, $^S\boldsymbol{\alpha}$, $^S\boldsymbol{\omega}$, and $^S\boldsymbol{g}$ to the complete set of inertial parameters contained in $^S\boldsymbol{\varphi}$, cf. Eq. (5).

$$\begin{pmatrix} ^S\boldsymbol{f} \\ ^S\boldsymbol{\tau} \end{pmatrix} = {^S\boldsymbol{A}}(^S\boldsymbol{a}, {^S\boldsymbol{\alpha}}, {^S\boldsymbol{\omega}}, {^S\boldsymbol{g}})\,^S\boldsymbol{\varphi} \quad (4)$$

$$\begin{aligned} ^S\boldsymbol{\varphi} \;=\; & [m, m\,^Sc_x, m\,^Sc_y, m\,^Sc_z, \\ & {^SI_{xx}}, {^SI_{xy}}, {^SI_{xz}}, {^SI_{yy}}, {^SI_{yz}}, {^SI_{zz}}]^T \end{aligned} \quad (5)$$

The data matrix $^S\boldsymbol{A}$ is given in Eq. (6) on the following page. Note that index ^{S} is omitted for brevity. The matrix $^S\boldsymbol{A}$, as defined above, does not consider an important characteristic of common strain gauge force-torque sensors. The forces and torques measured by these sensors typically show time-varying offsets that would deteriorate the estimation results. In order to handle these offsets, two approaches may be pursued: the sensor values may be zeroed in a known sensor orientation considering eliminated gravitational forces and associated torques by a pseudo gravity vector $^S\boldsymbol{g_{init}} = [^Sg_{init_x}, {^Sg_{init_y}}, {^Sg_{init_z}}]^T$; the offsets may be estimated directly. While the first approach requires zeroing each time a new object is gripped, the second approach does not entail any zeroing, but the number of estimated parameters increases from 10 to 16. The first approach leads to the matrix $^S\boldsymbol{A_{g_{init}}}$ which is composed of a matrix $^S\tilde{\boldsymbol{A}}_{\boldsymbol{g_{init}}}$ and a 6×6 zero matrix. $^S\tilde{\boldsymbol{A}}_{\boldsymbol{g_{init}}}$ considers the gravitational forces and associated torques that are eliminated by zeroing the sensor values.

$$^S\tilde{\boldsymbol{A}}_{\boldsymbol{g_{init}}} = \begin{pmatrix} ^Sg_{init_x} & 0 & 0 & 0 \\ ^Sg_{init_y} & 0 & 0 & 0 \\ ^Sg_{init_z} & 0 & 0 & 0 \\ 0 & 0 & ^Sg_{init_z} & -^Sg_{init_y} \\ 0 & -^Sg_{init_z} & 0 & ^Sg_{init_x} \\ 0 & ^Sg_{init_y} & -^Sg_{init_x} & 0 \end{pmatrix} \quad (7)$$

$$^S\boldsymbol{A_{g_{init}}} = \left[{^S\tilde{\boldsymbol{A}}_{\boldsymbol{g_{init}}}}\,\boldsymbol{0_{6\times 6}}\right] \quad (8)$$

The resulting matrix $^S\boldsymbol{A_{offs}}$ used for parameter estimation is the sum of Eq. (6) and Eq. (8).

$$^S\boldsymbol{A_{offs}} = {^S\boldsymbol{A}} + {^S\boldsymbol{A_{g_{init}}}} \quad (9)$$

Apart from sensor offsets, drift effects are an issue with common force-torque sensors. Since the estimation duration

$$
{}^{S}\boldsymbol{A} = \begin{pmatrix}
a_x - g_x & -\omega_y^2 - \omega_z^2 & \omega_x\omega_y - \alpha_z & \omega_x\omega_z + \alpha_y & 0 & 0 & 0 & 0 & 0 & 0 \\
a_y - g_y & \omega_x\omega_y + \alpha_z & -\omega_x^2 - \omega_z^2 & \omega_y\omega_z - \alpha_x & 0 & 0 & 0 & 0 & 0 & 0 \\
a_z - g_z & \omega_x\omega_z - \alpha_y & \omega_y\omega_z + \alpha_x & -\omega_y^2 - \omega_x^2 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & a_z - g_z & g_y - a_y & \alpha_x & \alpha_y - \omega_x\omega_z & \alpha_z + \omega_x\omega_y & -\omega_y\omega_z & \omega_y^2 - \omega_z^2 & \omega_y\omega_z \\
0 & g_z - a_z & 0 & a_x - g_x & \omega_x\omega_z & \alpha_x + \omega_y\omega_z & \omega_z^2 - \omega_x^2 & \alpha_y & \alpha_z - \omega_x\omega_y & -\omega_x\omega_z \\
0 & a_y - g_y & g_x - a_x & 0 & -\omega_x\omega_y & \omega_x^2 - \omega_y^2 & \alpha_x - \omega_y\omega_z & \omega_x\omega_y & \alpha_y + \omega_x\omega_z & \alpha_z
\end{pmatrix} \tag{6}
$$

is rather short (1.5s), drift effects can be neglected (at least for the employed force-torque sensor [10]).

