# COMPARISON OF METHODS FOR STATE ESTIMATION AND ON-LINE IDENTIFICATION IN SPEED AND POSITION CONTROL LOOPS

S. Beineke, F. Schütte, H. Grotstollen

University of Paderborn, Germany

Abstract. For high performance speed and position control of drives featuring elasticity observation of inner states becomes necessary, if only motor speed and current can be measured. In the presence of time-varying inertia or friction on-line identification of these parameters is also needed. In this paper different approaches for these tasks are compared which base on extended Kalman filters, extended Luenberger observers, basis function networks and their combinations.

Keywords. Speed control, time-varying inertia, on-line identification, state estimation, observers, Kalman-filters, basis function networks

## 1 INTRODUCTION

The speed and position control loops include the dynamics of the mechanical transfer elements and the working machine to be controlled, both of which can feature time-varying inertia or some kind of mechanical imperfectness such as elasticity, backlash and friction. For high dynamic speed control of electrical drives these mechanical imperfections have to be considered and the controller has to be adapted to the time-varying inertia. For many industrial applications the mechanic can be modelled as one-mass system or, in the presence of elasticity, as two-mass system, while neglecting backlash. For the latter system high dynamic control can be attained by active vibration damping. Both, the model of the two-mass system and one basic control scheme, proposed e.g. in [1],[2], are given in the signal flow chart of Fig.1.
The parameters of the system are motor side inertia $J_M$, load side inertia $J_L$, as well as the stiffness $C_F$ and damping constant $D_F$ of the shaft. The shaft torque $M_W$ and/or the difference speed between motor and load, $\Delta\omega = \omega_L - \omega_M$, are fed back for active vibration damping. Delay times are given by additional filters ($T_{F1}$, $T_{F2}$), discrete realization ($T_t$) and bandwidth of current control loop ($T_{Ei}$). $K_M$ is the proportional factor between motor current $i_q$ and torque $M_M$.

Because often only current and speed of the motor are available, for elastic systems an observer is used to estimate the non measurable states, such as load speed, shaft torsion and load torque.
If the load inertia is time-varying, adaptation of the controller gains becomes necessary implying on-line identification of the mechanical parameter $J_L$.
In the presence of deterministic load characteristics, e.g. friction, the observer can be extended, e.g. by a basis function network, to identify this characteristic. With this additional measure errors of the observers states can be reduced and the identification result can be used to improve control by feedforward compensation [3].
Therefore this paper deals with the following tasks for identification and observation in high dynamic speed control:

- Observation of load torque and, in the presence of elasticity, load speed and shaft torque
- On-line identification of time-varying mechanical inertia
- On-line identification of friction as an example for deterministic load characteristics

For these tasks a number of schemes from different engineering disciplines, e.g. signalprocessing, system theory or neural computing, seem to be suitable. Kalman filters, Luenberger observers, as well as their nonlinear („extended“) versions, and basis function networks are selected in this paper for comparative investigations. Results of these selected methods are presented for applications and conclusions are derived.

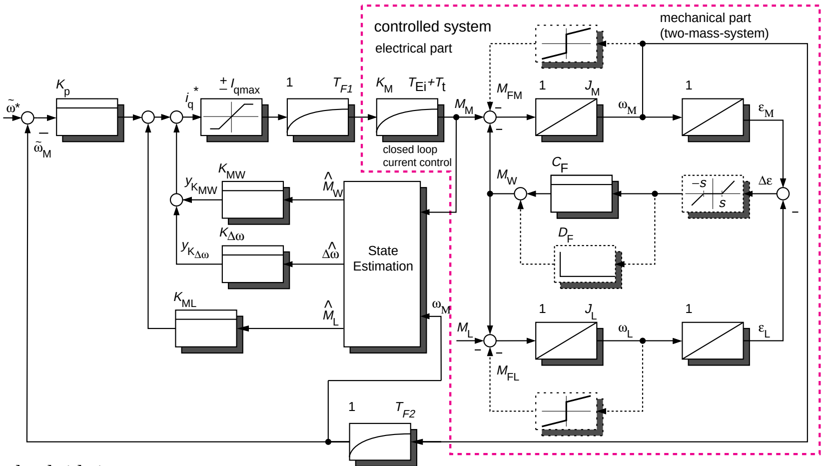


Fig. 1: Two-mass system and basic control scheme for active vibration damping

All concepts are implemented on a DSP-board with TMS320C40 and measurements are carried out on suitable mechanical laboratory set-ups [4]. The main data of the synchronous motor are $I_{qmax} = 9{,}5A$, $K_M = 1{,}27Nm/A$, time constant of current control $T_{Ei} = 300\mu s$. The sampling time of the speed control is $\Delta T = 500\mu s$; values of mechanical parameters can be taken from the results.

## 2 GENERAL DISCUSSION OF STATE OBSERVATION AND IDENTIFICATION

First observation and identification will be discussed on a general base to give some definitions and to show relationships between the concepts from different disciplines, mentioned in the introduction and applied in the following chapters.
From the view of system theory or signal processing technical systems can be described by the differential equations

$$d\boldsymbol{x}/dt = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}) + \boldsymbol{q} \qquad \boldsymbol{y} = \boldsymbol{h}(\boldsymbol{x}, \boldsymbol{u}) + \boldsymbol{r} \quad , \tag{1}$$

where the states are denoted as $\boldsymbol{x}$, measurable inputs as $\boldsymbol{u}$ and measurable outputs as $\boldsymbol{y}$. $\boldsymbol{q}$ and $\boldsymbol{r}$ are noise sources and their meanings are outlined below. The technical system is modelled by

$$d\hat{\boldsymbol{x}}/dt = \boldsymbol{f}(\hat{\boldsymbol{x}}, \boldsymbol{u}) \qquad \hat{\boldsymbol{y}} = \boldsymbol{h}(\hat{\boldsymbol{x}}, \boldsymbol{u}) \quad , \tag{2}$$

where the estimated values are indicated by the hat. System and model are depicted in the signal flow chart of Fig. 2. The mathematical states $\boldsymbol{x}$ can represent physical states $\boldsymbol{x}_S$ of the system (e.g. shaft torque, load torque, load speed) or parameters $\boldsymbol{x}_P$ of the system (e.g. load inertia).

