PARAMETER ESTIMATION FOR NONLINEAR CONTINUOUS-TIME STATE-SPACE MODELS FROM SAMPLED DATA

C. Bohn
Continental AG, Strategic Technology Department, Hanover, Germany

Keywords: Parameter estimation, system identification, state-space models, state estimation, filtering, Kalman filter, adaptive filtering, nonlinear systems.

Contents

Summary

The problem of parameter estimation for nonlinear state-space models is addressed. Two approaches to this problem are presented: (1) the state-augmentation approach, which consists of including the unknown system parameters in the state vector and estimating them through a state estimator, and (2) the prediction-error approach, which consists of tuning a predictor such that it will give optimal predictions and then recovering the system parameters from this optimum predictor.

For the development of both approaches, the concepts of state-space modeling and state estimation through filters are reviewed. This includes a unified formulation of four nonlinear filters: the extended Kalman filter, the first-order bias-corrected filter, the truncated second-order filter, and the modified Gaussian second-order filter. A discussion of the prediction-error approach stresses the fact that a prediction-error approach combines a predictor and an optimization algorithm with a sensitivity model for the predictor. For the unified filter formulation, such a sensitivity model is derived. The relationship between the two approaches as well as possible modifications and simplifications are discussed.

1. Introduction and Overview

The relevance of the system identification/parameter estimation problem for nonlinear systems has been discussed in the introduction to this topic (see *Identification of Nonlinear Systems*). Many of the identification methods discussed in *Identification of*

Nonlinear Systems focus on black-box models, i.e., models that do not have any direct relationship to the underlying scientific principles governing the behavior of the system (see *Identification of NARMAX and related models*, *System Identification using Neural Networks*, *System identification using Fuzzy Models*, and *System Identification using Wavelets*).

A black-box model is often sufficient, particularly if the focus is on control design and simulation only, or if the system is to complex to be modeled from underlying first-principles. If, however, the focus is on obtaining physical insight, determining the values of meaningful system parameters, or on monitoring the state of a process, a model derived from the underlying principles has to be used.

Often, such a model is given in form of differential equations (see *System Description in Time-Domain*), either in input-output form (incorporating higher derivatives of input and output signals), or in state-space form (as a set of first-order differential equations). The state-space form is especially attractive since the states of a system are often related to physical quantities, such as signals that correspond to stored energy (e.g., the voltage over a capacitor, the current through an inductor, displacement of a spring, velocity of a mass, etc.)

The estimation of parameters for input-output descriptions, i.e., differential equations, is discussed in the article preceding this one (see *Parameter Estimation for Differential Equations*). The present article addresses the problem of parameter estimation for nonlinear state-space models. It will become clear in this article that the algorithms employed for the identification of state-space models are quite complex compared to some of the black-box input-output models discussed in other articles within this section of EOLSS.

Therefore, the algorithms should only be employed if it is really necessary to obtain estimates of parameters that correspond to meaningful parameters in the original differential equations, i.e., if it is necessary to work with a so-called gray-box. If this is the case, black-box models employed for two reasons. First, the gray-box parameters cannot be recovered form the black-box. Second, most black-box parameter estimation methods require discrete-time models, and for nonlinear models it is not possible to obtain exact discrete-time equivalents. In any case, the relationship between the parameters of the discrete-time model and the continuous-time model will be quite complex.

The article is organized as follows. A few mathematical preliminaries are introduced in Section 2. A general approach to the parameter estimation problem, the recursive prediction-error approach, is reviewed in Section 3 (the prediction-error method is discussed in greater detail in *Prediction error method)*. It is shown that a prediction-error algorithm encompasses three parts: a predictor, an optimization algorithm, and a sensitivity model. The most popular optimization algorithm for parameter estimation, the recursive Gauss-Newton method, is presented in Section 3. Section 4 reviews the concept of state-space modeling and introduces state-estimation algorithms for nonlinear state-space models. These state estimators can easily be modified to also estimate unknown system parameters through the state-augmentation approach, which is

discussed in Subsection 5.1. They can also be used as predictors in a prediction-error method. In this case, sensitivity models are required, which are derived in Subsection 5.2. Some remarks regarding the relationship between these two methods and possible simplifications are given in Subsection 5.3. The conclusion follows in Section 6.

2. Mathematical Preliminaries

In this article, derivative operations are applied to matrix-vector equations. This section, therefore, clarifies what is understood by these derivative operations. Furthermore, auxiliary quantities that arise in these derivations are defined.

