Robust State Estimation of Complex Systems

Jan Hauth, Patrick Lang, and Andreas Wirsen

1 Challenges for Industry

The complexity of many technical applications and production processes is continuously increasing, due to growth in the technological possibilities of the produced goods. For biological processes, which are inherently very complex to begin with, complexity has quite different facets, which find their expression, for example, in the linkage of numerous subprocesses, in nonlinear system dynamics, and in combinations of the two. Moreover, in many cases, the descriptions of the processes and systems are plagued with significant uncertainties. In technical systems, these result from uncertainties regarding the parameters of integrated components and their time-dependent variability during operations, as well as disturbances originating in the external process environment. In biological systems, the natural fluctuations and variability typical of living systems mean that these uncertainties often play an even more important role. Therefore, when developing new medical compounds or devices, or when designing and controlling bioreactors, for example, it is imperative to take them into account.

Despite the increasing complexity of and unavoidable uncertainties in technically relevant systems, the requirements for ensuring a variety of process and system characteristics are also becoming increasingly stringent. Some of these characteristics are:

Product Quality The complexity and associated dynamic effects make continuous and complete monitoring of critical system parameters imperative, in order to be able to react quickly with suitable control measures to changes in system behavior and thus guarantee

J. Hauth · P. Lang (✉) · A. Wirsen
Fraunhofer-Institut für Techno- und Wirtschaftsmathematik, Fraunhofer-Platz 1,
67663 Kaiserslautern, Germany
e-mail: patrick.lang@itwm.fraunhofer.de

H. Neunzert, D. Prätzel-Wolters (eds.), *Currents in Industrial Mathematics*,
DOI 10.1007/978-3-662-48258-2_9

consistent product quality. The use of automated control methods also requires access to such system information.

System Reliability When there is a complex interplay of many components and this interplay is very dependent on the system's various operating modes, it is often impossible to give a meaningful *a priori* estimate of the lifetime of the individual components. In order to (economically) ensure continuous functioning of the total system, permanent monitoring of critical system components is therefore a sensible alternative. The replacement of components whose operating characteristics are deteriorating can then be scheduled intelligently. Such a predictive maintenance approach allows down time and maintenance costs to be minimized.

Ensuring both qualities requires access to critical information about dynamic system events. The most straightforward way to obtain the needed system and process information consists of directly measuring the crucial states and parameters using suitable sensor technology. However, directly monitoring all relevant system quantities is usually impossible, due to technical limitations in the available sensor technology and the limited number of suitable measurement sites. Moreover, due to the number of sensors that would be needed, direct measurements of all quantities would often be too expensive. Model-based state estimation offers one way around this problem. Here, system simulation on the basis of an existing system model is combined incrementally with each piece of available measurement information to derive the best possible estimate of the system's true state. The system model allows one to calculate the system quantities and parameters that are actually relevant, on the basis of simple functional inter-relationships, and thus represents a virtual sensor technology.

The characteristics of modern technical systems result in a variety of challenges for these state estimators.

Real-Time Capability For many applications requiring interventions to control and regulate the system, it is essential to deliver the needed system state estimates in what amounts to real time. Particularly for highly dynamic processes, this is a true challenge that requires the combined use of dimensionally restricted system models and correspondingly powerful (i.e., fast) hardware. Here, one must also ensure that the sensors being used are up to the dynamic challenges of the process in question.

Robustness Because state estimators often deliver the basis information for associated system control algorithms, a certain level of robustness must be guaranteed in the face of changes in system specifications. This applies both to short-term variations in the parameter values of particular components due to changing ambient conditions in the process environment and also to permanent variations in parameter values due to aging processes. Beyond this, most system parameters are initially specified with only limited exactness by the equipment suppliers. Nonetheless, one is interested in having the most exact information possible about the true dynamics of the system. One also wants to guarantee

the robustness of the state estimations in the face of disturbances arising from outside the system—which are often only partially known. The reliability of the sensor technology is another important consideration; occasional faulty measurements or even the complete loss of a particular sensor signal cannot lead to a collapse of the overall state estimation procedure. Here, under some circumstances, redundancy concepts must be incorporated to rule out such scenarios. In this context, the problem of non-synchronized sensor technology must also be managed in an appropriate fashion.

2 Challenges for Mathematics

In many technical, medical, and biological processes, mathematical state estimation is an important tool for determining process states that are hidden or not directly measureable, based on the synergetic combination of information from a system simulation and real measurements of various system quantities. When preparing state estimators, the following challenges present themselves:

System Model On the basis of a suitably defined system state that includes all information needed for the further dynamic development of the system, one uses the existing technical and/or biological understanding of the system, along with the relevant, available process data, to prepare a model that accurately predicts the future development of the system state. In many cases, this modeling leads to a state dynamic in the form of an ordinary differential equation system or a differential algebraic system. When one uses a purely knowledge-driven modeling approach, the result is a so-called white box model. As the proportion of data used in the modeling approach increases, the model is then referred to as a gray box and, ultimately, a black box model. The model of the state dynamic is supplemented by equations that permit calculation of the system quantities that are actually to be monitored on the basis of the system state. The relationships to the measured system quantities must also be captured appropriately. In particular, when designing a state estimator, the information content of the possible variants can be appraised and compared on the basis of the measurements. This supports the selection of the best possible measurement configuration.

Depending on the characteristics of the underlying application, one must ensure that the complexity of the model being developed is compatible with the time available for executing the state estimation. Highly dynamic applications, for example, demand state estimation in close to real time. If the dimension of the resulting model is too large, model reduction techniques can be used to generate an error-controlled approximation by means of a smaller system model. A variety of model reduction methods is available, depending on the type of state-space model being used.

Any special requirements for preparing the model and the associated state estimator result primarily from significant nonlinearities in the dynamic behavior of the underlying system. When dealing with large, networked systems, one must also decide whether to use a centralized or localized design for the state estimator(s).

Uncertainties The appropriate treatment of uncertainties during the modeling process is a key concern. These can be uncertainties in one's understanding of the physical or biological relationships that dominate the process or system. Or they can be uncertainties about parameters used in the model, which may be known with only limited accuracy and are often subject to short-term fluctuations caused by the process environment. Aging processes, such as wear and corrosion, also cause parameter values to drift over longer time periods. If important parameters are not known at all, a suitable parameter estimation process can be incorporated into an extended state estimation problem.

