Online Identification of Nonlinear Mechanics Using Extended Kalman Filters with Basis Function Networks

S. Beineke, F. Schütte, H. Grotstollen

Institute for Power Electronics and Electrical Drives
University of Paderborn, FB 14-LEA
Pohlweg 47-49, 33098 Paderborn, Germany
Phone: (49)-5251-602212 Fax: (49)-5251-603443 E-Mail: beineke@lea.uni-paderborn.de

Abstract – For high performance speed and position control of electrical drives fast online identification is needed for time-varying inertia or load conditions in combination with adaptive controllers. In this paper Extended Kalman Filters are applied and optimized for deterministic parameter variations by integrating basis function networks into the common structure of the Kalman Filter. It is shown that learning of nonlinear load or parameter characteristics becomes feasible by this measure and the performance of the Extended Kalman Filter can be improved.

I. INTRODUCTION

Speed and position control loops include the dynamics of the mechanical transfer elements and the working machine to be controlled, both of which feature some kind of mechanical imperfection such as elasticity, backlash and friction, sometimes in combination with time-varying parameters, e.g. inertia and/or stiffness. For high dynamic speed control of electrical drives these mechanical imperfections have to be considered and the controller has to be adapted according to the variation of parameters, which implies the need for an online identification scheme. For many industrial applications the mechanic can be sufficiently modelled as a two-mass-system containing elasticity and friction, while neglecting backlash, or even as an one-mass-system. High dynamic control of such systems can be attained by several control concepts, some of them feature active vibration damping [1]. Since only the mechanic can be sufficiently modelled as a two-mass-system containing elasticity and friction, while neglecting backlash, or even as an one-mass-system. High dynamic control of such systems can be attained by several control concepts, some of them feature active vibration damping [1]. Since only current and position of the motor are measured in cost-optimized drives, the use of adaptive schemes, such as observers, filters or “neural” networks, becomes necessary to estimate load torque, non measurable states and/or time-varying parameters [7].

In many applications the variation of parameters and load conditions can be described by deterministic characteristics. In the presence of friction the load torque \( M_L = M_f(\omega) \). In feed drives with e.g. screw spindles the stiffness of the system can be expressed by a characteristic of spring constant \( C_f = C_f(\varepsilon_{ep}) \) being depending on the actual position of the drive. For robotic applications variation of inertia is often given as a function of position, \( J_L = J_L(\varepsilon_{ep}) \), if certain trajectories are demanded.

Many methods are known to solve the problem of state and parameter estimation, see e.g. [3]. Here, the Extended Kalman-Filter (EKF) approach is selected for adaptation and it is combined with Basis Function Networks (BFN) to store the new information of the adaptation. Since the whole model combines the features of adaptation and memorization, it can be called a “learning model”. In order to demonstrate the main idea of this approach, it is applied to relatively simple tasks occurring frequently in one-mass systems.

II. ONLINE IDENTIFICATION USING EXTENDED KALMAN FILTERS

Because details about implementation of the EKF have been described in [6], only a short overview is given in this chapter. The Kalman Filter technique EKF is selected here for adaptation, because the EKF
  • takes explicitly measurement and process uncertainties into account,
  • makes use of a-priori information and
  • represents an efficient algorithm for real time applications due to its recursive working principle.

The basic idea of the EKF is to expand state vector \( x \) of the dynamic system by vector \( x_p \) of unknown parameters yielding the augmented state vector \( x = [x, x_p]^T \). The estimation algorithm is then applied to the augmented system of nonlinear state equations. To solve this problem several filter techniques exist, all involving some form of approximation. The EKF is an efficient algorithm, where a linear Taylor approximation of the system function at the previous state is performed [5]. The basic structure of the EKF is shown in Fig. 1 where the system is modelled by the nonlinear differential equations

\[
\frac{d}{dt} x = f(x, u, t) + q(t) \quad y = h(x, u, t) + r(t) .
\] (1)

Error \( e \) between the measured data \( y \) and the filter output \( \hat{y} \) is caused by inaccurately modelled dynamics \( f(.) \), denoted as system noise \( q \), and by an incorrect measurement equation \( h(.) \), denoted as measurement noise \( r \). The stochastic processes \( q \) and \( r \) are assumed to be white and characterized by their spectral densities \( Q_c \) and \( R_c \). Knowledge about \( Q_c \) and \( R_c \) is explicitly taken into account for calculating the Kalman gain \( K \). The prediction error \( e \) is weighted by the Kalman gain and