They can also be weights of a *neural network* or other adaptive schemes, where each weight $\tilde{\boldsymbol{w}}$ can be given a certain physical meaning, depending on the network´s structure. This yields $\boldsymbol{x} = (\boldsymbol{x}_S, \boldsymbol{x}_P, \tilde{\boldsymbol{w}})$ and therefore observation of states, identification of parameters and tuning the weights of a neural network can be considered as similar tasks. The prediction error $\boldsymbol{e}$, which is the difference between output signals of the technical system and the model, is caused by two disturbances, regarded as noise sources. Model errors occuring from inaccurate modelling or simplifications of the system equation $\boldsymbol{f}(\hat{\boldsymbol{x}}, \boldsymbol{u})$ are considered by the system noise $\boldsymbol{q}$ and errors of the measurement equation $\boldsymbol{h}(\hat{\boldsymbol{x}}, \boldsymbol{u})$ , caused e. g. by the measurement facilities, are considered by the measurement noise $\boldsymbol{r}$. The prediction error **e** is weighted by a matrix $\boldsymbol{K}$ and fed back to adapt the states of the model. Thus the update results as:

$$d\hat{\boldsymbol{x}}/dt = \boldsymbol{f}(\hat{\boldsymbol{x}}, \boldsymbol{u}) + \boldsymbol{K} \cdot \boldsymbol{e} \qquad \boldsymbol{e} = \boldsymbol{y} - \hat{\boldsymbol{y}} \tag{3}$$

As mentioned above, the mathematical model, given by Eq. (1) to Eq. (3), is suitable to describe the behaviour of many adaptive schemes, such as Kalman filters, adaptive filters, Luenberger observers, basis function networks, adaptive look-up tables, feedforward and recurrent neural networks and is therefore called the *general network* in the following. Depending on the application $\hat{\boldsymbol{x}}$ is called the state or the weight vector. In the following states and weights are used equivalently.

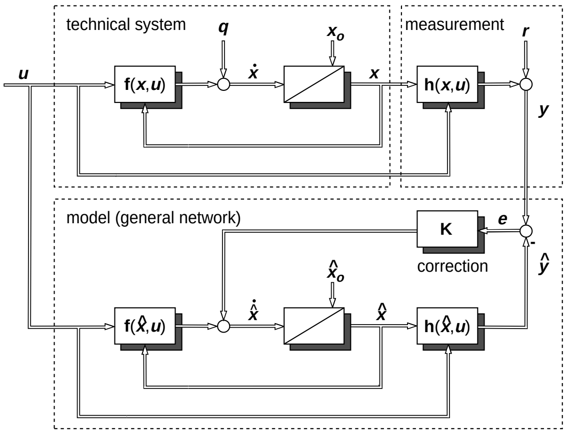


Fig. 2: Signal flow chart of system and *general network*

To classify different methods some definitions for special variants on this general network are given first:

The network is called *linear*, if the system´s dynamics can be represented as $\boldsymbol{f}(\hat{\boldsymbol{x}}, \boldsymbol{u}) = \boldsymbol{A} \cdot \hat{\boldsymbol{x}} + \boldsymbol{B} \cdot \boldsymbol{u}$ and $\boldsymbol{h}(\hat{\boldsymbol{x}}, \boldsymbol{u}) = \boldsymbol{C} \cdot \hat{\boldsymbol{x}} + \boldsymbol{D} \cdot \boldsymbol{u}$ .

It is called *linear in the weights*, if a representation is possible as $\boldsymbol{f}(\hat{\boldsymbol{x}}, \boldsymbol{u}) = \boldsymbol{A}(\boldsymbol{u}) \cdot \hat{\boldsymbol{x}} + \boldsymbol{B}(\boldsymbol{u})$ and $\boldsymbol{h}(\hat{\boldsymbol{x}}, \boldsymbol{u}) = \Phi(\boldsymbol{u}) \cdot \hat{\boldsymbol{x}}$ . Note, that the weights $\hat{\boldsymbol{x}}$ appear linear at the system´s output and therefore the well known linear optimization theory can still be applied although a nonlinear transformation of the network´s inputs is carried out.

The next important distinction is: The system model is called *feedforward network* or *statical network*, if $\boldsymbol{f}(\hat{\boldsymbol{x}}, \boldsymbol{u}) = \boldsymbol{0}$ , otherwise it is called *recurrent network* or *dynamic network*.

Thus, the general model of Fig. 2 is a nonlinear, dynamic network.

Further distinctions are given by determination of matrix $\boldsymbol{K}$. An *on-line, recursive optimization* of $\boldsymbol{K}$ regarding the assumptions about system noise $\boldsymbol{q}$ and measurement noise $\boldsymbol{r}$ is carried out by Kalman filters for dynamic networks, leading to different recursive least squares algorithms for statical networks. An *off-line calculation* of $\boldsymbol{K}$, e.g. by pole placement for linear systems, is carried out to design Luenberger observers for dynamic networks, or similar to find learning rates for least mean square algorithms, when the network is statical.