The *nr* by *ms* derivative of the *n* by *m* function $A(M)$ with respect to its *r* by *s* matrix argument $\boldsymbol{M}$ is formally defined as the Kronecker product of the *r* by *s* "matrix derivative operator" and the matrix function $\boldsymbol{A}$, that is,

$$\frac{\mathrm{d}A(\boldsymbol{M})}{\mathrm{d}\boldsymbol{M}} := \frac{\mathrm{d}}{\mathrm{d}\boldsymbol{M}} \otimes A(\boldsymbol{M}) = \begin{bmatrix} \frac{\mathrm{d}}{\mathrm{dM}_{11}} & \cdots & \frac{\mathrm{d}}{\mathrm{dM}_{1s}} \\ \vdots & \ddots & \vdots \\ \frac{\mathrm{d}}{\mathrm{dM}_{r1}} & \cdots & \frac{\mathrm{d}}{\mathrm{dM}_{rs}} \end{bmatrix} \otimes A(\boldsymbol{M}) \qquad (2.1)$$

$$= \begin{bmatrix} \frac{\mathrm{d}A(\boldsymbol{M})}{\mathrm{dM}_{11}} & \cdots & \frac{\mathrm{d}A(\boldsymbol{M})}{\mathrm{dM}_{1s}} \\ \vdots & \ddots & \vdots \\ \frac{\mathrm{d}A(\boldsymbol{M})}{\mathrm{dM}_{r1}} & \cdots & \frac{\mathrm{d}A(\boldsymbol{M})}{\mathrm{dM}_{rs}} \end{bmatrix}.$$

All other derivatives follow as simplifications of this general definition. For column vectors $\boldsymbol{y}$ (or scalars *y*) the term gradient is used for the expression $\mathrm{d}\boldsymbol{y}^{\mathrm{T}}(\boldsymbol{p})/\mathrm{d}\boldsymbol{p}$.

For implicit (matrix) functions, the concept of a partial derivative is defined according to

$$\frac{\partial A(\boldsymbol{B}_1,\ldots,\boldsymbol{B}_q,\boldsymbol{M})}{\partial \boldsymbol{M}} := \left.\frac{\mathrm{d}A(\boldsymbol{B}_1(\boldsymbol{M}),\ldots,\boldsymbol{B}_q(\boldsymbol{M}),\boldsymbol{M}^*)}{\mathrm{d}\boldsymbol{M}^*}\right|_{\mathrm{M}^*=\mathrm{M}}. \qquad (2.2)$$

The total derivative is then obtained from partial derivatives through the (matrix) chain rule as

$$\frac{\mathrm{d}\boldsymbol{A}^{(\mathrm{n}\times\mathrm{m})}\left(\boldsymbol{B}_1,\ldots,\boldsymbol{B}_\mathrm{q},\boldsymbol{M}\right)}{\mathrm{d}\boldsymbol{M}^{(\mathrm{r}\times\mathrm{s})}}$$
$$=\sum_{\mathrm{k}=1}^{\mathrm{q}}\left(\mathbf{I}_\mathrm{r}\otimes\frac{\partial\boldsymbol{A}\left(\boldsymbol{B}_1,\ldots,\boldsymbol{B}_\mathrm{q},\boldsymbol{M}\right)}{\partial\ \mathrm{row}\ \boldsymbol{B}_\mathrm{k}}\right)\left(\frac{\partial\ \mathrm{col}\left(\boldsymbol{B}_\mathrm{k}^\mathrm{T}\right)}{\partial\boldsymbol{M}}\otimes\mathbf{I}_\mathrm{m}\right)+\frac{\partial\boldsymbol{A}\left(\boldsymbol{B}_1,\ldots,\boldsymbol{B}_\mathrm{q},\boldsymbol{M}\right)}{\partial\boldsymbol{M}} \tag{2.3}$$

where two stacking operations are used: the "col" operator stacks a matrix columnwise into a column vector, and the "row" operator stacks a matrix rowwise into a row vector. The Kronecker permutation matrix is given as

$$\mathbf{U}_{\mathrm{kl}}:=\sum_{\mathrm{i}=1}^{\mathrm{k}}\sum_{\mathrm{j}=1}^{\mathrm{l}}\mathbf{E}_{\mathrm{ij}}^{(\mathrm{k}\times\mathrm{l})}\otimes\mathbf{E}_{\mathrm{ji}}^{(\mathrm{l}\times\mathrm{k})}, \tag{2.4}$$

where stands for a *k* by *l* matrix with a one in the *i, j* position and zeroes everywhere else. $\mathbf{E}_{\mathrm{ij}}^{(\mathrm{k}\times\mathrm{l})}$