Along with the uncertainties in parameters, the unavoidable errors associated with measurements also play an important role and must be treated appropriately. The technology of the sensors being used and the associated signal processing chain offer clues about how to model the system. Analyzing a sufficient number of measurements is of central importance for establishing appropriate distribution functions. The characteristics of the individual sensors, as well as knowledge about the time-points of the measurements and the relationships between these time-points for the different sensors, are both of central importance for designing the state estimator. Uncertainties in these quantities must also be suitably accounted for in the model.

Furthermore, there are almost always external effects or phenomena impacting the system that cannot be explicitly accounted for in the model due to a lack of detailed information. Here, rough disturbance models are the best one can do to treat these impacts. Depending on whether deterministic or stochastic phenomena predominate in the model, the state estimation problem tends to also be viewed in either a deterministic or stochastic light.

Performance Criteria The appropriate specification of the performance criteria depends on the desired characteristics of the state estimator being designed. Here, of course, the expected estimation errors play a central role, and special emphases result from the specifically chosen error norms and signal classes, for which the appropriate optimization is carried out. In many cases, the estimation errors are not weighted uniformly, since it proves advantageous to weight by specifying time horizons.

On the basis of the prepared model and the selected performance criterion, the weighting matrices for the combination of model simulation and measurement value can be defined explicitly in advance by solving the appropriate Riccati equations. This applies especially in the context of linear, time-invariant system models. Important, well-known variants here are first the Kalman filter and then the H_∞ filter. For nonlinear systems, linearization around certain operating points makes it possible to apply the linear concepts to a certain degree within the framework of the extended Kalman filter. Here, however, no optimality characteristics can be shown and the derived confidence intervals are not valid. Although, in the general nonlinear case, an optimal state estimator can indeed be specified in theory, direct calculation—as in the case of the Kalman filter—is not possible. One must therefore rely on approximations. The particle filter accomplishes one such approximate calculation with the help of a sequential Monte Carlo approach. In principle, this amounts to simulating in parallel numerous possible system trajectories across

an appropriate proposal distribution and weighting them by the measurement values with an importance sampling approach. An additional resampling step prevents degeneration of this method over time. The filter distribution is then given at every point in time as an empirical distribution (weighted samples), which is used to calculate approximation values for further quantities of interest, such as averages, variances, or confidence intervals. In contrast to the extended Kalman filter, this approach ensures mathematical convergence. On the other hand, the efficiency of the implemented algorithm depends strongly on the selection of suitable proposal distributions. Finding them is a problem that must be solved specifically for the application in question when preparing the model.

3 Previous Studies on This Subject

For many years, the System Analysis, Prognosis, and Control Department has been working in various application contexts with the subject of state estimations. Here, in many cases, existing methods have been adapted to the specific applications. Extensions and brand-new solutions for special problems have also been developed, however.

Robust Observers for Elastomechanical Systems The Department's first studies of state estimation came in connection with a project to develop an observer for turbo generator sets in power plants. These turbo generator sets consist of a long shaft on which the generator and, in general, several turbines are mounted. They are vulnerable to torsional vibrations, which can be induced by disturbances in the electrical grid. These vibrations can reach considerable amplitudes due to weak system damping, and the resulting material fatigue can substantially decrease the turbo generator set's life expectancy. Thus, the need arose for a monitoring system to permanently estimate and track the system's expected remaining operating life. Because the structures surrounding the turbo generator set limit access to the actual shaft to only a few places, a state estimator was developed on the basis of torque measurements at a single shaft position. Here, the starting point for describing the system dynamic is a high-dimensional, second order, linear state-space model, where the states describe the torsion angle of the shaft sections relative to the zero position. Whereas the resulting matrices for the moments of inertia and stiffness are quite well known from the finite element model, the damping matrix is generally subject to large uncertainties. At best, one has merely rough estimates for the modal damping. These boundary conditions meant that the project focused on adapting known approaches from the field of robust state estimation, with regard, in particular, to the high dimensionality. Especially for weakly damped systems, however, the required solution of certain Riccati equations is very poorly conditioned and presents problems for traditional solution methods. On the other hand, explicit formulas for approximation solutions can be specified for special system representations resulting from modal state-space transformations. Error estimates for the approximation quality can be derived from familiar matrix inequalities.

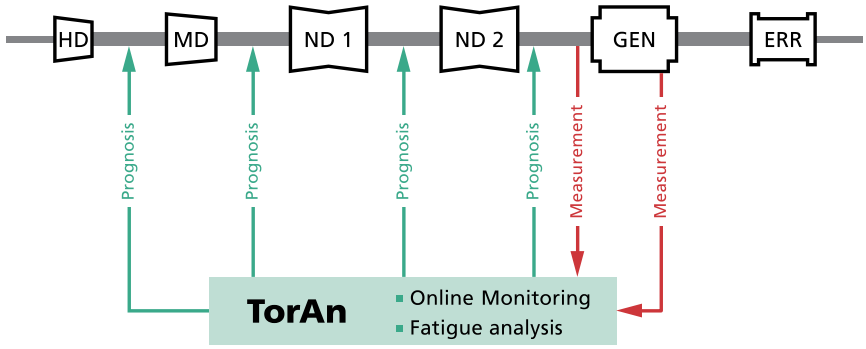

Fig. 1 Schematic drawing of an observer for power plant turbo generator sets with high-pressure turbine (HD), intermediate-pressure turbine (MD), low-pressure turbine (ND), Generator (GEN), and excitation machine (ERR)

To develop the state estimator for turbo generator shaft lines, the Department collaborated for many years on the industrial side with Siemens AG, in Mülheim, and service providers such as E.ON Anlagenservice. On the scientific side, we should mention our cooperation with the Electrical Drives and Mechatronic Chair, headed by Professor Dr. Stefan Kulig at the University of Dortmund—a cooperation that is still active today. Together with staff from Professor Kulig's department, we developed the torque monitoring and analysis system TorAn (see Fig. 1) introduced in Sect. 6. On the basis of the developed state estimators, this system delivers on-line predictions of the torsional vibration behavior of turbo generator sets at critical shaft components and determines the resulting material fatigue in the case of disturbances [5, 6]. To measure the torques needed to determine the correction term, a contact-free magnetostrictive sensor was further developed on behalf of the ITWM. In the course of further developments, TorAn was supplemented with the monitoring systems TorFat and TorStor. Unlike TorAn, however, these systems do not generate prognoses of the torsional vibrations by means of a state estimator. The focus of TorFat was to develop methods for rapid detection of highly critical torsional vibrations, such as sub-synchronous resonances. TorStor was designed to record torques within experiments and determine as precisely as possible such relevant system quantities as damping parameters or the resonant frequencies of the shaft line. To accomplish this, a filter had to be developed to allow for compensation of the periodic disturbances—the so-called run-out—resulting from the magnetostrictive measurement principle used by the contact-free torque sensor. The phenomenon of run-out and the possibilities for using a state estimator to filter these disturbances are described in Sect. 6. The torsional recording and analysis system developed at the ITWM have been adopted by power plants and large-scale industrial installations and are now in service around the world.