III. Estimation Trajectories

A popular approach to the design of excitation trajectories for parameter estimation is based on sinusoidal trajectories in joint space. A fixed number of N weighted sine and cosine functions respectively are superposed to compose the trajectory of a joint [11], [12]. Following this approach, the trajectory $q_i(t)$ of joint i is given by:

$$
s_i(t) = \sum_{k=1}^{N} \rho_{i,k} \sin(2\pi k f t) + \delta_{i,k} \cos(2\pi k f t) \tag{10}
$$

$$
q_i(t) = s_i(t) + q_{i,0} \tag{11}
$$

where $s_i(t)$ denotes the sinusoidal part. The parameter $q_{i,0}$ denotes a constant offset angle; $\rho_{i,k}$ and $\delta_{i,k}$ are the coefficients of the sine and the cosine part respectively. The base frequency f of the joint trajectories is identical to ensure periodicity. Note that trajectories according to Eq. (11) are jerk-limited (since the derivative of a sinusoidal function is sinusoidal again) and hence reduce undesired excitation of the mechanical structure of the manipulator. The weights $\rho_{i,k}$ and $\delta_{i,k}$ may be optimized w.r.t. a suitable criterion that evaluates the influence of the selected trajectory on the noise and bias sensitivity of the estimates.

Regarding this influence, the correlation matrix $\boldsymbol{\Upsilon}$ consisting of M ${}^{S}\boldsymbol{A}$ matrices compiled at subsequent instants of time plays a key role.

$$
\boldsymbol{\Upsilon} = {}^{S}\boldsymbol{A}_{\Xi}^{T}\, {}^{S}\boldsymbol{A}_{\Xi} \tag{12}
$$

$$
{}^{S}\boldsymbol{A}_{\Xi} = \left[{}^{S}\boldsymbol{A}_1^{T}\, {}^{S}\boldsymbol{A}_2^{T} \ldots {}^{S}\boldsymbol{A}_M^{T}\right]^{T} \tag{13}
$$

The sensitivity of ${}^{S}\boldsymbol{\varphi}$ to errors in ${}^{S}\boldsymbol{A}_{\Xi}$ can be shown to increase with increasing condition number $\kappa(\boldsymbol{\Upsilon})$ if the noise affecting ${}^{S}\boldsymbol{A}_{\Xi}$ and the force-torque measurements remain unchanged [13], [14]. To minimize the condition number of $\boldsymbol{\Upsilon}$, following steps are performed: first, a Monte Carlo search is carried out to find promising candidate trajectories for optimization; then, the `fmincon` function provided by the *Matlab* Optimization Toolbox [15] is applied to each of the candidate trajectories to minimize their $\kappa(\boldsymbol{\Upsilon})$. At each optimization iteration, the compliance to environmental and dynamical constraints is checked [4]. In Fig. 2, the joint angles of a resulting trajectory are plotted over time.

To provide jerk limitation also in the beginning and the end of the trajectory, the sinusoidal parts preceding and succeeding the active part of the trajectory are multiplied with suitable $6th$ order polynomials $f(t)$ and $g(t)$ respectively

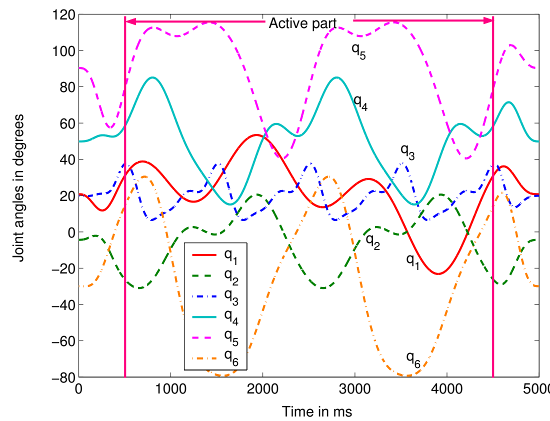

Fig. 2. Joint angles of an optimized trajectory. The highest frequency is limited to $2Hz$ to reduce excitation of the mechanical structure.

[16] yielding trajectory $q_i^*(t)$. t_1 and t_2 denote the begin and the end of the active part respectively.

$$
q_i^*(t) = \begin{cases}
q_i^*(t) = f(t)s_i(t) + q_{i,0} & for \quad t_0 < t < t_1 \\
q_i^*(t) = q_i(t) & for \quad t_1 < t < t_2 \\
q_i^*(t) = g(t)s_i(t) + q_{i,0} & for \quad t_2 < t < t_3
\end{cases} \tag{14}
$$

For instance, following constraints are applied to the initial part from t_0 to t_1:

$$
\begin{array}{ll}
f(t_0)s_i(t_0) = 0 & \frac{d}{dt}\left(f(t_0)s_i(t_0)\right) = 0 \\
\frac{d^2}{dt}\left(f(t_0)s_i(t_0)\right) = 0 & f(t_1)s_i(t_1) = s_i(t_1) \\
\frac{d}{dt}\left(f(t_1)s_i(t_1)\right) = \frac{d}{dt}\left(s_i(t_1)\right) & \frac{d^2}{dt}\left(f(t_1)s_i(t_1)\right) = \frac{d^2}{dt}\left(s_i(t_1)\right)
\end{array} \tag{15}
$$

These constraints guarantee zero acceleration at t_0 as well as steady acceleration at the transition to the active part at t_1 and hence a bounded jerk profile. Due to the structure of $q^*(t)$, these constraints can easily be converted into corresponding constraints on $f(t)$ that allow for the calculation of its coefficients. Similarly, the coefficients of $g(t)$ are obtained. Thus, jerk limitation is achieved during the entire estimation trajectory.