1. Contrary to [6] \( q \) and \( r \) are used instead of \( w \) and \( v \) for system and measurement noise in order to avoid confusion with the weights of a network, also denoted as \( w \) in this paper.
For implementation on a digital signal processor the continuous-discrete EKF is chosen, see Fig. 1, because the prediction step can be performed by any suitable integration method, whereas the update has to be performed discrete, because the observations (measurements) are only available at discrete times. Referring to Fig. 1, the a-priori estimate \( \hat{x}(t_k|t_{k-1}) \) defines an extrapolation (prediction) on the basis of the observations up to the time instant \( t_{k-1} \) and the a-posteriori estimate \( \hat{x}(t_k|t_k) \) gives the states after the correction step (filtering) implying all information available after the measurement at time instant \( t_k \). The respective equations are given in [6], and for a more general discussion on EKF refer to [4], [5].

The process parameters to be identified are usually modelled in the basic EKF as a random walk characterized by differential equation

\[
\frac{d}{dt} \hat{x}_p(t) = 0 + q_p(t) .
\]  

The corresponding diagonal elements in the discrete covariance matrix \( Q = Q_C \cdot T \) used in the Kalman Filter algorithm serve as design parameters to describe how fast the components of \( x_p \) are expected to vary within the sample time \( T \). It is to expect that the bigger these variances are chosen the better time-varying parameters can be tracked but the more noisy are the estimates at steady state (\( x_p \) constant).

### III. EKF With Basis Function Network

The basic EKF is extended by a Basis Function Network (BFN) to store information attained during adaptation. The output \( \hat{y}_N(t_k) = \hat{y}_N(\hat{x}(t_k), u_N(t_k)) \) of the BFN is given by the weighted sum of the transformed input vector components \( \Phi_i(\hat{x}(t_k), u_N(t_k)) \) abbreviated as \( \Phi_i(t_k) \) in the following:

\[
\hat{y}_N(t_k) = \sum \Phi_i(t_k) \hat{w}_i = \Phi^T(t_k) \hat{w} .
\]  

(3)

The components of the input vector can be measurable inputs \( u_N \) as well as estimated states \( \hat{x} \) of the EKF and the transformation is determined by the selected basis functions \( \Phi_i \). A typical network (BFN) with 20 piecewise linear, local basis functions (linear splines), suitable to learn a nonlinear characteristic and used throughout this paper, is shown in Fig. 2. By using this network \( \sum \Phi_i = 1 \) is valid for all inputs and therefore normalization is not necessary.

For the design of the EKF each weight \( \hat{w}_i \) of the network can be considered as an additional state of the augmented state vector yielding \( \hat{x} = [\hat{x}_S, \hat{x}_P, \hat{w}]^T \). In practice this approach is not feasible because the computation effort is largely blown up: e.g. usually 20 or more weights are used for the network and the number of computations of the EKF can increase quadratically with the number of its states. Instead, the whole network is considered as one single state \( \hat{x}_N \) for the EKF algorithm with \( \hat{x}_N \in [\hat{x}_S, \hat{x}_P]^T \) and the update of the network is then performed using the simple least mean squares (LMS) algorithm

\[
\Delta \hat{w}(t_k) = \eta \cdot \Phi(t_k) \cdot (y_N(t_k) - \hat{y}_N(t_k)) \quad ,
\]  

(4)

which usually needs only a few computations, especially if local basis functions are used. The reference of the BFN, the training data \( y_N(t_k) \), is determined by the EKF. This simplification does not lead to worse results, provided that the network has been designed appropriately.

A BFN is able to track exactly the given training data \( y_N(t_k) \) by adapting the learning rate \( \eta \) at each sample step

\[
\eta(t_k) = \tilde{\eta} / (\Phi^T(t_k) \Phi(t_k)) \quad \text{with} \quad \tilde{\eta} = 1 ,
\]  

(5)

which is called the normalized least mean squares method (NLMS) [8]. A learning rate \( \tilde{\eta} < 1 \) yields slower tracking, but additional filtering of the training data.

Input-output relation (3) and learning algorithm (4) are given as signal flow chart inside the grey frame of Fig. 1, which is also the basis of the following discussion.