Controller Design for Active Vibration Damping of Elastomechanical Systems Many model-based control approaches work on the principle of state feedback; that is, the estimation of the system state is part of the control algorithm. Thus, a close relation-

ship exists between state estimation and controller design. In this regard, the expertise in state estimation of elastomechanical systems described in the previous section was further pursued in projects involving active vibration damping. Only an optimal interplay between system structure and system control can produce the best-possible damping of oscillation behavior in relation to vibrations or noise reverberation, for example. Here too, for model-based controller design, one starts with the second order differential equation systems for describing the system dynamics of elastomechanical systems. Based on either an explicit estimate of the system states—as, for example, in the context of model predictive control—or an implicit state estimate—as, for example, in the context of optimal H_2 controlling or robust H_∞ controlling—the regulating variables for the actuators are defined in relation to the selected performance goals. Here, the model parameters are adjusted to "reality" by means of the state estimations. The estimated states then form the starting point for calculating the control input needed to achieve the desired performance behavior. Thus, in comparison with such classical control approaches as PID, model-based control also permits adjustment of performance quantities that are not directly measurable, but must be optimized nonetheless.

On behalf of Volkswagen AG, a MATLAB Toolbox was developed for automated controller design of active vibration damping in the drive train of a motor car, taking due consideration of nonlinear actuator behavior [39]. In other projects, we investigated the use of new "smart actuators" with nonlinear behaviors, such as hysteresis and saturation, and developed controller concepts for compensating these effects. Active noise reduction in vehicle interiors by means of smart actuators was one of our studies in this area [7].

Particle Filters The problem of state estimation in nonlinear system models having non-Gaussian disturbance processes can be solved approximately with the help of sequential Monte Carlo methods. Worth mentioning in particular is the particle filter algorithm, which works on a set of weighted samples (particles). Parameter estimation problems can also be addressed by including parameters in the state set or by means of additional Markov Chain Monte Carlo (MCMC) approaches.

Within the Department, the methodology of particle filters was initially investigated and adapted for state estimation on the basis of hysteresis-prone, nonlinear component models from the automobile industry. Subsequently, these techniques were then used primarily in the context of state and parameter estimation in biological systems. New methodological developments took place in connection with the explicit treatment of uncertainties in the measurement time-points in the particle filter approach. It was also shown that a model predictive controller (MPC) can be realized by suitably coupling two particle filters.

In this field of endeavor, the Department has worked together for years with the System Biology Department of Professor Mats Jirstrand, from the Fraunhofer Chalmers Center in Gothenburg. Here, the particular focus of activity was the development of a Mathematica-based system biology toolbox.

4 Modeling Principles

In this section, we will present filtering theory from the standpoint of a very general stochastic approach. Stochastic state-space models form the basis for these reflections. They separate the modeling of non-observable, internal system states from those system quantities observed by means of measurements. Both are modeled using coupled stochastic processes. Here, the stochastics is used to model disturbances and uncertainties, as well as intrinsic system variability. The state process models the dynamics of the system, while the measuring process describes the measurement/observation procedure. The underlying spaces for state and measuring processes are kept very general: they are not restricted to discrete or real spaces, and mixtures are also possible. Nor are there restrictions on the time-points at which measurements may take place: multiple measurements with different sampling times can be modeled, along with uncertainties in measurement time-points. The advantage of this approach is that it makes possible a very flexible mathematical modeling; models do not have to be restricted to a particular model class in advance. The disadvantage is that drawing inferences about internal states on the basis of observed measurements on real systems becomes very difficult and can only be made in this generality using Monte Carlo approaches.

4.1 Inference in Complex Systems

Real systems always exhibit a certain variability. Whereas, in technical systems, one uses design and control mechanisms to try and keep this variance small—or at least under control—in biological systems, this is not feasible in most cases. Living cells, for example—even those of the same type and age—differ greatly from one another in their characteristics: they are different in size and shape, or they are at different developmental stages. When using such microorganisms in bioreactors to produce pharmaceutical ingredients, this variability is also ultimately transmitted to the technical systems involved. But even in non-biological technical systems, (mostly undesired) variabilities can also arise, through aging and wear, for example, or through faulty system behavior.

The appropriate mathematical tool for dealing with these uncertainties is probability theory. While it often suffices in simple (mostly technical) systems to calculate using averages, and one can therefore limit oneself to deterministic calculations (i.e., the solution of differential equations), in many naturally-arising complex systems, this is not justified. In these cases, therefore, we choose a stochastic approach right from the start.

We obtain qualitative information about technical or natural systems by observing them; we obtain quantitative information by making measurements. The measurement process itself should always be viewed independently from the actual system (see Fig. 2). The actual state of the system at any point in time is not apparent to us. In this sense, the process that describes the state of the system is hidden from our view. The measurements serve to nonetheless make indirect quantitative statements about the current state of the

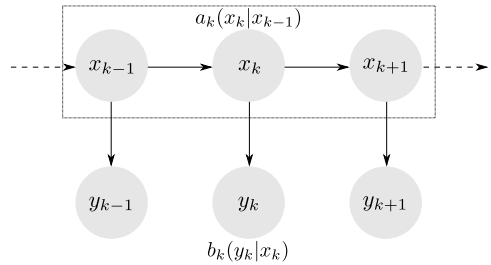

Fig. 2 A state-space model in discrete time. Here, x_k is the state vector at time k, y_k is the measurement at time k, dependent only on state x_k at the same time. The stochastic dependencies are given here by conditioned densities: $a_k(x_k \mid x_{k-1})$ describes the dependency of the current state x_k exclusively on the basis of the previous state x_{k-1} (Markov property of the state process) and $b_k(y_k \mid x_k)$, the dependency of the measurement y_k exclusively from the current state x_k. Only the measurements are observable; the states x_k themselves are hidden (non-observable)

system we are studying. Both processes—the hidden state process and also the observation process—must be viewed as having uncertainties: the state process, due to internal variabilities in the system or unobservable and/or unmodeled external disturbances; the measurement process, due to measurement errors or inaccuracies.