The condition number of a trajectory is commonly calculated employing joint accelerations and velocities based on joint angle setpoints [11], which are not subjected to noise or disturbances. However, due to material oscillations, backlash, and controller imperfections, the *actual* trajectory of the end-effector deviates from the *desired* trajectory and hence the *measured* acceleration and angular velocity values as well. Thus, a trajectory with a low condition number based on joint angle setpoints may exhibit a higher condition number if joint angle measurements are used for computation. Moreover, the same trajectory may exhibit different condition numbers with slightly different setups. Table I presents condition numbers of two resulting trajectories tr_1 and tr_2 based on joint angle setpoints and acceleration sensor data respectively. The condition numbers based on sensor data are clearly higher than those based on joint angle setpoints and additionally the trajectory with lower κ_{set} exhibits higher κ_{sens}. Therefore, the decision on an estimation trajectory should be based on condition numbers that have been obtained experimentally rather than those obtained from joint angle setpoints. Compared with the dependency of the condition number on the signal source, the dependency on the load, o_1 or o_2 respectively (cf. Fig. 5), may be neglected.

TABLE I

Condition numbers based on joint setpoints κ_{set} and sensor data κ_{sens}. As can be seen, the differences between κ_{set} and κ_{sens} are considerable, whereas the dependency of κ on the employed load is negligible.

Trajectory	Load	κ_{set}	κ_{sens}
tr_1	o_1	7.72	23.23
tr_1	o_2	7.72	23.42
tr_2	o_1	8.18	16.89
tr_2	o_2	8.18	14.44

IV. Estimation Approaches

This Section first presents considerations concerning the signals employed to compose the data matrix. Then, the popular recursive least-squares method and the recursive instrumental variables method are briefly reviewed. Finally, our recursive total least-squares method is introduced.

A. The Data Matrix and its Signal Sources

The linear and angular accelerations as well as the angular velocities contained in the data matrix ${}^S\boldsymbol{A}$ may be measured by dedicated sensors, derived from actual joint angle values [4] or from joint angle setpoints [6]. Depending on the employed controller, joint angle setpoints may significantly deviate from the actual values. Regarding common manipulators, material oscillations, backlash, etc. additionally lead to significant deviations between acceleration and velocity values based on the joint angle setpoints and the actual values. As mentioned above, acceleration sensor data, which may exhibit unfavorable noise and disturbance characteristics [10], may be used for parameter estimation as well. To demonstrate the differences between these signals, Fig. 3 shows an angular acceleration signal derived from the joint angle setpoints that are fed into the controller and an angular acceleration signal measured by our JR3 acceleration sensor. Neither the measured acceleration signals, which are affected by noise and other disturbances, nor the signals derived from joint angle setpoints, which are not influenced by material oscillations, backlash and other effects, are identical with the actual accelerations. Incorporating these signals in data matrices inevitably raises the question, whether ordinary

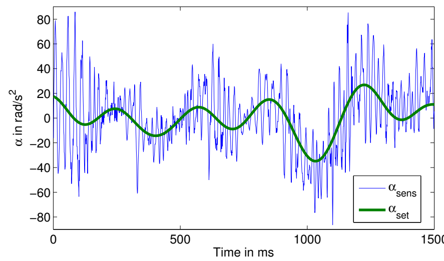

Fig. 3. Angular acceleration derived from joint angle setpoints α_{set} and measured angular acceleration α_{sens}. The differences are striking.

least-squares methods are appropriate for this estimation problem, since their error-model assumes error-free data matrices.

B. Recursive Least-Squares Method

This Subsection briefly reviews the recursive least-squares (RLS) method and its error model. The RLS method is based on the assumption that the data matrix ${}^S\boldsymbol{A}$ is free of errors; errors are merely modelled as an additive vector $\boldsymbol{e}$. Thus, the error model of the RLS method is described by the following equation:

$$\begin{pmatrix} {}^S\boldsymbol{f} \\ {}^S\boldsymbol{\tau} \end{pmatrix}_{\Xi} + \boldsymbol{e} = {}^S\boldsymbol{A}_{\Xi} {}^S\boldsymbol{\varphi} \qquad (16)$$

Ξ denotes a stack of matrices, as in Eq. (13), or a stack of vectors respectively. This error model does not consider errors in the data matrix ${}^S\boldsymbol{A}({}^S\boldsymbol{a}, {}^S\boldsymbol{\alpha}, {}^S\boldsymbol{\omega}, {}^S\boldsymbol{g})$. Regarding the parameter estimation problem at hand, however, ignoring errors in the data matrix means neglecting noise and disturbances in accelerations, angular velocities, and the gravity vector. Particularly acceleration signals may be heavily affected by noise and disturbances while force-torque signals may show clearly less noise [10]. Thus, the LS error model may be regarded as inadequate for the estimation of inertial parameters.

The recursive least-squares algorithm is given by the following equations [17]:

$$\boldsymbol{K_k} = \boldsymbol{\Sigma_{k-1}} {}^S\boldsymbol{A}_k^T \left[{}^S\boldsymbol{A_k} \boldsymbol{\Sigma_{k-1}} {}^S\boldsymbol{A}_k^T + \boldsymbol{\Lambda} \right]^{-1} \qquad (17)$$

$$\boldsymbol{\Sigma_k} = \left[\boldsymbol{1} - \boldsymbol{K_k} {}^S\boldsymbol{A_k} \right] \boldsymbol{\Sigma_{k-1}} \qquad (18)$$

$${}^S\hat{\boldsymbol{\varphi}}_k = {}^S\hat{\boldsymbol{\varphi}}_{k-1} + \boldsymbol{K_k} \left[\begin{pmatrix} {}^S\boldsymbol{f_k} \\ {}^S\boldsymbol{\tau_k} \end{pmatrix} - {}^S\boldsymbol{A_k} {}^S\hat{\boldsymbol{\varphi}}_{k-1} \right] \qquad (19)$$

$\boldsymbol{K_k}$ denotes the gain matrix at time-step k; $\boldsymbol{\Sigma_k}$ is the parameter covariance matrix; $\boldsymbol{1}$ denotes an identity matrix of appropriate size. The estimated parameter vector is denoted by ${}^S\hat{\boldsymbol{\varphi}}_k$. The matrix $\boldsymbol{\Lambda}$ contains the measurement noise covariances.