Backpropagation or other nonlinear techniques are not subject of this paper because convergence is very critical and a systematic analysis is still missing. Therefore they seem not to be suitable for on-line tasks, yet**.**

Generally parameter identification requires sufficient system excitation, known as *persistent excitation*, because otherwise correlation between measurement data will be high and the estimation will be very slow, or even not consistent. Therefore additional measures become necessary.

## 3 PRACTICAL APPLICATIONS: ONE-MASS SYSTEMS

A mechanical system can be regarded as stiff, if for the product of its eigenfrequency $\omega_e$ and the sum of the delay times $T_\Sigma$, given by torque control loop, filtering and digital implementation, the relation $\omega_e T_\Sigma > 1{,}5$ holds [1]. In this case, elasticity can be almost neglected for high performance control and the model can be reduced to an one-mass system. The one-mass system is described by one of the following, equivalent differential equations

$$d\omega/dt = (1/J_\Sigma)(K_M \cdot i_q - M_L) \quad \text{(product form)}, \tag{4}$$

$$d\omega/dt = (1/J_\Sigma \cdot K_M \cdot i_q) - (1/J_\Sigma \cdot M_L) \quad \text{(sum form)}, \tag{5}$$

where load and motor side inertias are combined to the sum of inertia $J_\Sigma = J_L + J_M$ and load and motor speeds are assumed to be identical, $\omega = \omega_M = \omega_L$. For one-mass systems the time-varying inertia $J_\Sigma$ and load torque $M_L$ have to be estimated on-line. For systems containing friction it will be sufficient in most applications to model friction torque as a statical function of speed, $M_L = M_F(\omega)$ , although more advanced friction models, e.g. nonlinear state space representations [3], are available. The identified function can then be used to improve control.

Estimation of inertia and load torque (or friction) can be performed by different networks, which can be either statical or dynamic, and which can either be based on the product form Eq. (4) or the sum form Eq. (5).

### 3.1 Basis function network to estimate friction characteristics

The statical friction characteristic $M_L = M_F(\omega)$ can be learned by a basis function network with speed as input, $u = \omega$, and estimated friction as output, $\hat{y} = \hat{M}_L$. The input/output relation is given by a weighted sum of the components of the transformed input vector $\Phi_i(\omega)$ , see Fig. 3:

$$\hat{y}(u) = \hat{M}_L(\omega) = \sum \Phi_i(\omega) \cdot \hat{w}_i = \Phi^T(\omega) \cdot \hat{\boldsymbol{w}} \tag{6}$$

Note, that usually the output of the network should be normalized,

$$\hat{y}(u) = \frac{\Phi(\omega)}{\sum \Phi_i(\omega)} \cdot \hat{\boldsymbol{w}} \ , \tag{7}$$

to obtain a smooth function approximation. Throughout this paper only basis functions with triangular shape are applied, so that the relation $\sum \Phi_i(\omega) = 1 \forall \omega$ is valid. The basis functions can be derived from the identification results, given in Fig. 6(c) and Fig. 10(c).

An error signal, which is the difference between the output of the network and a measurable value, $e = (M_L - \hat{M}_L)$ , is fed back for adjusting the weights of the basis function networks. In the case that $J_\Sigma$ and $K_M$ are exactly known and measurement works ideal the reference value $M_L$ can be directly derived from Eq. (4) or Eq. (5). Basis function networks are *statical networks*, $\boldsymbol{f}(\hat{\boldsymbol{w}}, \boldsymbol{u}) = \boldsymbol{0}$ , with the weights being equivalent to the states of the network, $\hat{\boldsymbol{x}} = \hat{\boldsymbol{w}}$, and *linear in weights*, $\boldsymbol{h}(\hat{\boldsymbol{w}}, \boldsymbol{u}) = \Phi^T(\boldsymbol{u}) \cdot \hat{\boldsymbol{w}}$ . Thus, many linear optimization methods are applicable. Here adaptation is performed by *least mean squares method* (LMS), in the neural network literature often referenced as *Delta rule*. The update is described with the time-continuous learning rate $\eta_C$ or the time-discrete learning rate $\eta = \eta_C \cdot \Delta T$ by:

$$d\hat{\boldsymbol{w}}/dt = \eta_C \cdot \Phi(\omega) \cdot e \qquad \Delta\hat{\boldsymbol{w}}(k) = \eta \cdot \Phi(\omega(k)) \cdot e(k) \tag{8}$$

Feedforward and training of the network, given by Eq. (6) and Eq. (8), can be represented by the signal flow chart of Fig. 3. Hence, the network behaves like a set of integrators coupled by the basis functions.

Learning is stable as long as the discrete learning rate is chosen between $0 < (\eta \cdot \Phi^T \Phi) < 2$ , while a smaller learning rate allows better filtering. Shape and number of basis functions are selected using a-priori knowledge. Scaling has great influence on the condition of autocorrelationmatrix, and thus on the adaptation results, see e.g. [9].

Estimation of the friction characteristic with a basis function network can be seen as a special case of the more involved task described in 3.2, where inertia is estimated simultaneously. Thus, no additional results are presented here.

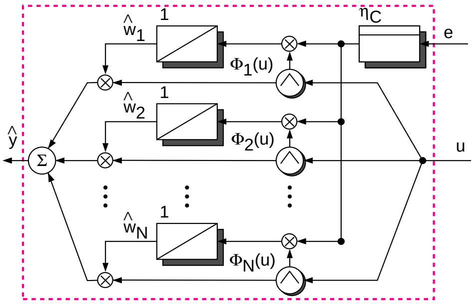

Fig. 3: Signal flow chart of basis function network

### 3.2 Estimation of time-varying inertia and friction characteristic with basis function network

For estimation of time-varying inertia and friction characteristic Eq. (4) is discretized with Euler´s-rule, assuming small sampling times $\Delta T$:

$$\Delta\omega(k) = \underbrace{\frac{\Delta T}{J_\Sigma} \cdot K_M \cdot i_q(k-1)}_{g} - \underbrace{\frac{\Delta T}{J_\Sigma} \cdot M_F(\omega(k-1))}_{f_F(\omega)} \qquad (9)$$

This model is represented by the statical network, shown in Fig. 4, which is still *linear in its weights*. This is the main difference to an approach published in [7], where a highly nonlinear neural network (multi layer perceptron) was used.