As a result of these multiple stochastic dependencies, the measurement results collected over time for a dynamic, changing system exhibit complicated correlations among one another, so that simple statistical evaluations of the measurements are not adequate.

The key that allows us to draw any inferences at all from noisy measurements about the internal system states lies in the stochastic dependencies of both processes. These dependencies affect, first, the time dependencies of the system states among each other and, second, the stochastic dependencies of the measurement process on the system process. Here, as well, probability theory offers a self-contained and extremely efficient tool that allows us to both model such systems and also draw inferences in a rigorous and unambiguous manner.

The Bayesian approach, in particular, which allows each quantity to be equipped with distributions, delivers here via Bayes's law a universal tool with which any inference problem subject to uncertainty can be at least theoretically solved in a transparent and simple fashion. This last trait—the simplicity of the theoretical solution—does not necessarily transfer to practical calculations. Here, one finds analytical and, thus, easily calculable solutions in only very few instances. In the case of state filtering, there are exactly two: systems with a finite number of discrete states and linear systems with Gaussian disturbances. For the latter, the solution is given by the familiar Kalman filter.

In all other cases, the calculation proves difficult. Only two developments in the second half of the 20th century—powerful computers and Monte Carlo methods—finally made it possible to execute these calculations for complex cases as well. This advance has not nearly run its course. Many algorithms, particularly in the area of state filtering or parameter estimation in dynamic systems, are new. The particle filter for state estimation in nonlinear systems, for example, is not yet 20 years old, and a promising method

for joint Bayes estimation of states and parameters—supported by convergence proofs—seems only to have recently been found in the form of the new PMCMC approach [16].

4.2 *A Posteriori* Path and Filter Distributions

The question arises as to what the existing measurement data allows one to claim, in the best case, about a system's internal states. In dynamic systems that are mathematically represented by means of state-space models, the trajectories (paths) of the internal (current) system states play an important role. Because we are considering stochastic systems, the paths are not defined deterministically, but are subject to random distributions. These random distributions are initially specified by the system model and define the possible temporal developments of the system. One sometimes speaks of the prior or *a priori* distributions of the system trajectories (paths), since these are the probability distributions that are valid *before* the measuring process begins. In systems with a large proportion of stochastic disturbances, the system's range of possibilities is typically very broad, that is, the prior probability distribution of the path is very wide. It is the task of the filter to modify the system's prior probabilities with the help of the likewise randomly disturbed measurement data, so that system paths that do not fit the data become less probable and system paths that explain the measurements satisfactorily become more probable. These probability distributions, which describe the system states and/or trajectories *after* measurement data collection, are then referred to as posterior or *a posteriori* distributions. These are conditional probabilities that are dependent on the measurements. The selection of both the prior distributions for the system's state trajectories and the distributions for the measurements as functions of the state trajectories belongs to the model design process. Once these distributions have been defined, the posterior distribution of the states is—from a probability theory standpoint—the best possible information that one can obtain about the system's development on the basis of the measurement data. The mathematical result delivering the posterior distribution is Bayes's Theorem.

Thus, we are actually interested in the posterior distributions of the state trajectories. Although these path distributions are often very difficult to treat, under certain circumstances, it is not even necessary to consider complete paths. This is so when the current system states at each point in time already contain all the information about the future development of the system. In this instance, it is no longer necessary to consider past states (or entire past paths), since this would deliver no additional information. One then describes the system as having the Markov property. In systems having the Markov property, it therefore suffices to consider the current distributions of the states over time, instead of the path distributions. If one now calculates the corresponding posterior distributions of the current system states at a given time t, taking only into consideration those measurements made before or at time t, then one obtains exactly the filter distributions. The filter distribution at each time t is therefore the posterior distribution of the states at time t, given the measurements up to time t. It turns out that the filter distributions for consecutive

measurement time-points can be calculated recursively. At this level of generality, the state filtering can then be performed with sequential Monte Carlo methods, the most important of which is the particle filter, a combination of importance sampling and re-sampling over time. Here, theoretical convergence results are available. The filter distribution serves as the foundation for further important applications, such as parameter estimation and control.

4.3 Parameter Estimation with the Maximum Likelihood Approach

Parameter estimation in stochastic state-space systems is an extremely difficult problem. In cases where the system dynamics can simply be modeled using ordinary differential equations—that is, without stochastic noise in the states and/or correlated noise in the measurements—the problem is often considered as a deterministic optimization problem on the basis of a Maximum Likelihood (ML) approach. An overview of these approaches having a focus on biological applications can be found in [38] and [22]; see [17] also, where other aspects are considered, such as identifiability. A generalization of the ML approach via introduction of more flexible cost functions is offered by the Prediction Error Estimation methods [20]. In contrast, if one takes as a basis a model that assumes additional stochastic disturbances in the state dynamics, then one arrives at an optimization problem with constraints, where these constraints are given by stochastic differential equations (SDEs). In this case, the internal system states can no longer be directly observed or calculated and must therefore be estimated, along with the parameters, on the basis of existing measurement data. Toward this end, the method used for parameter estimation must be supplemented by the appropriate state filter methods. An overview of ML estimation for this case is found in [40]. If the underlying SDEs are linear, then the Kalman filter delivers an exact solution of the filter distribution. If the SDEs are nonlinear, then one typically relies on linearized versions of the Kalman filter, such as the Extended Kalman Filter (EKF) or the Unscented Kalman Filter (UKF), in order to obtain approximations of average values and co-variances of the filter distribution over time. All these approximations based on the Kalman filter have a crucial disadvantage, however: they approximate the filter distribution over time, in the best case, only with a Gaussian normal distribution. Therefore, they cannot properly approximate multi-modal distributions (that is, those with multiple local maxima in the probability density) or skewed distributions. Better approximations are given by simulation-based methods (sequential Monte Carlo, SMC), to which the particle filter also belongs. Good convergence results have been achieved here [24]. However, these algorithms still exhibit significant problems when applied to simultaneous estimation of dynamic states and fixed parameters ([15, 31, 45]; see [16] also).