C. Recursive Instrumental Variables Method

The equations of the recursive instrumental variables method do not deviate remarkably from those of the RLS method, but in contrast to the RLS method, the RIV method can yield unbiased estimates in the presence of correlated noise.

$$\boldsymbol{K_k} = \boldsymbol{\Sigma_{k-1}} {}^S\boldsymbol{C}_{\boldsymbol{k}}^{\boldsymbol{T}} \left[{}^S\boldsymbol{B_k} \boldsymbol{\Sigma_{k-1}} {}^S\boldsymbol{C}_{\boldsymbol{k}}^{\boldsymbol{T}} + \boldsymbol{\Lambda} \right]^{-1} \tag{20}$$

$$\boldsymbol{\Sigma_k} = \left[\boldsymbol{1} - \boldsymbol{K_k} {}^S\boldsymbol{B_k} \right] \boldsymbol{\Sigma_{k-1}} \tag{21}$$

$${}^S\hat{\boldsymbol{\varphi}}_{\boldsymbol{k}} = {}^S\hat{\boldsymbol{\varphi}}_{\boldsymbol{k-1}} + \boldsymbol{K_k} \left[\begin{pmatrix} {}^S\boldsymbol{f_k} \\ {}^S\boldsymbol{\tau_k} \end{pmatrix} - {}^S\boldsymbol{B_k} {}^S\hat{\boldsymbol{\varphi}}_{\boldsymbol{k-1}} \right] \tag{22}$$

The variables in eqs. 20 – 22 are similar to those in eqs. 17 – 19 except for the matrices ${}^S\boldsymbol{C_k}$ and ${}^S\boldsymbol{B_k}$. Both ${}^S\boldsymbol{B_k}$ and ${}^S\boldsymbol{C_k}$ correspond to ${}^S\boldsymbol{A_k}$ but the signal sources of their elements may be different. The data matrix ${}^S\boldsymbol{B_k}$ contains data variables ${}^S\boldsymbol{\omega_d}$, ${}^S\boldsymbol{\alpha_d}$, ${}^S\boldsymbol{a_d}$ and ${}^S\boldsymbol{g_d}$, whereas the instrumental variables matrix $\boldsymbol{C_k}$ contains instrumental variables denoted by ${}^S\boldsymbol{\omega_{inst}}$, ${}^S\boldsymbol{\alpha_{inst}}$, ${}^S\boldsymbol{a_{inst}}$ and ${}^S\boldsymbol{g_{inst}}$ [3]. If the instrumental variables are selected appropriately, i.e., basically showing a high correlation with the data variables but a low correlation with the noise, the RIV method can yield unbiased estimates in the presence of correlated noise [17]. However, the RIV method ignores errors in the data matrix as well.

D. Total Least-Squares Approach

The fundamental drawback of ordinary least-squares methods in inertial parameters estimation is their error model as stated above. Both LS and RLS assume the data matrix ${}^S\boldsymbol{A}$ to be free of errors and constrain errors to the force-torque vector, although this assumption is clearly not valid here, since all variables contained in the data matrix are subjected to noise and disturbances. Especially the acceleration signals $\boldsymbol{a}$ and $\boldsymbol{\alpha}$ prove to be affected by considerable noise and disturbances [10]. One may presume that these errors may be neglected compared with those of the force-torque values. Experimental results presented in Section V show that this supposition does not hold (regarding our setup). Therefore, an approach explicitly considering errors in all involved variables, i.e., accelerations, angular velocities, and forces-torques will achieve better performance than traditional LS methods. The total least-squares method (TLS) explicitly considers errors in both the force-torque vector and the data matrix, but its application to on-line parameter estimation seems cumbersome due to the required update of the singular value decomposition (SVD) at each estimation cycle. An efficient incremental SVD [18], however, enables the on-line application of the total least-squares technique.

In contrast to RLS and RIV, the recursive total least-squares method is based on the more appropriate error model

$$\begin{pmatrix} {}^S\boldsymbol{f} \\ {}^S\boldsymbol{\tau} \end{pmatrix}_{\Xi} + \boldsymbol{e} = ({}^S\boldsymbol{A}_{\Xi} + \boldsymbol{E}){}^S\boldsymbol{\varphi} \tag{23}$$

that considers errors $\boldsymbol{E}$ in the data matrix and errors $\boldsymbol{e}$ in the force-torque vector. Thus, noise and disturbance are not only considered in the force-torque signals (cf. Eq. 16) but also in the acceleration and angular velocity signals. Fig. 4 illustrates the difference between the least-squares (LS) approach and the total least-squares (TLS) approach regarding a two-dimensional line-fitting problem. In Fig. 4a), the error measure employed by the ordinary least-squares approach is depicted; Fig. 4b) shows the error measure of the total least-squares approach. The LS approach assumes the variables on the x-axis to be free of errors and hence measures the fitting error as the sum of the squared *vertical* distances between the data points and the estimated line. In contrast, the TLS approach considers the sum of the squared *orthogonal* distances between the data points and the line thus incorporating errors in both dimensions in the error measure. Regarding the estimation problem at hand, employing the orthogonal distances in the error measure means considering noise and disturbance in forces-torques *as well as* accelerations and angular velocities.