3. The Prediction-Error Approach to Parameter Estimation

The prediction-error approach is a very general concept for the parameter estimation problem. A large number of time-domain parameter estimation methods can be interpreted as prediction-error algorithms. Even though the prediction-error method is discussed in detail elsewhere in this encyclopedia (see *Prediction error method*), its basic principles are reviewed here.

The prediction-error approach starts with the formulation of a predictor for a dynamical model. A predictor is a dynamical model that generates a prediction of the future output of a system from the knowledge of past inputs and outputs (and, in some case, statistical properties of other signals acting upon the system). Depending on the model structure, i.e., the mathematical equations used to represent dynamic behavior of the model, different predictors arise. The prediction of the system output *y* at time is often written as $\mathrm{t_k}\hat{\mathrm{y}}\left(\mathrm{t_k}\left|\mathrm{t_{k-1}}\right.\right)$, where " stresses the fact that it is an estimate and the "^" $\mathrm{t_k}\left|\mathrm{t_{k-1}}\right.$" notation indicates that it is an estimate of the output at time using all information available up to and including time $\mathrm{t_k t_{k-1}}$. An alternative notation, which is used in this article, is , where denotes that the estimate has been computed from knowledge of all values up to *but not including* . In other words, the prediction for the output at time has been computed without knowledge of the actual true output (which is exactly what makes it a prediction). The terms at $\hat{\mathrm{y}}\left(\mathrm{t_k^-}\right),\ \mathrm{t_k^-t_k t_k t^-}$ are sometimes referred to as a priori estimates, whereas the quantities at t^+ are called a posteriori values. The superscript is dropped when it is clear that "–" i$\hat{\mathrm{y}}\left(\mathrm{t_k}\right)$ is a prediction. If the model for a dynamical system is known, a predictor can be constructed. The predictor then also

depends on the parameters of the system. If the system parameters are unknown, a predictor can be constructed in which these system parameters are adjustable values. The prediction-error method consists of using such a parameterized predictor and adjusting the parameters until the predictor is "optimal," i.e., it gives the "best" prediction of future system outputs.

In order to judge this optimality, a cost functional has to be introduced. In most cases, the least-squares cost functional, i.e., a weighted sum of squared prediction errors is used, which is given as

$$\mathrm{J_{LS}}\left(\mathrm{t_k}, \boldsymbol{p}\right) = \frac{1}{2}\sum_{\mathrm{i=1}}^{\mathrm{k}} \boldsymbol{e}^{\mathrm{T}}\left(\mathrm{t_i}, \boldsymbol{p}\right) A^{-1}\left(\mathrm{t_i}\right) \boldsymbol{e}\left(\mathrm{t_i}, \boldsymbol{p}\right), \tag{3.1}$$

where $\boldsymbol{e}$ is the prediction error, i.e., the difference between the true output and its prediction

$$\boldsymbol{e}\left(\mathrm{t_i}, \boldsymbol{p}\right) = \boldsymbol{y}\left(\mathrm{t_i}\right) - \hat{\boldsymbol{y}}\left(\mathrm{t_i}, \boldsymbol{p}\right) \tag{3.2}$$

and $\boldsymbol{A}$ is a weighting matrix. Often, $\boldsymbol{A}$ is the covariance matrix of the prediction errors, which is sometimes also provided by the predictor. In equations (3.1) and (3.2), the dependence on the system parameter vector $\boldsymbol{p}$ has been included. Cost functionals other than the weighted sum of squared errors (or the log-likelihood function given below) are also possible and are sometime employed to reduce the sensitivity to outliers. Often, errors that fall outside specified boundaries, i.e., large errors (outliers) are weighted linearly instead of quadratic, or constant costs are associated to these errors. These concepts stem from robust statistics. Modifying the cost functional does, however, complicate the development of the optimization algorithm.