4.4 Parameter Estimation with the Bayesian Approach

The Bayesian context differs from the "classical" ML approach in that a prior probability distribution is assigned to the parameter vector. The parameters are thus treated as random variables, just like the state and measurement variables. The prior distribution reflects knowledge about the parameters before considering the measurements. The estimation problem thus consists of determining the posterior distribution, that is, the probability distribution that describes knowledge about the parameters after incorporation of the measurement results (observations). At least theoretically, this can be calculated with the aid of Bayes's Theorem, if the prior distribution and the observations are given. For nontrivial problems, however, this requires calculating high-dimensional integrals, for which there are no analytical solutions. In practice, then, calculation presents great difficulties. Simulation-based methods once again offer a remedy—in this case, Markov Chain Monte Carlo (MCMC) methods. They represent a generally applicable tool for approximating posterior distributions. Here as well, however, problems arise with the joint estimation of dynamic states and fixed parameters. For example, the design of good distribution proposals for standard MCMC methods, such as the Metropolis-Hastings sampler, is practically impossible. Therefore, these methods cannot be used profitably for estimations in stochastic state-space models.

It would therefore be desirable to find an approach that combines the dynamic SMC method with the static MCMC method—with SMC as a suitable tool for estimating the states, and MCMC as a suitable tool for estimating the posterior distribution of the parameters. One would then have at one's disposal a general tool for estimating parameters in stochastic state-space models. For a long time, this combination approach remained unattainable, since calculating the acceptance probabilities of the MCMC methods presupposes knowledge of the density function of the particle distributions. Andrieu et al. [16] were the first to succeed, when they used an auxiliary variable approach (extension of the state-space by the ancestral path distributions) to show that knowledge of the approximate data likelihoods alone suffices; the particle filter delivers this knowledge for free, so to speak. Their promising approach, known as Particle Markov Chain Monte Carlo (PMCMC), is generally applicable and is backed up by good convergence results.

An alternative to PMCMC remains an established approach in which the fixed states are provided with an artificial dynamic by allowing the parameters to change their values slightly over time in a stochastic way. Thus, in the Bayesian context, states and parameters are placed on the same level conceptually: parameters can be added to the system states (augmented states) and estimated jointly via filter methods. In order for this approach to work well, however, it is important for the variances of the artificial parameter dynamics to be well chosen, a task that is quite difficult in many instances.

4.5 Nonlinear Mixed Effects Models

Estimating in Nonlinear Mixed Effects Models (NLME) requires estimating both global and individual parameters. With classical Maximum Likelihood estimations, there is a large conceptual difference between these two types of parameters. Whereas the individual parameters are conceived as random variables that are appropriately outfitted with probability distributions, the global parameters remain pure constants whose "true" values are simply unknown. If the equations underlying the model are nonlinear, this leads to likelihood functions that can no longer be directly evaluated. In this case, one must work with approximations. The tool NONMEM [18] has become established for certain application areas in cases where the state dynamics are modeled using deterministic ODE. In [41], in contrast, for NLME models based on stochastic differential equations (SDE), an estimation algorithm is proposed that relies on the Extended Kalman Filter (EKF) for filtering SDE. This approach was added to NONMEM [46]. In [19], a comparison was performed between ODE-based and SDE-based methods for parameter estimation in NLME models. The result here was that the estimations of the variabilities between the individual parameters generally assume smaller values for the SDE model. Donnet and Samson [27] propose combining a stochastic version of the Expectation Maximization Algorithm (for estimating global parameters) with MCMC methods (for estimating the states and the individual parameters). However, because MCMC methods exhibit problems regarding the use of joint state and parameter estimation (as mentioned above), the MCMC approach was replaced in [28] by the more suitable PMCMC method of Andrieu et al. [16].

In contrast to the Maximum Likelihood approach, in the Bayesian context, the global parameters are also supplied with (prior) probabilities, and the conceptual differences between global and individual parameters do not exist. The Mixed Effects model can be understood here simply as a hierarchical stochastic model with independent and dependent parameters [14, 43, 44]. Simulation-based (Monte Carlo) methods can be easily adapted to this case. However, the above-mentioned difficulties and requirements for the SMC and MCMC methods (or combinations of the two) become even more pronounced due to the correspondingly larger number of states and parameters in NLME models, since the number of states and individual parameters must be multiplied by the number of individual parameters.

4.6 The State-Space Model

We now want to provide the mathematical foundations that give a concrete form to our descriptions of the previous sections, and do so in a very generalized context. We consider the state-space models in continuous time, that is, we take as given that the state process, in particular, is a continuous-time Hidden Markov process with corresponding continuous-time transition kernels.

4.6.1 The State Process

Let $(\Omega, \mathscr{A}, \mathsf{P})$ be a probability space, and for each $t \in [t_0, \infty)$ with $t_0 \in \mathbf{R}$, let $(\mathscr{X}_t, \mathscr{B}_{\mathscr{X}_t})$ be an arbitrary measurable space. Furthermore, for each $t \in [t_0, \infty)$, let $X_t : \Omega \to \mathscr{X}_t$ be a $\mathscr{A} - \mathscr{B}_{\mathscr{X}_t}$ measurable random variable, such that $X_{[t_0,\infty)} := (X_t)_{t\in[t_0,\infty)}$ is a continuous-time Markov process with general state-space

$$\mathscr{X}_{[t_0,\infty)} := \prod_{t_0 \le s} \mathscr{X}_s.$$

For each $t \in [t_0, \infty)$, $\mathscr{L}_{X_t}$ denotes the pushforward measure of P under X_t, that is, $\mathscr{L}_{X_t}(B) := \mathsf{P}(X_t^{-1}(B))$ for all $B \in \mathscr{B}_{\mathscr{X}_t}$. Moreover, $\mathscr{L}_{X_{[t_0,\infty)}}$ denotes the pushforward measure of P under $X_{[t_0,\infty)} := (X_s)_{s\in[t_0,\infty)}$ (with the corresponding product algebra). Analogously,

$$\mathscr{X}_{[t_0,t]} := \prod_{t_0 \le s \le t} \mathscr{X}_s \quad \text{for each } t \ge t_0$$

denotes the state-space restricted to the interval $[t_0, t]$, and $\mathscr{L}_{X_{[t_0,t]}}$ denotes the corresponding pushforward measure. For each s and t, with $t > s \ge t_0$, let $K_{s,t}(x_s, \mathrm{d}x_t)$ be the Markov kernel of the process $X_{[t_0,\infty)}$ from time s to time t.

An important special case for $X_{[t_0,\infty)}$ is given by a multi-dimensional Itô process on $\mathscr{X}_t = \mathbf{R}^n$ (equipped with the corresponding Borel σ-algebra), defined by a stochastic differential equation (SDE)

$$\mathrm{d}X_t = a(X_t, t)\mathrm{d}t + B(X_t, t)\mathrm{d}\mathscr{W}_t,$$

with drift $a(x, t)$, diffusion matrix $B(x, t)$, multi-dimensional standard Wiener process $\mathscr{W}_t$, and initial value given by the random variable X_{t_0}. In this case, it is possible to sample directly (at least approximately) from the kernels $K_{s,t}$, when a suitable discretization method is applied, for instance, the Euler–Maruyama method.