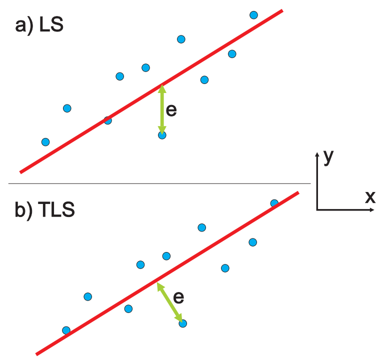

Fig. 4. **Line fitting with LS and TLS. In contrast to the LS approach, the TLS approach considers errors in the x-direction as well.**

The basic steps of the proposed recursive total least-squares approach may be summarized as follows:

1) Calculate an initial SVD with a standard SVD algorithm using a small number of data matrices (here: 2 – 3).
2) Compile a new input matrix consisting of the current data matrix ${}^S\boldsymbol{A}$ and force-torque vector $[{}^S\boldsymbol{f}, {}^S\boldsymbol{\tau}]^T$ and perform an SVD update incorporating the new data.
3) Update estimate: If the deviation between the smallest singular values is less than ε, transform left singular vectors; compute TLS solution from the left singular vectors.
4) Continue with 2 or stop estimation.

The crucial issue regarding on-line estimation using TLS approaches are the high costs of the SVD or SVD updates respectively. Over the last decades, several algorithms for efficiently updating the SVD or approximations have been proposed, e.g.: the so-called ULV decomposition may be used to yield an approximate TLS solution [19], [20]; Gu and Eisenstat [21] proposed a stable and very efficient SVD update algorithm. The update algorithm proposed by Brand [18], however, provides an exact and stable update of the SVD with maintainable computational costs, especially if an update of the right singular vectors is not required and the number of rows is small. Furthermore, it permits adding an arbitrary number of columns in one update-step, e.g., an entire data matrix including current force-torque vector. Compared to the computational costs of the SVD update, those of the remaining steps to obtain the TLS estimate – a potential Householder transform and the final computation of the estimates – are negligible.

The proposed recursive total least-squares algorithm combines the well-known TLS algorithm [22] with an efficient incremental SVD update [18]. To decrease the computational complexity of the approach, the right singular vectors are not updated since they are not necessary for computing the TLS solution.

In the first step, an initial SVD of at least $j = 2$ data matrices and force-torque vectors respectively is computed.

$$\left[\left[\boldsymbol{D}\left[{}^S\boldsymbol{A}_1\begin{pmatrix} {}^S\boldsymbol{f}_1 \\ {}^S\boldsymbol{\tau}_1 \end{pmatrix}\right]\boldsymbol{T}\right]^T, \ldots, \left[\boldsymbol{D}\left[{}^S\boldsymbol{A}_j\begin{pmatrix} {}^S\boldsymbol{f}_j \\ {}^S\boldsymbol{\tau}_j \end{pmatrix}\right]\boldsymbol{T}\right]^T\right] \quad (24)$$
$$= \boldsymbol{M_j} \overset{\text{SVD}}{=} \boldsymbol{U_j S_j V_j^T}$$

Let $\boldsymbol{M_k} \overset{\text{SVD}}{=} \boldsymbol{U_k S_k V_k^T}$ denote the SVD of the initial data or the SVD of the last step respectively. The incremental SVD update exploits the relation:

$$[\boldsymbol{M_k N_k}] \overset{\text{SVD}}{=} [\boldsymbol{U_k} \ \ \boldsymbol{J_k}] \begin{bmatrix} \boldsymbol{S_k} & \boldsymbol{L_k} \\ \boldsymbol{0} & \boldsymbol{G_k} \end{bmatrix} \begin{bmatrix} \boldsymbol{V_k} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{1} \end{bmatrix}^T \quad (25)$$
$$= \left[\boldsymbol{U_{k+1} S_{k+1} V_{k+1}^T}\right] \overset{\text{SVD}}{=} \boldsymbol{M_{k+1}}$$

where

- $\boldsymbol{1}$ denotes the identity matrix
- $\boldsymbol{N_k}$ denotes the transpose of the new data matrix and current forces-torque vector $\left[{}^S\boldsymbol{A}_k\begin{pmatrix} {}^S\boldsymbol{f}_k \\ {}^S\boldsymbol{\tau}_k \end{pmatrix}\right]$ multiplied with the diagonal weighting matrices $\boldsymbol{D}$ and $\boldsymbol{T}$.

$$\boldsymbol{N_k} = \left[\boldsymbol{D}\left[{}^S\boldsymbol{A}_k\begin{pmatrix} {}^S\boldsymbol{f}_k \\ {}^S\boldsymbol{\tau}_k \end{pmatrix}\right]\boldsymbol{T}\right]^T \quad (26)$$

- $\boldsymbol{L_k}$ is the projection of $\boldsymbol{N_k}$ onto the orthogonal basis formed by the left singular vectors.

$$\boldsymbol{L_k} = \boldsymbol{U_k^T N_k} \quad (27)$$

- $\boldsymbol{G_k}$ is the projection of $\boldsymbol{N_k}$ on the subspace orthogonal to $\boldsymbol{U_k}$ obtained by QR decomposition.

$$\boldsymbol{J_k G_k} \overset{\text{QR}}{=} \boldsymbol{N_k} - \boldsymbol{U_k L_k} \quad (28)$$

In order to update the SVD, the middle matrix $\boldsymbol{Z_k}$ in Eq. (25) has to be diagonalized.