The estimation of the unknown parameter vector $\boldsymbol{p}$ can be carried out by minimizing the cost functional J. This can be done using any optimization method, but since the cost functional is quadratic, the Gauss-Newton method is a popular choice. This requires the gradient of the cost functional, which is given as

$$\frac{\mathrm{d}\ \mathrm{J_{LS}}\left(\mathrm{t_k}, \boldsymbol{p}\right)}{\mathrm{d}\boldsymbol{p}} = \sum_{\mathrm{i=1}}^{\mathrm{k}} \frac{\mathrm{d}\boldsymbol{e}^{\mathrm{T}}\left(\mathrm{t_i}, \boldsymbol{p}\right)}{\mathrm{d}\boldsymbol{p}} A^{-1}\left(\mathrm{t_i}\right) \boldsymbol{e}\left(\mathrm{t_i}, \boldsymbol{p}\right), \tag{3.3}$$

or, substituting equation (3.2), as

$$\frac{\mathrm{d}\ \mathrm{J_{LS}}\left(\mathrm{t_k}, \boldsymbol{p}\right)}{\mathrm{d}\boldsymbol{p}} = -\sum_{\mathrm{i=1}}^{\mathrm{k}} \frac{\mathrm{d}\hat{\boldsymbol{y}}^{\mathrm{T}}\left(\mathrm{t_i}, \boldsymbol{p}\right)}{\mathrm{d}\boldsymbol{p}} A^{-1}\left(\mathrm{t_i}\right) \boldsymbol{e}\left(\mathrm{t_i}, \boldsymbol{p}\right). \tag{3.4}$$

It is thus clear that the prediction-error method requires both the prediction and the gradient of the prediction. This is required to adjust the parameter vector in the direction of the negative gradient, i.e., in a direction that reduces the prediction errors. The algorithm that gives this gradient is called the *sensitivity model* of the predictor, since it

describes the changes in the prediction for infinitesimally small parameter changes.

In the case of offline identification, the parameter estimation would be carried out in a two-step procedure: (1) with the current parameter estimates, the predictions (and the prediction errors) are generated and the gradient of the cost functional is computed (this is a simulation step), and (2) an improved parameter estimate is obtained through an optimization method (optimization step). These steps are repeated until the parameter estimates have converged.

Often, however, the parameters are estimated in real time, i.e., during the operation of the system (*online identification*, see *Recursive algorithms*). In this case, the parameter estimate is corrected after each new measurement becomes available, i.e., in a recursive fashion. Usually, a recursive algorithm is derived from the non-recursive variant under the assumption that the previous parameter estimate was optimal, i.e., made the gradient of the cost functional zero. The gradient is thus approximated as

$$\frac{\mathrm{d}\, \mathrm{J}_{\mathrm{LS}}(\mathrm{t}_\mathrm{k}, \boldsymbol{p})}{\mathrm{d}\boldsymbol{p}} \approx -\frac{\mathrm{d}\hat{\boldsymbol{y}}^{\mathrm{T}}(\mathrm{t}_\mathrm{k}, \boldsymbol{p})}{\mathrm{d}\boldsymbol{p}} A^{-1} \boldsymbol{e}(\mathrm{t}_\mathrm{k}, \boldsymbol{p}). \tag{3.5}$$

The most widely used recursive estimation algorithm, the recursive Gauss-Newton algorithm for a least-squares cost functional, is given in Table 1. As mentioned above, this algorithm requires the prediction error and the gradient of the prediction as "external inputs". The former is generated by the predictor, whereas the latter is obtained from the sensitivity model.

Instead of the least-squares cost functional, a maximum-likelihood cost functional can also be employed. Based on the assumption that the prediction errors are normally distributed (which, at best, holds approximately for nonlinear systems), the negative log-likelihood function is given as

$$\mathrm{J}_{\mathrm{ML}}(\mathrm{t}_\mathrm{k}, \boldsymbol{p}) = \frac{1}{2}\sum_{\mathrm{i}=1}^{\mathrm{k}} \left(\boldsymbol{e}^{\mathrm{T}}(\mathrm{t}_\mathrm{i}, \boldsymbol{p}) A^{-1}(\mathrm{t}_\mathrm{i}, \boldsymbol{p}) \boldsymbol{e}(\mathrm{t}_\mathrm{i}, \boldsymbol{p}) + \ln \det A(\mathrm{t}_\mathrm{i}, \boldsymbol{p}) \right). \tag{3.6}$$

Here, both the prediction errors $\boldsymbol{e}$ and their covariance matrix $\boldsymbol{A}$ depend on the parameter vector. The gradient of this cost functional can be computed but becomes considerably more complex than the gradient of the least-squares cost functional. For this reason, the least-squares cost functional is usually preferred.