To simplify notation it is defined that \( M+N \) states, estimated by the Kalman-Filter, are augmented to the vector \( \hat{x} = [\hat{x}_{KF}, \hat{x}_N]^T \), whereas \( N \) states \( \hat{x}_N \) of the EKF are also

---

**Fig. 1:** General structure of Extended Kalman Filter (EKF) in combination with a Basis Function Network (BFN)

**Diagram Description:**

- **Dynamic System:** Inputs \( f(x,u) \) and \( r(t) \) feed into a system with states \( x(t) \) and \( y(t) \).
- **Integration:** The state evolution is described by the differential equation \( \frac{dx}{dt} = f(x,u) \).
- **Measurement:** Observations \( m(x) \) are fed into the system.
- **Extended Kalman Filter (EKF):** Processes \( x(t) \) and \( y(t) \) to estimate \( \hat{x}(t) \).
- **Basis Function Network (BFN):** Interacts with the EKF to provide \( \hat{y}_N(t_k) \).
- **Correction:** Adjusts the estimate \( \hat{x}(t) \) based on observations.
- **Normalization:** Ensures \( \sum \Phi_i = 1 \) for inputs.

---
used as training data \( y_N \) of the BFN:

\[
y_N(t_k) = M_{IN} \cdot \hat{x}(t_k | t_k) \quad \text{with} \quad M_{IN} = [\theta_{N,M} I_N],
\]

(6)

Identity matrices of size \( k \) are denoted by \( I_k \) and \( \theta_{k \times l} \) symbolizes a zero matrix with \( k \) rows and \( l \) columns. For fitting the BFN into the EKF two main structures are defined, depending how the output of the network \( y_N \) is processed by the EKF. These structures are given by Fig. 1 in combination with matrices \( M_{KF} \) and \( M_{OUT} \) as chosen below.

1. Parallel structure:

\[
M_{KF} = \begin{bmatrix} I_M & 0_{M \times N} \\ 0_{N \times M} & 0_{N \times N} \end{bmatrix} \quad \text{and} \quad M_{OUT} = \begin{bmatrix} 0_{M \times N} \\ I_N \end{bmatrix}
\]

(7)

The filtered EKF estimation is only used to adapt the weights of the network. The BFN works parallel and does not influence the behaviour of the EKF itself.

2. Integrated structure:

\[
M_{KF} = \begin{bmatrix} I_M & 0_{M \times N} \\ 0_{N \times M} & 0_{N \times N} \end{bmatrix} \quad \text{and} \quad M_{OUT} = \begin{bmatrix} 0_{M \times N} \\ I_N \end{bmatrix}
\]

(8)

Additionally the characteristic stored in the BFN is used for the prediction step of the EKF, because prediction is performed by the BFN instead of the EKF yielding also additional filtering for \( \tilde{F} < 1 \).

The integrated structure can be further distinguished, if the parameter change is still considered as random walk, as given by (2), or if the information stored in the network is considered in the system matrices by modelling variations of the parameters \( \hat{x}_P = x_N \) as

\[
\frac{d}{dt} \hat{x}_P = \frac{d}{dt} x_N + q_P = \frac{d}{dt} y_N(x, u) + q_P
\]

(9)

\[
\frac{\partial \hat{y}_N}{\partial x} \cdot \frac{dx}{dt} + \frac{\partial \hat{y}_N}{\partial u} \cdot \frac{du}{dt} + q_P
\]

Altogether three approaches result, which will be outlined in more detail by examples given in the following chapters.

IV. GENERAL ASPECTS OF THE EXPERIMENTS

The EKF with BFN is applied to the following tasks occurring in many industrial applications.

- Task 1: Online identification of a nonlinear inertia characteristic depending on the position and observation of load torque for an one-mass system.
- Task 2: Online identification of time-varying mechanical inertia and a load characteristic (friction) for an one-mass system.

The basic algorithms of these tasks are given by an EKF, which estimates load torque \( M_L \) and the inverse of the total inertia \( g = 1 / J_L \) simultaneously, yielding the augmented state vector \( \hat{\xi} = [\hat{\xi}_M, \hat{\omega}_M, M_L, \hat{\xi}]^T \). According to [6] the inverse inertia is chosen as state variable to reduce the linearization error. The respective component of the state vector is then replaced by a BFN for the examples as given below.