$$\boldsymbol{Z_k} = \begin{bmatrix} \boldsymbol{S_k} & \boldsymbol{L_k} \\ \boldsymbol{0} & \boldsymbol{G_k} \end{bmatrix} \quad (29)$$

Let

$$\boldsymbol{U'_k S'_k V'^T_k} \overset{\text{SVD}}{=} \boldsymbol{Z_k} \quad (30)$$

denote the SVD of the middle matrix. The SVD of this comparatively small matrix may be calculated using a standard SVD approach, e.g. [23]. Before the diagonalization of the middle matrix $\boldsymbol{Z_k}$ is performed, the volume v of $\boldsymbol{N_k}$ orthogonal to $\boldsymbol{U_k}$ is calculated.

$$v = \sqrt{det(\boldsymbol{G_k^T G_k})} \quad (31)$$

If this volume is very small, i.e., $v < \varepsilon$ with ε in the range of machine precision, $\boldsymbol{Z_k}$ is reduced to $\boldsymbol{Z_k} = [\boldsymbol{S_k} \ \ \boldsymbol{L_k}]$ and the updated left singular vectors in matrix $\boldsymbol{U_{k+1}}$ and singular value matrix $\boldsymbol{S_{k+1}}$ are given by

$$\boldsymbol{U_{k+1}} = \boldsymbol{U_k U'_{k;1:r,1:r}} \quad (32)$$
$$\boldsymbol{S_{k+1}} = \boldsymbol{S'_{k;1:r,1:r}} \quad (33)$$

where r denotes the rank and $\boldsymbol{U'_{k;1:r,1:r}}$ denotes a submatrix of $\boldsymbol{U'}$ at time-step k containing its first r rows and r columns. Otherwise, the SVD update procedure is performed according to

$$\boldsymbol{U_{k+1}} = [\boldsymbol{U_k} \ \boldsymbol{J_k}]\,\boldsymbol{U'_k} \quad (34)$$
$$\boldsymbol{S_{k+1}} = \boldsymbol{S'_k} \quad (35)$$

Note that both an explicit calculation of $\boldsymbol{V'_k}$ and an update of $\boldsymbol{V}$ are not necessary since $\boldsymbol{V}$ is not involved in the TLS solution. To prevent $\boldsymbol{U}$ from loosing othogonality, Brand [18] proposed a re-orthogonalization method. However, regarding the conducted experiments, re-orthogonalization has proven unnecessary. After the update of the SVD, the matrix $\boldsymbol{S_{k+1}} = diag(s_{k+1;1}, \ldots, s_{k+1;n+1})$ is regarded. If the smallest singular value $s_{k+1;n+1}$ is not unique, i.e. $s_{k+1;n+1-p} = s_{k+1;n+1-(p-1)} = \ldots = s_{k+1;n+1}$ (using an ε environment), $\boldsymbol{U_{k+1}}$ is transformed with a Householder transform $\boldsymbol{H}$ [22], i.e., $\boldsymbol{W_{k+1}} = \boldsymbol{U_{k+1} H}$, so that $\boldsymbol{W_{k+1;n+1,n-p+1\ldots n}} = \boldsymbol{0}$ (p elements of the last row); otherwise $\boldsymbol{W_{k+1}} = \boldsymbol{U_{k+1}}$. Please consult [22] for numerical and implementation issues. The last step merely consists of the calculation of the inertial parameter estimates based on $\boldsymbol{W_{k+1}}$:

$$\hat{\varphi}_{k+1;i} = \frac{-t_{i,i} w_{k+1;i,n+1}}{t_{n+1,n+1} w_{k+1;n+1,n+1}} \quad (36)$$

where $t_{i,i}$ denotes an element of the weighting matrix $\boldsymbol{T}$. Note that a solution exists only if $w_{k+1;n+1,n+1} \neq 0$.

V. Experimental Results

A. Description of the Experimental Setup

The presented experiments have been conducted with a Stäubli RX60 industrial manipulator equipped with a wrist-mounted JR3 force-torque and acceleration sensor (85M35A3-I40-D 200N12). Instead of the standard Stäubli controller, we use an open control architecture, as presented in [24]. The parameter estimation has been performed with a frequency of $1000Hz$ and the estimation duration has been limited to $1.5s$.

Fig. 5 depicts three rigid test-loads of masses ranging from $1kg$ to $1.6kg$, whose inertial parameters have been estimated in the experiments. The test-loads have shapes that simplify theoretical calculation of their inertial parameters to establish ground truth data. For instance, the theoretically determined parameter vector of o_1 is:

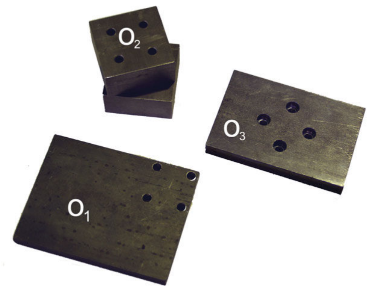

Fig. 5. **Employed test-loads made of steel. o_2 consists of two identical blocks that may be stacked or used separately.**

$$
\begin{aligned}
\hat{\boldsymbol{\varphi}}_{\boldsymbol{th}} &= [1.00kg, -43.5kgmm, 28.5kgmm, 22.5kgmm,\\
&\quad 2.21 \cdot 10^{-3}kgm^2, 1.22 \cdot 10^{-3}kgm^2, 0.96 \cdot 10^{-3}kgm^2,\\
&\quad 3.74 \cdot 10^{-3}kgm^2, -0.63 \cdot 10^{-3}kgm^2, 4.95 \cdot 10^{-3}kgm^2]^T
\end{aligned}
$$

As a performance measure, the relative estimation error $e_{rel}(\hat{x})$ has been used.

$$e_{rel}(\hat{x}) = \left| \frac{\hat{x} - x_{th}}{x_{th}} \right| \cdot 100\% \qquad (37)$$

Since the distal part of the force-torque sensor which is attached to the load is not massless, it contributes to the estimated load mass. This effect has been eliminated by estimating the inertial parameters of the distal sensor part according to [4] and subtracting them from the estimation result.