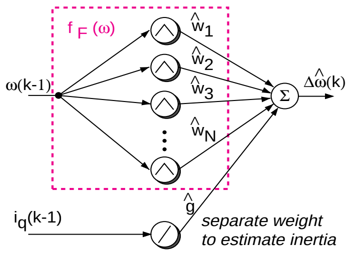

Fig. 4: Structure of basis function network used in 3.2.

In this network $N$ weights are used in combination with a nonlinear transformation of the input $u$, performed by triangular shaped basis functions $\Phi_i$, to estimate $f_F(\omega)$, and the $(N+1)$th weight is used to estimate $g$, which corresponds to the inverse of inertia. The input of the network is given by the vector $\boldsymbol{u} = (\omega, i_q)^T$.

The LMS algorithm is used to adapt the weights:

$$\Delta\hat{w}_i(k) = \eta_1 \cdot \Phi_i(\omega(k-1)) \cdot e(k)$$
$$\Delta\hat{g}(k) = \eta_2 \cdot i_q(k-1) \cdot e(k) \quad ,$$
$$\text{with} \quad e(k) = \Delta\omega(k) - \Delta\hat{\omega}(k) \qquad (10)$$

In order to yield a similar fast convergence for weights $\hat{w}_i$ and $\hat{g}$ the condition of the optimization problem, which is equivalent to the condition of the autocorrelation matrix, has to be good. This can be obtained by either scaling the basis functions or chosing different learning rates. Both measures lead to similar results. Given a desired convergence speed the learning rates can be derived analytically using a stochastic approach [9]. Learning is stable as long as the relation $0 < \eta_1 \cdot \sum \Phi_i^2 + \eta_2 \cdot i_q^2 < 2$ holds.

Usually the learnig rates are set to small values to filter noise. Here, the measured data of $\Delta\omega$ are highly noisy. Optimization of the LMS learning rule is performed by using a training data storage in which the last $Z$ relevant measurement data vectors are first stored and are then taken randomly for the update to avoid strong correlation between subsequently following training data. A distance measure is introduced to distinguish wether a new measurement is considered as relevant and is recorded in the data storage. In this way the training data storage is only updated when the system is excited sufficiently, which is a necessary condition for closed loop operation. In the presence of strong correlation the weights of the network converge very slowly. The influence of the training data storage is principally demonstrated in Fig. 5, where the on-line adaptation of two weights $w_1$, $w_2$ of one network is shown, which is trained without (a) and with (b) a data storage using the same training data.

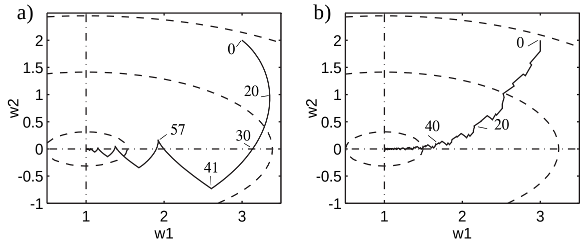

Fig. 5: Convergence of weights, a) without, b) with data storage (The numbers indicate the number of training steps.)

When implemented on a fast signal processor or a parallel hardware not only the actual measured data but also some more $z$ data from the data storage can be used for adaptation during one sampling interval, which reduces the learning time by a factor of $z$ in the mean.

Both measures (data storage, more than one update per data sample) yield a fast on-line adaptation as shown in Fig. 6. The reference speed is changing sinusoidally, with $\omega^*(t) = 140\,rad/s \cdot \sin(7{,}4s^{-1} \cdot t)$, because sinusoidal speed references have been found to be well suited to identify friction characteristics. The results are recorded in a closed speed control loop with PI controller. Parameters used for learning are also given in Fig. 6. All weights are initialized with zero at $t = 0$. Learning rates $\eta_i$ are adapted after $t = 1s$ yielding a fast adaptation at start of identification and good filtering afterwards. The inertia is sufficiently determined after $t = 2s$, see Fig. 6(a). Identification of inertia is necessary before friction can be calculated correctly due to the modelling approach, see Eq. (9), and after one more oscillation of reference speed the friction characteristic is determined in the whole speed interval under consideration, as shown in Fig. 6(c).

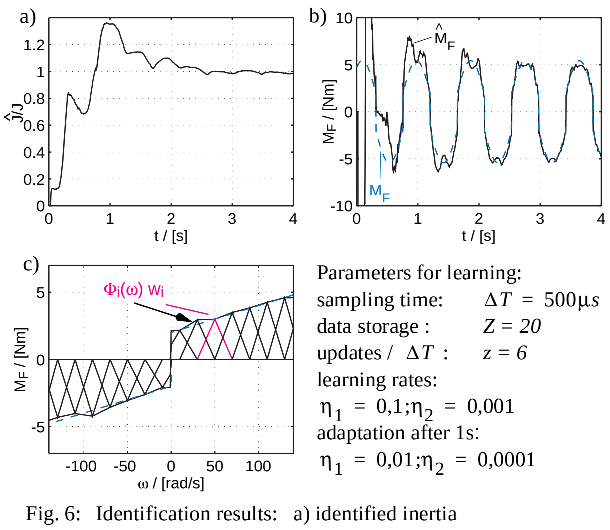

Fig. 6: Identification results: a) identified inertia b) identified friction c) identified friction characteristic

### 3.3 Estimation of friction characteristic with Luenberger observer extended by basis function network

Integration of a general regression neural network (GRNN), which is a special kind of basis function network, in a Luenberger observer, including a stability proof using Lyapunov theory, has been already published in [5]. This approach was also applied to two-mass systems, when measurement of the load speed is feasible [6].