As input of the EKF the current component \( i_q \), which is (nearly) proportional to motor torque, and as output the position \( \epsilon_M \) are chosen yielding the measurement equation

\[
\hat{y} = h(\hat{x}, u) = \hat{\epsilon}_M.
\]

All identification results, except where otherwise denoted, are based on a closed loop cascaded position control with an adaptive PI-speed-controller and a feedforward scheme to reduce the tracking error, as depicted in Fig. 3. In order to filter measurement noise, the estimates \( \hat{\epsilon}_M \) and \( \hat{\omega}_M \) of the EKF are fed back into the controller instead of measurements \( \epsilon_M \) and \( \omega_M \). With this measure the performance of the closed loop control becomes quite insensitive to the resolution of the incremental encoder and high performance can be achieved also with lower resolution, lower costs encoders.

As for every online parameter identification scheme “sufficient excitation” has to be guaranteed to obtain consistent estimation results. Note, that sufficient excitation is provided by the reference trajectories given in the examples and thus identification is active during the whole time. Nevertheless the sufficient excitation is recognized by a method presented in [6], where a signal, which is derived from the drive torque \( M_M \)
and combined with the reference velocity $\omega_M^*$, triggers the parameter estimation.

The algorithms are implemented on a DSP board (dSPACE DS1003) comprising a TMS320C40 and measurements are carried out on suitable mechanical laboratory set-ups [2]. A permanent magnet synchronous motor (PSM) with the data $J_{L,off} = 0.0137\,\text{kgm}^2$, $I_{q,\text{max}} = 15\,\text{A}$, $K_M = 1.272\,\text{Nms/A}$, and $M_N = 7\,\text{Nm}$ performs as actuator. Note, that the torque ripple for $i_q = \text{const.}$ due to slots and pole harmonics achieves up to 6% of $M_N$ and causes additional system noise. Referring to Fig. 3 the time constants of filters and delays are given by $T_{E,i} = 0.2\,\text{ms}$, $T_{F,j} = 1\,\text{ms}$, $T_{F,n} = 1.5\,\text{ms}$ and the controllers are specified by the parameters $K_L = 28\,\text{s}$, $T_N = 8.9\,\text{ms}$, $K_p(t) = J_{E}(t)/2T_EK_M$.

V. IDENTIFICATION OF INERTIA CHARACTERISTIC
AND OBSERVATION OF LOAD TORQUE

In a first example it is assumed that the inertia varies deterministically with the position, e.g. a robot manipulator has to follow a certain trajectory and the inertia seen from one single drive varies with time, which can be described by $J_M(\xi_M)$. The characteristic is identified by a BFN with input $u_N = \xi_M$, basis functions $\Phi(\hat{x},u_N) = \Phi(u_N) = \Phi(\xi_M)$ as depicted in Fig. 2, and output $\hat{y}_N = \hat{g}_N(\xi_M)$.

The system equation of the EKF becomes

$$
\begin{bmatrix}
\dot{\hat{\xi}}_M \\
\dot{\hat{\omega}}_M \\
\dot{\hat{\xi}}_N
\end{bmatrix} = 
\begin{bmatrix}
f(\hat{x},u) \\
\hat{g}_N(\xi_M) \\
\hat{g}_M(\xi_M)
\end{bmatrix} + 
\begin{bmatrix}
\hat{\omega}_M \\
0 \\
0
\end{bmatrix} 
\begin{bmatrix}
-M_L + k_M \cdot i_q \\
0 \\
0
\end{bmatrix}
\frac{d\hat{g}_M(\xi_M)}{dt} 
\text{or [0]}
\end{array}
$$

(11)

The measurement noise $r$ can be calculated directly from the resolution of the incremental encoder used, see [6], yielding $R = 5.75 \cdot 10^{-10} \, \text{rad}^2$.

The variance matrix $Q$ of the system noise is a measure for model uncertainties and additional disturbances. Here $q_{33}$ determines directly the dynamic of the estimation $\hat{\xi}_M$ and $q_{44}$ the respective dynamic of $\hat{\xi}_N$. $Q$ is derived by use of some assumptions yielding

$$
Q(t_k) = \text{diag}(0, 5e^{-5}(\text{rad/s})^2, 5e^{-4}(\text{Nm})^2, q_{44}(t_k))
$$

(12)

Parameter $q_{11}$ can be set to zero because the equation $d\hat{\xi}_M/dt = \omega_M$ contains obviously no noise. $q_{44}$ is adapted according to the actual estimated inertia as

$$
q_{44}(t_k) = \hat{q}_{44} \cdot (\hat{g}(\xi_k)/J_{off})^2.
$$

(13)

Because inertia changes very fast during this experiment, $\hat{q}_{44}$ is set to the high value $q_{44} = 5\,\text{kg}\cdot\text{m}^2$.