B. Performance Results of Estimation Approaches

Regarding the performance of the different estimation approaches, significant differences can be observed. Fig. 6 exemplifies the relative estimation errors of the estimation approaches. The acceleration signals have been provided by a JR3 acceleration sensor. The angular velocity signals have been obtained by the Kalman filtering approach described in [4]. Unless otherwise noted, accelerations and angular velocities based on joint angle setpoints serve as instrumental variables for the RIV method. The RLS approach shows significant errors especially affecting the elements of the inertia matrix. Employing the RIV method significantly decreases the estimation error. However, even better estimation results have been achieved by the RTLS approach. As can be seen, the RTLS estimates outperform the RIV estimates in all but one parameters.
If the angular acceleration signals are derived from the joint angle measurements as described in [4] instead of measuring them directly, the estimation error is reduced considerably. Fig. 7 shows the corresponding relative estimation errors of the different estimation approaches. In this case, the RTLS approach outperforms RIV and RLS even more clearly than in the previous diagram. The superior performance of the RTLS approach can be observed regardless of the employed load, trajectory, and acceleration signal source. Regarding the convergence of the estimates, minor differences can be observed.

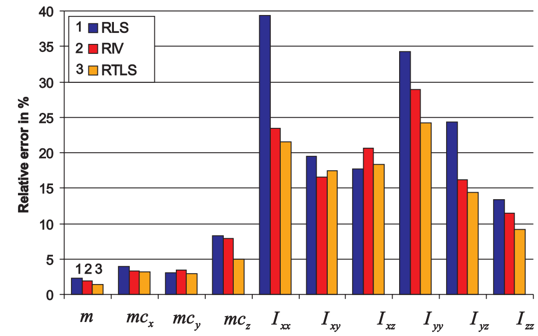

Fig. 6. **Relative errors of the different estimation techniques using object o_1 and trajectory tr_2. Acceleration measurements are provided by a JR3 acceleration sensor. The total least-squares approach shows the best overall performance.**

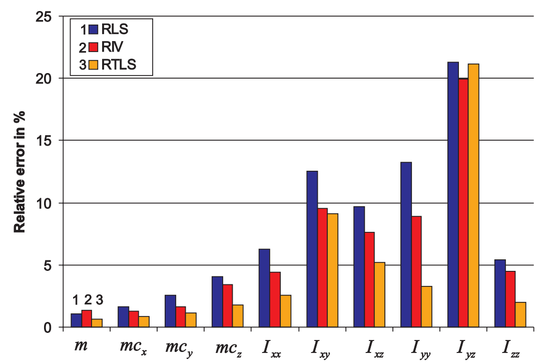

Fig. 7. **Relative errors of the different estimation techniques using object o_1 and trajectory tr_2. Linear acceleration signals are provided by the JR3 acceleration sensor but angular accelerations are derived from the joint angle measurements according to [4]. The total least-squares approach again shows the best overall performance.**

Fig. 8 exemplarily depicts the evolution of the RTLS and RIV estimates of mc_z over time using o_2 and tr_2. The RTLS estimate approaches the true value not as quickly as the RLS estimate in the beginning, which is also observed for other parameters, loads, and trajectories. Nevertheless, the convergence behaviors of the estimation approaches justify the limitation of the estimation duration to $1.5s$.

C. Performance Results for Different Signals

Fig. 9 presents relative estimation errors achieved with the RTLS approach using different acceleration signals. *KF* denotes acceleration signals derived from encoder measurements using the Kalman filtering approach mentioned above. *Mixed* denotes a combination of the linear accelerations provided by the JR3 acceleration sensor and angular accelerations based on the filtering approach. Concerning our experimental setup, combining angular acceleration signals

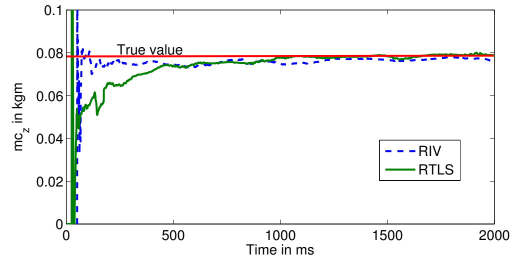

Fig. 8. **Evolution of the estimate of mc_z over time using o_2 and tr_2.**

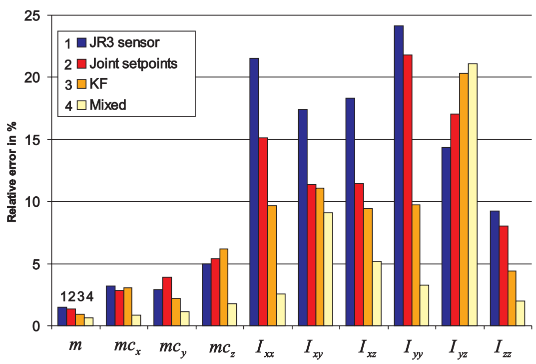

Fig. 9. **Relative RTLS estimation errors using o_1 and tr_2, but different acceleration and angular velocity signals.**

based on the filtering approach with linear acceleration signals provided by the JR3 sensor yields the best results.