4.6.2 Observations/Measurements

Let the process $X_{[t_0,\infty)}$ be observed via M random variables $Y_{1:M}$ with values in the measurable spaces $(\mathscr{Y}_j, \mathscr{B}_{\mathscr{Y}_j})$. Each single observation (measurement) Y_j depends on the state variable X_{t_j} at some time t_j and on the observation time (measurement time) t_j itself. We assume that, given the observation time t_j and the state $X_{t_j} = x_{t_j}$, the variable Y_j is independent of all other variables, and that the conditional probability measure can be expressed via some conditional probability density $g_j(y_j \mid x_{t_j}, t_j)$ with respect to a given reference measure $\mu_{\mathscr{Y}_j}$ on $(\mathscr{Y}_j, \mathscr{B}_{\mathscr{Y}_j})$. We

place no further conditions on g, such as linear dependence on the states, a normal distribution, or the like.

4.6.3 Observation Times/Measurement Times

The observation times (measurement times) t_j for $j = 1, \ldots, M$ are typically assumed to be deterministically given and known. At the ITWM, a variant of the particle filter was developed that is able to directly account for uncertainties in the measurement times themselves [1, 8]. Here, it is assumed that the observation times t_j are realizations of the random variables T_j. These variables thus model the uncertainty about the exact measurement times.

We will consider initially the standard case, which presupposes that all t_j are deterministically given and known. Formally, this corresponds to the case in which all t_j are random, but observed, so that all other emerging probabilities can be seen as conditionally depending on them. Therefore, we will always express this dependence on the time-points in our notation $g_j(y_j \mid x_{t_j}, t_j)$ for the observation density. For simplicity's sake, we also presuppose that the observation times $t_{1:M}$ are strictly arranged in ascending order, so that $t_0 < t_1 < \cdots < t_M$.

The standard particle filter is usually formulated for discrete-time Markov processes $X_{t_{0:M}} := (X_{t_j})_{j\in\{0,\ldots,M\}}$ with general state-space, so that the state variables are only defined for the initial time t_0 and those time-points $t_1, \ldots, t_M$ for which measurements exist. This case is included as a special case in a more generalized framework, however, in which the state variable X_t is defined for all times $t \geq t_0$ (one only has to pick out the states at the discretely given measurement times and ignore the others).

4.6.4 Full Model and Filter Model

The full model is given by the joint density of the variables $X_{t_{0:M}}$ and $Y_{1:M}$ (conditioned on the observation times $T_{1:M} = t_{1:M}$) with respect to the product measure $\mathscr{L}_{X_{t_{0:M}}} \prod_{j=1}^{M} \mu_{\mathscr{Y}_j}$:

$$f^{X_{t_{0:M}}, Y_{1:M} \mid T_{1:M}}(x_{t_{0:M}}, y_{1:M} \mid t_{1:M}) := \prod_{j=1}^{M} g_j(y_j \mid x_{t_j}, t_j). \tag{1}$$

The filter distribution at time t_k, in contrast, is based on a reduced model.

This filter model is given by the joint distribution density of the variables $X_{t_{0:k}}$ and $Y_{1:k}$ (given that $T_{1:M} = t_{1:M}$) with respect to the product measure $\mathscr{L}_{X_{t_{0:k}}} \prod_{j=1}^{k} \mu_{\mathscr{Y}_j}$:

$$f^{X_{t_{0:k}}, Y_{1:k} | T_{1:M}}(x_{t_{0:k}}, y_{1:k} \mid t_{1:M}) := \prod_{j=1}^{k} g_j(y_j \mid x_{t_j}, t_j). \tag{2}$$

This probability density is based on the state sequence $X_{t_{0:k}}$. In contrast, we can concentrate on the single state X_{t_k} by considering the joint density of the variables X_{t_k} and $Y_{1:k}$ (given that $T_{1:M} = t_{1:M}$) with respect to $\mathscr{L}_{X_{t_k}} \prod_{j=1}^{k} \mu_{\mathscr{Y}_j}$. This density can be calculated by marginalization as follows:

$$\begin{aligned} &f^{X_{t_k}, Y_{1:k} | T_{1:M}}(x_{t_k}, y_{1:k} \mid t_{1:M}) \\ &\quad := \int_{\{\tilde{x}_{t_{0:k}} \in \mathscr{X}_{t_{0:k}} : \tilde{x}_{t_k} = x_{t_k}\}} f^{X_{t_{0:k}}, Y_{1:k} | T_{1:M}}(\tilde{x}_{t_{0:k}}, y_{1:k} \mid t_{1:M}) \mathrm{d}\mathscr{L}_{X_{t_{0:k}}}(\tilde{x}_{t_{0:k}}). \end{aligned} \tag{3}$$

The filter density at time t_k with respect to $\mathscr{L}_{X_{t_k}}$ can be calculated by means of Bayes's Theorem:

$$f^{X_{t_k} | Y_{1:k}, T_{1:M}}(x_{t_k} \mid y_{1:k}, t_{1:M}) := \frac{f^{X_{t_k}, Y_{1:k} | T_{1:M}}(x_{t_k}, y_{1:k} \mid t_{1:M})}{f^{Y_{1:k} | T_{1:M}}(y_{1:k} \mid t_{1:M})} \tag{4}$$

with

$$f^{Y_{1:k} | T_{1:M}}(y_{1:k} \mid t_{1:M}) := \int_{\mathscr{X}_{t_{0:k}}} f^{X_{t_{0:k}}, Y_{1:k} | T_{1:M}}(x_{t_{0:k}}, y_{1:k} \mid t_{1:M}) \mathrm{d}\mathscr{L}_{X_{t_{0:k}}}(x_{t_{0:k}}). \tag{5}$$