Weighting matrix	$\boldsymbol{S}(\mathrm{t}_{\mathrm{k+1}}) = A(\mathrm{t}_{\mathrm{k+1}}) + \frac{\mathrm{d}\hat{\boldsymbol{y}}(\mathrm{t}_{\mathrm{k+1}}, \hat{\boldsymbol{p}}(\mathrm{t}_\mathrm{k}))}{\mathrm{d}\hat{\boldsymbol{p}}^{\mathrm{T}}(\mathrm{t}_\mathrm{k})} \boldsymbol{P}(\mathrm{t}_\mathrm{k}) \frac{\mathrm{d}\hat{\boldsymbol{y}}^{\mathrm{T}}(\mathrm{t}_{\mathrm{k+1}}, \hat{\boldsymbol{p}}(\mathrm{t}_\mathrm{k}))}{\mathrm{d}\hat{\boldsymbol{p}}(\mathrm{t}_\mathrm{k})}$	(3.7)
Gain matrix	$\boldsymbol{L}(\mathrm{t}_{\mathrm{k+1}}) = \boldsymbol{P}(\mathrm{t}_\mathrm{k}) \frac{\mathrm{d}\hat{\boldsymbol{y}}^{\mathrm{T}}(\mathrm{t}_{\mathrm{k+1}}, \hat{\boldsymbol{p}}(\mathrm{t}_\mathrm{k}))}{\mathrm{d}\hat{\boldsymbol{p}}(\mathrm{t}_\mathrm{k})} \boldsymbol{S}^{-1}(\mathrm{t}_{\mathrm{k+1}})$	(3.8)

Parameter update	$\hat{\boldsymbol{p}}(t_{k+1}) = \hat{\boldsymbol{p}}(t_k) + \boldsymbol{L}(t_{k+1})\boldsymbol{e}(t_{k+1})$	(3.9)
Covariance update	$\boldsymbol{P}(t_{k+1}) = \frac{1}{\lambda}\left(\boldsymbol{P}(t_k) - \boldsymbol{L}(t_{k+1})\boldsymbol{S}(t_{k+1})\boldsymbol{L}^{\mathrm{T}}(t_{k+1})\right)$	(3.10)
Initial values	Initial parameter estimate $\hat{\boldsymbol{p}}(0)$; often set to zero. Initial covariance matrix; usual choice $P(0) = \alpha \mathbf{I},\ \alpha >> 1$.	

Table 1 Recursive Gauss-Newton algorithm for a quadratic cost functional

The prediction-error methods thus consists of a predictor (that gives the output prediction and possibly the prediction error covariance matrix $\boldsymbol{A}$), a sensitivity model (that gives the gradient of the prediction error and possibly the gradient of its error covariance matrix $\boldsymbol{A}$), and an optimization algorithm (that governs how the parameter estimates are corrected, usually the Gauss-Newton algorithm of Table 1). The prediction-error method is thus, at least in principle, applicable to any system for which a mathematical model is available and for which a predictor (and a sensitivity model) can be formulated. This applies both to linear and nonlinear models as well as to continuous-time and discrete-time models. $\hat{y}$

-
-
-

Bibliography

Ahmed, M. S. 1994. An innovation representation for nonlinear systems with application to parameter and state estimation. *Automatica* 30:1967-74. [In this article, the author proposes the use of an innovation representation (corresponding to a constant-gain filter) for parameter and state estimation. The algorithm derived in this paper is essentially an adaptive version of Safonov and Athans' (1978) constant-gain filter. This approach has also been suggested by Ljungquist and Balchen (1993, 1994).]

Bohn, C. 2000. *Recursive parameter estimation for nonlinear continuous-time systems.* Berlin: Pro Business. [A doctoral dissertation that addresses the problem of parameter estimation. Contains a detailed formulation of adaptive versions of the extended Kalman filter and higher-order filters. The focus is on the sensitivity-model based approach which was first introduced by Mehra and Tyler (1974).]

Bohn, C. and H. Unbehauen. 2000. The application of matrix differential calculus for the derivation of simplified expressions in approximate nonlinear filtering algorithms. *Automatica* 36:1553-60. [This article presents a unified framework for higher-order state estimators. Matrix differential calculus is used to derive expressions that are much more compact than the ones previously available in the literature. The results of matrix differential calculus are taken from Weinmann (1994).]