VI. Conclusion

An on-line *total* least-squares approach to inertial parameters estimation of robot loads has been presented. The proposed estimation approach considers errors in both the data matrix which contains accelerations and angular velocities and the measurement vector that contains forces and torques. In contrast to the proposed recursive total least-squares (RTLS) approach, ordinary least-squares approaches merely consider errors in the measurement vector and assume the elements of the data matrix to be error-free thus ignoring the effects of noise that affects the acceleration measurements. Experiments comparing the performance of two popular parameter estimation methods, i.e., the recursive least-squares method (RLS), and the recursive instrumental variables method (RIV), with the performance of the proposed recursive *total* least-squares method (RTLS) clearly recommend the latter approach, that proves superior regardless of the employed load and trajectory.

Acknowledgments

We would like to thank *QNX Software Systems* for providing free software licenses as well as Harald Meyer-Kirk and Lars Regner for manufacturing mechanical accessories for the experimental setup.

References

[1] J. Gámez García, A. Robertsson, J. Gómez Ortega, and R. Johansson. Generalized contact force estimator for a robot manipulator. In *Proc. of IEEE International Conference on Robotics and Automation*, pages 4019–4024, 2006.

[2] T. Kröger, D. Kubus, and F. M. Wahl. 6d force and acceleration sensor fusion for compliant manipulation control. In *Proc. of IEEE/RSJ International Conference on Intellegent Robots and Systems*, pages 2626–2631, 2006.

[3] D. Kubus, T. Kröger, and F. M. Wahl. Improving force control performance by computational elimination of non-contact forces/torques. In *Proc. of IEEE International Conference on Robotics and Automation*, 2008.

[4] D. Kubus, T. Kröger, and F. M. Wahl. On-line rigid object recognition and pose estimation based on inertial parameters. In *Proc. of IEEE/RSJ International Conference on Intellegent Robots and Systems*, 2007.

[5] C.H. An, C. G. Atkeson, and J. M. Hollerbach. *Model-based Control of a Robot Manipulator*. The MIT Press series in artificial intelligence. The MIT Press, 1988.

[6] K. Kozlowski. *Modelling and Identification in Robotics*. Advances in Industrial Control. Springer, 1998.

[7] M. Niebergall and H. Hahn. Development of a measurement robot for identifying all inertia parameters of a rigid body in a single experiment. *IEEE Transactions on Control Systems Technology*, 9(2):416–423, March 2001.

[8] A. Winkler and J. Suchy. Dynamic force/torque measurement using a 12dof sensor. In *Proc. of IEEE/RSJ International Conference on Intellegent Robots and Systems*, pages 1870–1875, 2007.

[9] P. J. McKerrow. *Introduction to Robotics*. Addison-Wesley Publishers Ltd., 1991.

[10] T. Kröger, D. Kubus, and F. M. Wahl. 12d force and acceleration sensing: A helpful experience report on sensor characteristics. In *Proc. of IEEE International Conference on Robotics and Automation*, 2008.

[11] J. Swevers, C. Ganseman, D. B. Tükel, J. de Schulter, and H. Van Brussel. Optimal robot excitation and identification. *IEEE Trans. on Robotics and Automation*, 13(5):730–739, 1997.

[12] J. Swevers, W. Verdonck, B. Naumer, S. Pieters, and E. Biber. An experimental robot load identification method for industrial application. *The International Journal of Robotics Research*, 21(8):701–712, 2002.

[13] M. Gautier and W. Khalil. Exciting trajectories for the identifcation of base inertial parameters of robots. *The International Journal of Robotics Research*, 11(4):362–375, 1992.

[14] D. J. Pierce and R. J. Plemmons. Tracking the condition number for RLS in signal processing. *Mathematics of Control, Signals, and Systems*, 5(1):23–39, 1992.

[15] The MathWorks Inc. The mathworks homepage. http://www.mathworks.com/products/optimization/functionlist.html [date: 02/15/2008]. Internet, 2008.

[16] D. Kubus, T. Kröger, and F. M. Wahl. Estimating inertial load parameters using force/torque and acceleration sensor fusion. In *VDI-Bericht Nr. 2012*, 2008.

[17] L. Ljung. *System Identification: Theory for the User*. Prentice Hall, 1999.

[18] M. Brand. Incremental singular value decomposition of uncertain data with missing values. In *Proceedings of the ECCV 2002, Lecture Notes on Computer Science 2350*, 2002.

[19] D. L. Boley and K. T. Sutherland. Recursive total least squares: An alternative to the discrete kalman filter. Technical Report TR 93-32, University of Minnesota, 1993.

[20] J. L. Barlow and P. A. Yoon. Solving recursive tls problems using the rank-revealing ulv decomposition. In *Proceedings of the second international workshop on recent advances in total least squares techniques and errors-in-variables modeling*, pages 117–126, Philadelphia, PA, USA, 1997. Society for Industrial and Applied Mathematics.

[21] M. Gu and S. C. Eisenstat. A stable and fast algorithm for updating the singular value decomposition. Technical report, Yale University, New Haven, 1994.

[22] G. H. Golub and C. F. Van Loan. *Matrix Computations*. The Johns Hopkins University Press, 1990.

[23] W. H. Press, S. A. Teukolski, W. T. Vetterling, and B. P. Flannery. *Numerical Receipes in C++*. Cambridge University Press, 2002.

[24] B. Finkemeyer, T. Kröger, and F. M. Wahl. Executing assembly tasks specified by manipulation primitive nets. *Advanced Robotics*, 19(5):591–611, June 2005.