In 4.2 it will be shown how learning rates for basis function networks, which are integrated in an observer, can be simply derived using linearization methods. The basic linear Luenberger observer for the one-mass system leads to a linear dynamic network whose structure is depicted in Fig. 7. The Luenberger feedback gains $\boldsymbol{K} = (k_1, k_2)^T$ can be calculated using pole placement or another linear design technique. Note, that estimation of the load torque is performed by a simple integrator with the prediction error e, weighted by the feedback gain $k_2$, as input.

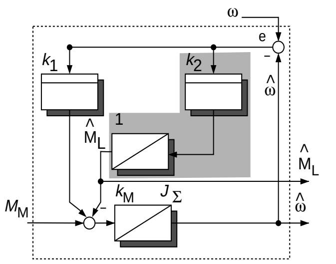


Fig. 7: Structure of Luenberger observer for one-mass system

For estimation of the friction characteristic this simple integrator including the feedback path, drawn with grey background in Fig. 7, is replaced by the basis function network, which is a connectionist structure of integrators as shown in Fig. 3.

The input of the basis function network is the measurable speed $u = \omega$, output is $\hat{y} = \hat{M}_L$ and the error signal is given by $e = \omega - \hat{\omega}$. The task is now to find a learning rate $\eta$, so that the behaviour of the whole observer with the integrated basis function network with regard to model errors is similar to that of the linear Luenberger observer. How this learning rate can be derived, is outlined for the two-mass system in 4.2., which represents a more involved task. No further results are shown here, because identification is not as critical as for the two-mass system.

### 3.4 Estimation of time-varying inertia and load torque with extended Kalman filter (EKF)

The system is described by Eq. (11) leading to a *dynamic, nonlinear network* with input $u = i_q$ and output $\hat{y} = \hat{\omega}$ and three weights (or states) corresponding to speed $\hat{\omega}$, load torque $\hat{M}_L$ and, in order to reduce linearization error, the inverse of inertia $\hat{g} = 1/\hat{J}_\Sigma$.

$$\frac{d}{dt}\begin{bmatrix}\omega\\ M_L\\ g\end{bmatrix} = \boldsymbol{f}(\boldsymbol{x},\boldsymbol{u}) = \begin{bmatrix} g(-M_L + k_M \cdot i_q)\\ 0\\ 0\end{bmatrix} + \begin{bmatrix}q_1\\ q_2\\ q_3\end{bmatrix} \qquad (11)$$

$$y = h(\boldsymbol{x}, u) = \omega + r$$

The EKF [10] has to be applied to the nonlinear model which is based on a linearization of the actual estimation of the weights. The prediction step is calculated by an integration method and the update or correction step has to be performed in a discrete way, if the algorithm is implemented on a digital processor.

The system noises $\boldsymbol{q}$ and $\mathbf{r}$ can be directly taken into account for the filter design by chosing the noise parameters of the covariance matrices $\boldsymbol{Q}$ and $\boldsymbol{R}$. Noise parameter $q_3$ directly determines the dynamics of the parameter estimation of $g$ and the measurement noise $\boldsymbol{r}$ can be calculated from the resolution of the incremental encoder used. For a more detailed description refer to [8].

Identification results are based on a closed loop control with an adaptive PI-speed-controller and are depicted in Fig. 8. A method for „recognition of sufficient excitation“ is implemented to ensure a sufficient accuracy of the estimation. Therefore the signal $d_y$ is derived from the drive torque $M_M$ and is combined with the reference velocity to trigger the parameter estimation, see Fig. 8(c). Fig. 8 shows the step response of the controlled system and the identification results of load torque $\hat{M}_L$ and inertia $\hat{J}_\Sigma$, starting with an unknown $J_\Sigma$.

### Comparison of results and conclusions for one-mass systems

For identification of inertia and observation of load torque in one-mass systems statical or dynamic networks can be applied.

If inertia is known, the networks used to model the system remain *linear* or *linear in weights* and estimation of load torque or load characteristics, such as friction, is not critical. Estimation of load torque is always stationary exact in dynamic networks because an inaccurate estimation of load torque would lead to an increasing output error, which is not the case for statical networks.

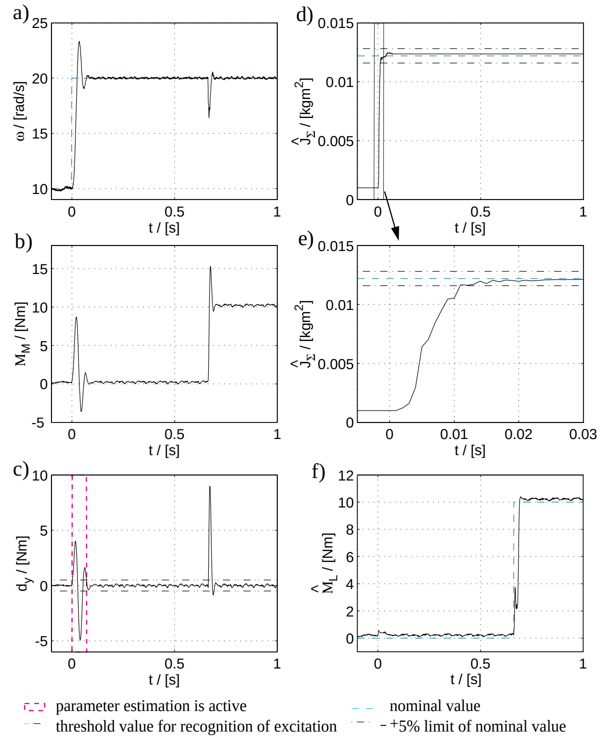


Fig. 8: Results for estimation with extended Kalman filter, showing reference speed and real speed (a), motor torque (b), signal to trigger parameter estimation (c), estimated inertia (d,e) and estimated load torque (f).