Bohn, C. and H. Unbehauen. 2001. Sensitivity models for nonlinear filters with application to recursive parameter estimation for nonlinear state-space models. *IEE Proceedings – Control Theory and Applications* XX:xxx-xx. (to be updated in the final version) [Sensitivity models are derived for the general filtering algorithm presented by Bohn and Unbehauen (2000). The paper also contains an application of an adaptive extended Kalman filter.]

Chu, Q. P., J. A. Mulder and P. T. L. M. van Woerkom. 1996. Modified recursive maximum likelihood adaptive filter for nonlinear aircraft flight-path reconstruction. *Journal of Guidance, Control, and Dynamics* 19:1285-95. [An adaptive extended Kalman filter is used for state and parameter estimation. The difference to the sensitivity-model based approach (Mehra and Tyler 1973; Jakoby and Pandit 1987; Bohn 2000; Bohn and Unbehauen 2001) is that instead of a sensitivity model the gradient of the cost functional is determined through finite differences.]

Gavel, D. T. and S. G. Azevedo. 1982. Identification of continuous time systems – An application of Ljung's corrected extended Kalman filter. *Preprints of the 6th IFAC Symposium on Identification and System Parameter Estimation.* Arlington, VA. 1191-95. [A version of the recursive prediction-error algorithm is developed for linear continuous-time state-space models.]

Jakoby, W. and M. Pandit. 1987. A prediction-error method for recursive identification of nonlinear systems. *Automatica* 23:491-96. [An adaptive extended Kalman filter is constructed for the estimation of parameters of a nonlinear discrete-time state-space model. Contrary to the constant-gain approaches (Ljungquist and Balchen 1993, 1994; Ahmed 1994), a time-varying filter gain is used. The noise covariance matrices are assumed to be known.]

Ljung, L. 1979. Asymptotic behavior of the extended Kalman filter as a parameter estimator for linear systems. *IEEE Transactions on Automatic Control* 24:36-50. [The behavior of the extended Kalman filter is analyzed. It is shown that "if the noise characteristics of the model is [sic] chosen and hoc, …the system parameters will in general be biased…. The cause of this bias … comes from incorrect noise assumptions associated with the model." A modification to the extended Kalman filter (that essentially changes it to a prediction-error method) is proposed to avoid this bias.]

Ljungquist, D. and J. G. Balchen. 1993. Recursive prediction-error methods for online estimation in nonlinear state-space models. *Proceedings of the 32nd IEEE Conference on Decision and Control.* San Antonio. 714-19; also 1994. Recursive prediction-error methods for online estimation in nonlinear state-space models. *Modeling, Identification and Control* 15:109-21. [In these articles, a constant-gain extended Kalman filter is used for the parameter estimation of nonlinear discrete-time state-space models. This is essentially the same approach that is used by Ahmed (1994).]

Zhou, W. W. and M. Blanke. 1986. Identification of a class of nonlinear state-space models using RPE techniques. *Proceedings of the 25th IEEE Conference on Decision and Control.* Athens, Greece. 1637-42.; 1989. Identification of a class of nonlinear state-space models using RPE techniques. *IEEE Transactions on Automatic Control* 34:312-16. [An adaptive filter of the first-order bias-corrected type is proposed as a parameter estimator for nonlinear state-space models. Besides the fact that a first-order bias-corrected filter is used, this approach is similar to the constant-gain approaches of Ljungquist and Balchen (1993, 1994) and Ahmed (1994). Interestingly, a constant filter gain is used even though the algorithm would have allowed for a time-varying filter gain.]

Biographical Sketch

Christian Bohn was born in 1969 in Hamburg, Germany. After receiving the *Diplomingenieur* degree in electrical engineering from the Technical University of Braunschweig, Germany, in 1994, he worked as a research assistant at the Control Engineering Laboratory at the Ruhr-University Bochum, Germany. He received the doctorate in electrical engineering for his thesis on parameter estimation for nonlinear continuous-time state-space models from the Ruhr-University Bochum in 2000. Since 2000, Dr. Bohn has been working in the Strategic Technology Divison of Continental AG in Hanover, Germany, a

manufacturer of tires and rubber products and major supplier of automotive electronics. His work is concerned with developing active and adaptive vibration control systems, and his technical interest includes control strategies for automotive applications, e.g., active and semi-active suspension systems, and the application of estimation and signal processing techniques to obtain information about relevant vehicle states and parameters.