The results presented in Fig. 4 are obtained by Hardware-in-the-loop simulation, see [2], where the reaction torque due to time-varying inertia and the load torques are calculated online and produced by another stiff coupled, torque controlled PSM-drive. A predefined trajectory is given as reference value for the position, as it can be derived from Fig. 4(a), where reference $\xi_M^*$, estimated $\hat{\xi}_M$ and actual $\xi_M$ positions are shown (no visible difference between these values). In Fig. 4(c) actual $\omega_M$ and estimated speed $\hat{\omega}_M$ are depicted (also no difference visible), while in Fig. 4(b) the current $i_q$ is shown. Note, that excitation is recognized as “sufficient” during the whole time interval. To explain the current signal, the varying inertia and the additional feedforward control mechanism must be taken into account. Because the total inertia increases during the positive acceleration phases, which is equivalent to increasing energy given by $0.5 \cdot J_M \omega_M^2$, current $i_q$ is limited to its maximum value. In Fig. 4(d) the nominal inertia $J_L(t)$ and the estimated value $\hat{J}_L(t)$ are depicted. To demonstrate the efficiency of the method and to show the different estimation results between the parallel and the integrated structure, a very fast variation of inertia is chosen. Within $0.7\,\text{s}$ $J_M$ is changing by factor 10. In order to track the fast variation a relatively high value is determined for the noise parameter, which determines the dynamic of the estimation $\hat{J}_L$ leading to a relative noisy estimate. During the time interval $[0, t_1]$ the parallel structure, defined by $M_{KF} = I_A$ and $M_{OUT} = \theta_{441}$, is chosen. The estimation result of the basic EKF is plotted as bold dashed line in Fig. 4(d). During this interval the estimated inertia is only stored in the network, but not used for prediction.
At $t_1$ the nonlinear inertia characteristic displayed in Fig. 4(e) is stored in the network, which can not be improved by further estimation with the parallel structure. An improvement is only attainable using the integrated structure. In order to demonstrate the improvement the estimation algorithm is switched at $t_1$ to the integrated structure given by

$$
M_{KF} = \begin{bmatrix}
I_3 & \theta_{3x3} \\
\theta_{1x3} & 0
\end{bmatrix} \quad M_{OUT} = \begin{bmatrix}
\theta_{3x1} \\
1
\end{bmatrix} \rightarrow \hat{\eta}_N = \hat{g}(\eta), \quad (14)
$$

that within $[t_1, t_3]$ the network-based estimation of $J_\Sigma$ is depicted as solid line. For the integrated structure the system function $f_\Sigma$ should theoretically consider the information

$$
\frac{d\hat{\eta}_N(\varepsilon_M)}{dt} = \frac{d\hat{\eta}_N(\varepsilon_M)}{d\varepsilon_M} \frac{d\varepsilon_M}{dt} \approx \frac{d\hat{\eta}_N(\varepsilon_M)}{d\varepsilon_M} \omega_M
\quad (15)
$$

But in many practical cases the simplification of setting this derivation to zero leads to very similar and sometimes even better results and therefore it is used here. Nevertheless, simulations have shown that the approach (15) can be advantageous, especially for fast parameter variations and low system noise. A simulative comparison is given in Fig. 5, but more investigations are still needed. Because the derivation $\frac{d\hat{\eta}_N(\varepsilon_M)}{d\varepsilon_M}$ has to be calculated, it seems to be suitable to use higher order polynomials (e.g. B-Splines [8]) instead of linear basis functions to yield smoother characteristics.

At the beginning the identification works with a (normalized) learning rate of $\eta = 1$, see (5) and estimation noise remains constant, see Fig. 4 for $[t_1, t_2]$. Because the EKF can make use of the characteristic stored in the network yielding lower system noise $q_d(t)$, the learning rate is set to a lower value at $t_2$ ($\tilde{\eta} = 0.1$). Note, that approximately a multiplication of the learning rate $\eta$ by factor $k$ is similar to multiplication of the respective noise parameter by factor $k^2$, which means a degree of freedom for design. Then the state used for the prediction step of the EKF is additionally filtered by the BFN leading to a smoother curve of the identified inertia, see Fig. 4(d) for $t = [t_3, t_4]$. Fig. 4(f) shows the stored characteristic at the end of the experiment ($t_5$).