If inertia has to be estimated on-line, identification can base on two different models (product oder sum form). Using the sum form and a statical network leads to a basis function network, which is linear in weights although a nonlinear characteristic is identified simultaneously. The sum form features the disadvantage that all weights have to be adapted when the inertia is changing. Thus the product form should be preferred, if inertia varies instaneously. In this case additional, nonlinear adaptation rules have to be applied to ensure stability and convergence.

For every identification method supervision of persistent excitation is required. For statical networks this tasks is done by introducing a data storage, for the Kalman filter estimation of inertia is only active if reference speed signal and motor torque change suitable. Our experiences have shown that Kalman filters can perform well also with very small excitation if the noise parameters have been chosen appropriately.

Basis function networks can be well analyzed and are therefore well suited to learn characteristics on-line. During design one has to take care for good condition of autocorrelation matrix which can be ensured using analytical methods but often also "try and error" immidiately leads to success. To avoid correlation of the input data modified training (e.g. using data storage) is recommended. It is further possible to integrate basis function networks into dynamic networks, which have been designed as Luenberger observers or Kalman filters.

## 4 PRACTICAL APPLICATIONS: TWO-MASS SYSTEMS

For two-mass systems identification of time-varying inertia or deterministic load characteristic is required in combination with observation of non measurable states. Thus, only dynamic (recurrent) networks are suitable. Here Luenberger observers and extended Kalman filters are considered, which become nonlinear for the tasks described in 4.2. and 4.3.

### 4.1 Estimation of load torque with Luenberger observers

For this task disturbance observers are applicable, which can be represented by a linear dynamic network as shown in Fig. 9. Similar to 3.3, the coefficients of the correction matrix $\boldsymbol{K} = (k_1, k_2, k_3, k_4)$ can be calculated by an appropriate linear design technique. Pole placement is chosen here, yielding a Luenberger observer, which can be regarded as state-of-the-art. Structure of observer and feedback path are shown in Fig. 9. Note, that the gain $k_4$ is responsible for weighting the output error of the observer (network) in order to correct the estimated value of the load torque $\hat{M}_L$, leading to the time discrete update:

$$\hat{M}_L(k+1) = \hat{M}_L(k) + \Delta T \cdot k_4 \cdot e(k) \quad ,$$
$$\text{with} \quad e(k) = \omega_M(k) - \hat{\omega}_M(k) \tag{12}$$

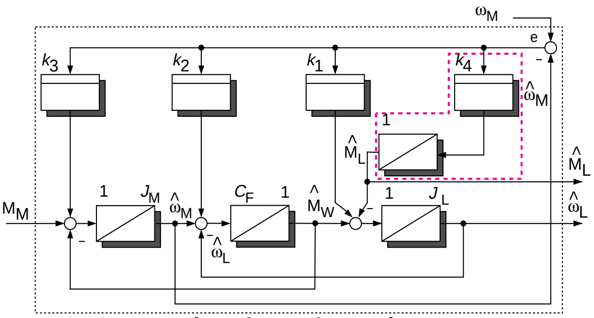


Fig. 9: Structure of Luenberger observer for two-mass system

### 4.2 Estimation of friction characteristic with Luenberger observer extended by basis function network

To estimate the friction characteristic of a mechanical load, the integrator whose output represents the load torque and which is drawn with grey background in Fig. 9, is replaced by a basis function network. Referring to the structure depicted in Fig. 3, the input of the network is the estimated load speed $u = \hat{\omega}_L$, the output is the estimated the load torque $\hat{y} = \hat{M}_L(\hat{\omega}_L)$ and the error signal is derived from measured and estimated motor speed, which is $e(k) = \omega_M(k) - \hat{\omega}_M(k)$ for discrete implementation. Although the approach is very similar to the one used for the one-mass system, the nonlinearity of the network is increased by the fact that the input of the network is now an inner state of the observer, instead of being a measurable value. Therefore learning becomes now a more involved task.

Nevertheless a learning rule for this network can be derived, as shown in [2] and briefed here.
Considering the discrete implementation the input/output relation (prediction) and the update at step *k* become

$$\hat{M}_L(\hat{\omega}_L, k) = \frac{\sum \Phi_i(\hat{\omega}_L(k-1)) \cdot \hat{w}_i(k-1)}{\sum \Phi_i(\hat{\omega}_L(k-1))} \tag{13}$$

$$\Delta\hat{w}_i(k) = \eta \cdot \Phi_i(\hat{\omega}_L(k-1)) \cdot e(k) \quad . \tag{14}$$

Assuming that speed is constant (steady state) or changes only slowly, the following approximations hold:

$$\Phi(\hat{\omega}_L(k)) \approx \Phi(\hat{\omega}_L(k-1)) \quad \text{and} \tag{15}$$

$$\hat{M}_L(\hat{\omega}_L(k-1), k) \approx \hat{M}_L(\hat{\omega}_L(k), k) \tag{16}$$