For general (nonlinear) models, the practical calculation of the filter density is extremely difficult. Nonetheless, a Monte Carlo approximation can be calculated with the help of the particle filter. This is based on the crucial fact that filter densities $f^{X_{t_k} | Y_{1:k}, T_{1:M}}$, in contrast to the density of the full model, can be calculated recursively over time. This takes place in two steps. First, we consider the filter distribution at time t_{k-1} given by the probabilities

$$\begin{aligned} &\mathsf{P}(X_{t_{k-1}} \in B \mid Y_{1:k-1} = y_{1:k-1}, T_{1:M} = t_{1:M}) \\ &\quad = \int_B f^{X_{t_{k-1}} | Y_{1:k-1}, T_{1:M}}(x_{t_{k-1}} \mid y_{1:k-1}, t_{1:M}) \mathrm{d}\mathscr{L}_{X_{t_{k-1}}}(x_{t_{k-1}}) \end{aligned} \tag{6}$$

for each set $B \in \mathscr{B}_{\mathscr{X}_{t_{k-1}}}$. We initially obtain the prediction distribution, that is, the distribution of X_{t_k} given by the data $Y_{1:k-1}$ and $T_{1:M}$ by using the Markov kernel K_{t_{k-1}, t_k}:

$$\begin{aligned} &\mathsf{P}(X_{t_k} \in B \mid Y_{1:k-1} = y_{1:k-1}, T_{1:M} = t_{1:M}) \\ &\quad = \int_B \int_{\mathscr{X}_{t_{k-1}}} f^{X_{t_{k-1}} | Y_{1:k-1}, T_{1:M}}(x_{t_{k-1}} \mid y_{1:k-1}, t_{1:M}) \mathrm{d}\mathscr{L}_{X_{t_{k-1}}}(x_{t_{k-1}}) K_{t_{k-1}, t_k}(x_{t_{k-1}}, \mathrm{d}x_{t_k}) \end{aligned} \tag{7}$$

for each set $B \in \mathscr{B}_{\mathscr{X}_{t_k}}$.

In the second step, we then use Bayes's Theorem to obtain the filter distribution at time t_k:

$$\begin{aligned}
&\mathsf{P}(X_{t_k} \in B \mid Y_{1:k} = y_{1:k}, T_{1:M} = t_{1:M}) \\
&\quad = \int_B \frac{g_k(y_k \mid x_{t_k}, t_k)}{f^{Y_k|Y_{1:k-1},T_{1:M}}(y_k \mid y_{1:k-1}, t_{1:M})} \\
&\qquad \times \int_{\mathscr{X}_{t_{k-1}}} f^{X_{t_{k-1}}|Y_{1:k-1},T_{1:M}}(x_{t_{k-1}} \mid y_{1:k-1}, t_{1:M}) \mathrm{d}\mathscr{L}_{X_{t_{k-1}}}(x_{t_{k-1}}) \\
&\qquad \times K_{t_{k-1},t_k}(x_{t_{k-1}}, \mathrm{d}x_{t_k}) \qquad (8)
\end{aligned}$$

for each set $B \in \mathscr{B}_{\mathscr{X}_{t_k}}$, with the normalizing constant

$$\begin{aligned}
&f^{Y_k|Y_{1:k-1},T_{1:M}}(y_k \mid y_{1:k-1}, t_{1:M}) \\
&\quad := \int_{\mathscr{X}_{t_k}} g_k(y_k \mid x_{t_k}, t_k) \\
&\qquad \times \int_{\mathscr{X}_{t_{k-1}}} f^{X_{t_{k-1}}|Y_{1:k-1},T_{1:M}}(x_{t_{k-1}} \mid y_{1:k-1}, t_{1:M}) \mathrm{d}\mathscr{L}_{X_{t_{k-1}}}(x_{t_{k-1}}) \\
&\qquad \times K_{t_{k-1},t_k}(x_{t_{k-1}}, \mathrm{d}x_{t_k}). \qquad (9)
\end{aligned}$$

4.7 Particle Filter Algorithms for State Estimation

Particle filters [21, 30, 32] belong to the class of SMC methods used for state filtering in state-space models. Thus, with the appropriate adaptations and/or extensions, they also form the basis for parameter estimations. The standard particle filter works on discrete-time, nonlinear and non-Gaussian models and can be easily adapted for use on continuous-time systems with discrete-time measurements. The idea of the particle filter is to store a representation of the current filter distribution at each time-point by means of a set of weighted realizations (weighted samples or particles). This particle set is propagated through time in a suitable manner by adapting the realizations and the particle weights via the system dynamics and/or the measurements available at each time-point.

4.7.1 Importance Sampling

A key element of the particle filter is sequential importance sampling. We assume that a second Markov chain $\tilde{X}_{t_{0:M}}$ is given for the same state-space with pushforward measure $\mathscr{L}_{\tilde{X}_{t_j}}$ and Markov kernels $\tilde{K}_{t_{j-1},t_j}(x_{t_{j-1}}, \mathrm{d}x_{t_j})$ for $j = 1, \ldots, M$. We also assume that for each $x_{t_{j-1}} \in \mathscr{X}_{t_{j-1}}$, the measure $K_{t_{j-1},t_j}(x_{t_{j-1}}, \cdot)$ is absolutely continuous with respect to the measure $\tilde{K}_{t_{j-1},t_j}(x_{t_{j-1}}, \cdot)$.

It follows that the Radon–Nikodym derivative (written as a conditional probability density)

$$\varrho_{t_j|t_{j-1}}(x_{t_j} \mid x_{t_{j-1}}) := \frac{K_{t_{j-1},t_j}(x_{t_{j-1}}, \mathrm{d}x_{t_j})}{\tilde{K}_{t_{j-1},t_j}(x_{t_{j-1}}, \mathrm{d}x_{t_j})}$$

exists. We also require that the pushforward measure $\mathscr{L}_{X_{t_0}}$ under X_{t_0} is absolutely continuous with respect to the corresponding pushforward measure $\mathscr{L}_{\tilde{X}_{t_0}}$ under $\tilde{X}_{t_0}$ with the Radon–Nikodym derivative

$$\varrho_{t_0}(x_{t_0}) := \frac{\mathrm{d}\mathscr{L}_{X_{t_0}}(x_{t_0})}{\mathrm{d}\mathscr{L}_{\tilde{X}_{t_0}}(x_{t_0})}.$$

Sequential importance sampling can be performed under those circumstances in which we are able to draw random realizations from both the initial distribution $\mathscr{L}_{\tilde{X}_{t_0}}$ and the kernels

$$\tilde{K}_{t_{j-1},t_j}(x_{t_{j-1}}, \cdot)$$

for each $x_{t_{j-1}} \in \mathscr{X}_{t_{j-1}}$, and when we can calculate $\varrho_{t_0}(x_{t_0})$ and $\varrho_{t_j|t_{j-1}}(x_{t_j} \mid x_{t_{j-1}})$ pointwise.