Note, that usually the two step method, applied here, 1. using the parallel structure to obtain a rough estimate of the characteristic, 2. improving the results by switching to the integrated structure and reducing the learning rate, yields the fastest adaptation combined with the best identification results.

These results lead to the conclusion that especially for small excitation and fast parameter variations better results are obtained using the integrated instead of the parallel structure. But in many cases the parallel one will be sufficient, because the identification with a basic EKF related to the variation of the parameter which has to be estimated will be fast enough. Because of the implementation and the computational effort the basic EKF should be preferred in these cases.

VI. ONLINE IDENTIFICATION OF TIME-VARYING INERTIA AND FRICTION CHARACTERISTIC

In the following a friction characteristic is identified as an example for a nonlinear load. This example is especially well suited to demonstrate how identification results can be improved by integrating the BFN into the EKF. In principal this improvement can be explained by the increasing modeling capabilities when combining EKF with BFN. The state equation is now given by (16), in which compared to (11) the order of states $\hat{\eta}$ and $M_L$ is only changed due to formal reasons, in order to keep notation in accordance to previous chapters.

$$
\begin{bmatrix}
\hat{\varepsilon}_M \\
\hat{\omega}_M \\
\hat{M}_{LN}(\hat{\omega}_M)
\end{bmatrix} = \begin{bmatrix}
f(\hat{x}, u) = -\left(\hat{g} \cdot \hat{M}_L + \hat{g} \cdot (k_M \cdot i_q)\right) \\
0 \\
0
\end{bmatrix} \quad (16)
$$

The friction characteristic is identified by a BFN with input $\hat{x} = \hat{\omega}_M$, basis functions $\Phi(\hat{x}, \eta_N) = \Phi(\hat{x}) = \Phi(\hat{\omega}_M)$ as depicted in Fig. 2, output $\eta_N = M_{LN}(\hat{\omega}_M)$ and $\tilde{\eta} = 0.1$.

The noise parameters are selected equal to chapter V

$$
Q(t_k) = diag(0, 5e^{-5}(rad/s)^2, q_{33}(t_k), 5e^{-4}(Nm)^2) \quad , (17)
$$

except the noise parameter for inertia variation is set to a lower value in order to reduce the influence of Coulomb friction on inertia identification, as discussed below:

$$
q_{33}(t_k) = \hat{q}_{33} \cdot (\hat{g}(t_k)/\Sigma_{off})^2 \cdot \hat{q}_{33} = 0.001kg m^{-2} \quad (18)
$$

Experiments are carried out using the PSM and a second motor connected by a synchronous belt and a mechanical brake at the second motor. The resulting eigenfrequency of $f_e = 85Hz$ means an additional system noise because this dynamic is not modelled inside the EKF.

To compare the results of the basic EKF (parallel structure) with those obtained by the integrated structure (EKF with BFN) identification is performed using a non adaptive controller. Again, a trajectory is given as reference for the position, see Fig. 6(a), yielding the courses of speed as shown in Fig. 6(b). Because Coulomb friction causes a (nearly) step-like change of the load torque at zero speed, estimation results of
has to be introduced at speeds near zero for avoiding chattering effects known from switching controllers:
\[ \dot{M}_{LN}(\hat{\omega}_M) = \gamma_N(\hat{\omega}_M) = 0 \quad \text{for } |\hat{\omega}_M| < 0.3 \text{ (rad)/s} \quad (19) \]

Identification of friction at speeds very close to zero and especially of stick friction is therefore still not possible, yet.

VII. SUMMARY

The EKF was found to be a powerful algorithm if parameter estimation has to be performed online, especially if states have to be observed simultaneously. It was shown how BFNs can be integrated into the basic EKF and by this integrated structure

- the identification of nonlinear characteristics becomes feasible and
- the prediction and therefore the performance of parameter identification and state observation can be improved.

The computational efforts do not increase largely because simple LMS algorithm can be chosen for training the BFN.

It seems further to be possible to combine the EKFs proposed in [6] with BFNs to obtain online identification schemes for other nonlinear characteristics, even for more complicated mechanics modelled as two-mass systems.

ACKNOWLEDGMENT

The authors acknowledge gratefully the support of the German Research Council (DFG) for financing Project „Selbsteinstellende Antriebsregelung mit neuronaler Hardware“ and A. Rolfsmeier for implementing the algorithms and for his great support investigating EKFs.

REFERENCES


[8] M. Brown, C. Harris: "Neurofuzzy Adaptive Modelling and Con-