The output of the network during one sampling intervall is only changed by the weighted output error $(\omega_M - \hat{\omega}_M)$, so that the prediction at step *k+1* becomes

$$\hat{M}_L(\hat{\omega}_L, k+1) = \hat{M}_L(\hat{\omega}_L, k) + \eta \cdot \frac{\sum \Phi^2{}_i(\hat{\omega}_L(k))}{\sum \Phi_i(\hat{\omega}_L(k))} \cdot e(k) \quad . \tag{17}$$

Comparing the update term of Eq. (17) with the update of the common Luenberger observer given by Eq. (12), the whole observer shows the desired behaviour when the learning rate $\eta$ is calculated as

$$\eta = k_4 \cdot \Delta T \cdot \frac{\sum \Phi_i(\hat{\omega}_L(k))}{\sum \Phi^2{}_i(\hat{\omega}_L(k))} \quad . \tag{18}$$

Usually networks are constructed that the expression of Eq. (18) for determing $\eta$ is nearly constant and that a constant learning rate results. With Eq. (18) an analytic equation is given for calculating a learning rate from a Luenberger gain $k_4$, derived by a linear design.

By means of such an extended observer identification of friction characteristics becomes feasible. Load torque estimations of the linear observer and the observer extended by basis function network are compared in Fig. 10(a),(b) where the speed controller is given a sinusoidal reference signal. An additional load torque is applied as indicated. The extended observer performs better if the load change is deterministic. In this case load torque is a function of speed, the load characteristic is identified yielding the friction characteristic shown in Fig. 10(c) as result. After identification of friction the observer can follow the step like changes at zero speed nearly without delay. With the usual linear observer an identification of load characteristic is not feasible. It performs better at stochastic load changes because the load model, a single integrator, fits better.
It is important to note that identification of friction characteristic does not work with every reference signal. E.g. step like references are not well suited, because the assumptions made in Eq. (15) are not valid. At steady state (constant speed) the extended observer works still stable.

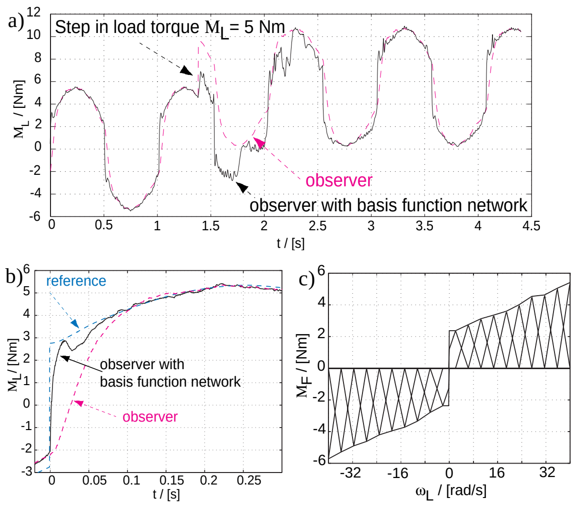


Fig. 10: a) Load torque estimation of linear observer and extended observer for two-mass system with eigenfrequency $f_e = 25 Hz$, $J_L = J_M = 0{,}012 kgm^2$
b) Load torque estimation (enlargement of detail)
c) Identified friction characteristic

### Comparison of results and conclusions for extended observer

Estimation of friction characteristic is a more involved task for two-mass systems than for one-mass systems. If only motor speed can be measured the system model becomes a *nonlinear, dynamic network*. For stochastic variations a simple integrator should be selected as disturbance model, because it is obviously easier to adjust one weight than a network. But especially when the load torque changes quickly and deterministically, as in the case of Coulomb friction, the performance of the observer can be increased by use of a more sophisticated disturbance model. Application of this approach is recommended for off-line identification of load characteristics in closed loop control if suitable references (e.g. sinus) can be applied. On-line identification makes sense if load torque changes can be expressed by a time-varying characteristic and the application demands references that makes identification feasible.

### 4.3 Estimation of time-varying parameters with extended Kalman filter (EKF)

To estimate the non measurable states *and* time-varying parameters, which can be load inertia and stiffness for two-mass systems, the EKF is used instead of the common Luenberger observer. The approach is similar to the one used for the one-mass system, but a more complicated network, which is dynamic and nonlinear, has to be used. The vector of physical states $\boldsymbol{x}_S = (\omega_L, M_L, M_W, \omega_M)$ is augmented with the parameter vector $x_P = (C_F, 1/J_L)$ yielding the weight vector. For a more detailed description refer to [8]. Here only some results are shown in Fig.11 to demonstrate the validity of the approach.

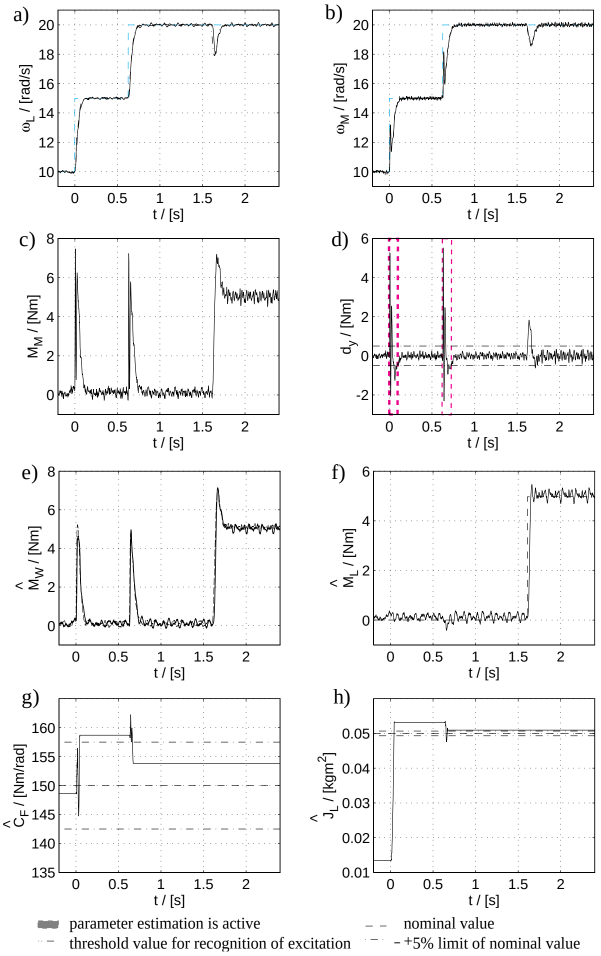


Fig. 11: Results of estimation with extended Kalman filter for a two-mass system, showing reference and real value for load speed (a) and motor speed (b), motor torque (c), signal to trigger parameter estimation (d), estimation of shaft torque (e), load torque (f), stiffness constant (g) and load side inertia (h).

### Conclusions of Kalman filter results

The advantage of the Kalman filter is that time-varying parameters can be regarded as system noise and can be taken into account for the design. Therefore Kalman filters can be more powerful in principal for parameter identification than Luenberger observers. Especially they seem to be well suited when physical parameters and states have to be estimated simultaneously, which requires an adaptation of feedback gain $\boldsymbol{K}$. Experiences have shown that identification with Kalman filters performs also with small excitation, especially when the noise parameters are chosen well. The disadvantage is the great computation effort which results from the on-line adaptation of the correction gain $\boldsymbol{K}$. Especially, when many weights have to be adjusted at once, as e.g. for basis function networks, Kalman filters can not be applied efficiently. In the case of state observation, if the systems model is accurate and linear, Luenberger observers can lead to same results, when appropriately designed. In the case of nonlinearities, which are always present when parameters and states are estimated, adaptation rules for $\boldsymbol{K}$ have to be introduced for observers, which can not be deduced easily. Kalman filters adapt $\boldsymbol{K}$ automatically and behave more robust, which is also true in the case of noise occuring from modelling errors or measurement.

## 5. SUMMARY

For comparative studies of different approaches (observers, filters, other (e.g. neural networks) an unified description is advantageous. Thus, a nonlinear, dynamic network has been introduced. If only parameters have to be identified and no observation of inner states is needed, use of statical networks is feasible, as it has been shown for the one-mass system. If observation and parameter estimation is carried out simultaneously, as for two-mass systems, only dynamic networks can be applied. As expected, no approach can be declared as "the best for every application". Thus, our experiences have been summarized as rule-like short conclusions after presenting the results that can be taken as guidelines when selecting an appropriate algorithm for an own solution.

## ACKNOWLEDGEMENT

Thanks to the German Research Council for financing the project "Selbsteinstellende Antriebsregelung mit neuronaler Hardware". Additionally the authors would like to thank M. Henke, who has been involved in research on basis function networks, and A. Rolfsmeier for investigation and implementation of Kalman filter approaches.

## REFERENCES

1. Schäfer U.: "Development of Nonlinear Speed- and Position Controls for Compensation of Coulomb Friction and Backlash for an Electrical Driven Two-mass System" (in German), Ph. D. Thesis, Techn. University of Munich, 1992.
2. Schütte F., Beineke S., Henke M., Grotstollen H.: Speed Control of an Elastic Two-mass System with On-line Identification of Friction and Active Vibration Damping at Limit of Regulated Variable (in German), SPS/IPC/DRIVES´96, Nov. 1996, Sindelfingen, Germany, pp. 303-315.
3. Armstrong-Helouvry B., Dupont P., Canudas de Wit C.: A Survey of Models, Analysis Tools and Compensation Methods for the Control of Machines with Friction, Automatica, Vol. 30, No. 7, July 1994, pp. 1083-1138.
4. Schütte F., Bünte A., Grotstollen H.: Hardware-in-the-Loop-Simulation of Electro-Mechanical Systems (in German), Fachtagung Leistungselektronische Aktoren und intelligente Bewegungssteuerungen, Magdeburg, Germany, 1996, pp. 151-158.
5. Schröder D., Schäffner C., Lenz U.: Neural-Net Based Observers for Sensorless Drives, IECON 1994, Bologna, Italy, pp. 1599-1610.
6. Frenz Th., Schröder D.: Learning Unknown Nonlinearities Using a Discrete Observer in Combination with Neural Networks, IAS 1995, Orlando, USA, pp. 1800-1806.
7. Lou Y., Holtz J.: Compensation of Friction in Motion Control Systems Using a Neural Network, Proceedings 37th SKI, 1992, Ilmenau, Germany, pp. 322-332.
8. Schütte F., Beineke S., Rolfsmeier A., Grotstollen H.: Online Identification of Mechanical Parameters Using Extended Kalman Filters, IAS 1997, New Orleans, USA.
9. Brown M., Harris C.: "Neurofuzzy Adaptive Modelling and Control", Prentice Hall, 1994.
10. V. Krebs.: "Nonlinear Filtering", *in German*, Oldenbourg Verlag, München Wien, 1980.

## ADDRESSES OF THE AUTHORS

S. Beineke/ F. Schütte/ Prof. Dr.-Ing. H.Grotstollen
Institute for Power Electronics and Electrical Drives
University of Paderborn, FB-14 LEA
Pohlweg 47–49, D-33098 Paderborn, Germany
Tel.: 0049-5251-602212
Fax.: 0049-5251-603443
e-mail: beineke@lea.uni-paderborn.de