Using

$$K_{t_{k-1},t_k}(x_{t_{k-1}}, \mathrm{d}x_{t_k}) = \varrho_{t_k|t_{k-1}}(x_{t_k} \mid x_{t_{k-1}})\tilde{K}_{t_{k-1},t_k}(x_{t_{k-1}}, \mathrm{d}x_{t_k}),$$

we can then rewrite the recursive formula (8) for the filter distribution at time t_k as follows:

$$\begin{aligned}
&\mathsf{P}(X_{t_k} \in B \mid Y_{1:k} = y_{1:k}, T_{1:M} = t_{1:M}) \\
&\quad = \int_B \frac{g_k(y_k \mid x_{t_k}, t_k)}{f^{Y_k|Y_{1:k-1},T_{1:M}}(y_k \mid y_{1:k-1}, t_{1:M})} \\
&\qquad \times \int_{\mathscr{X}_{t_{k-1}}} f^{X_{t_{k-1}}|Y_{1:k-1},T_{1:M}}(x_{t_{k-1}} \mid y_{1:k-1}, t_{1:M}) \\
&\qquad \times \varrho_{t_k|t_{k-1}}(x_{t_k} \mid x_{t_{k-1}})\mathrm{d}\mathscr{L}_{X_{t_{k-1}}}(x_{t_{k-1}}) \\
&\qquad \times \tilde{K}_{t_{k-1},t_k}(x_{t_{k-1}}, \mathrm{d}x_{t_k})
\end{aligned} \tag{10}$$

for each $B \in \mathscr{B}_{\mathscr{X}_{t_k}}$.

The direct calculation of the normalizing constants

$$f^{Y_k|Y_{1:k-1},T_{1:M}}(y_k \mid y_{1:k-1}, t_{1:M})$$

(while the values $y_{1:M}$ are considered to be fixed) is unnecessary.

Sequential importance sampling is then performed as follows: we randomly draw a number N of realizations $x^i_{t_0}$ from $\mathscr{L}_{\tilde{X}_{t_0}}$ and calculate the corresponding unnormalized weights

$$w^i_{t_0} := \varrho_{t_0}\big(x^i_{t_0}\big) \quad \text{for all } i = 1, \ldots, N.$$

We then randomly draw for all $k = 1, \ldots, M$ realizations $x^i_{t_k}$ from the kernel

$$\tilde{K}_{t_{k-1},t_k}\big(x^i_{t_{k-1}}, \mathrm{d}x_{t_k}\big)$$

for each $i = 1, \ldots, N$ and calculate the unnormalized weights

$$w^i_{t_k} := \varrho_{t_k|t_{k-1}}\big(x^i_{t_k} \mid x^i_{t_{k-1}}\big) g_k\big(y_k \mid x^i_{t_k}, t_k\big) w^i_{t_{k-1}} \quad \text{for all } i = 1, \ldots, N.$$

For suitable integrable functions h (for example, when certain restrictions are fulfilled on the rate with which h may increase relative to x; see [34] for details), one can approximate the expected value of h with respect to the filter density conditioned on the observations $Y_{1:k} = y_{1:k}$, given by

$$\begin{aligned}
&\mathsf{E}\big[h(X_{t_k}) \,\big|\, Y_{1:k} = y_{1:k}, T_{1:M} = t_{1:M}\big] \\
&\quad := \mathsf{E}_{f^{X_{t_k}|Y_{1:k}=y_{1:k},T_{1:M}=t_{1:M}}(\cdot|y_{1:k},t_{1:M})}\big[h(X_{t_k})\big] \\
&\quad = \int f^{X_{t_k}|Y_{1:k},T_{1:M}}(x_{t_k} \mid y_{1:k}, t_{1:M}) h(x_{t_k}) \mathrm{d}\mathscr{L}_{X_{t_k}}(x_{t_k}),
\end{aligned} \tag{11}$$

by means of

$$\mathsf{E}_{t_k,N}\big[h(X_{t_k}) \,\big|\, Y_{1:k} = y_{1:k}, T_{1:M} = t_{1:M}\big] := \frac{\sum_{i=1}^N w^i_{t_k} h(x^i_{t_k})}{\sum_{i=1}^N w^i_{t_k}} \tag{12}$$

where N is the number of particles. It can be shown that as N approaches infinity, these empirical expected values converge to the expected values from the filter distribution:

$$\lim_{N\to\infty} \mathsf{E}_{t_k,N}\big[h(X_{t_k}) \,\big|\, Y_{1:k} = y_{1:k}, T_{1:M} = t_{1:M}\big] = \mathsf{E}\big[h(X_{t_k}) \,\big|\, Y_{1:k} = y_{1:k}, T_{1:M} = t_{1:M}\big]. \tag{13}$$

If we are in a position to sample from the Markov kernels X_{t_j} of the states themselves, then we can select $\tilde{X}_{t_j} = X_{t_j}$ (at least in distribution), from which $\varrho_{t_0}(x_{t_0}) \equiv 1$ and $\varrho_{t_j|t_{j-1}}(x_{t_j} \mid x_{t_{j-1}}) \equiv 1$ follow. This selection is indeed standard, but it is not always

the best choice with regard to the effectiveness of the particle filter algorithm. Finding a Markov chain $\tilde{X}_{t_{0:M}}$ different than $X_{t_{0:M}}$ that can improve the effectiveness of the algorithm, however, is an application-specific, and not always simple, task.

4.7.2 Resampling

Sequential importance sampling converges when the number of samples (particles) increases exponentially over time. This is not practicable; typically, N is even held constant over time. However, when the number N of particles remains constant over time, the particles propagated by sequential importance sampling quickly degenerate, since most of the normalized weights converge rapidly toward 0.

The degree of degeneracy of the particle set is often measured by an estimator for the so-called Effective Sample Size (ESS). This estimator at time t is given by

$$n_{\mathrm{ESS}} := \frac{1}{\sum_{i=1}^{N} (\tilde{w}_t^i)^2}, \tag{14}$$

where

$$\tilde{w}_t^i := \frac{w_t^i}{\sum_{i=1}^{N} w_t^i} \tag{15}$$

refer to the normalized weights.

The ESS estimator assumes its maximum value N (number of particles), when all weights are equal, and it approaches 1 when the variance of the weights, and thus the degree of degeneracy, becomes large. To avoid this degeneration, one must insert a resampling step in the algorithm, to be performed when the ESS drops below a certain threshold $N_{\mathrm{Threshold}}$ (usually selected to be $N/2$).

Resampling at time-point s_ℓ is based on given, non-negative (unnormalized) selection weights $v_{s_\ell}^i$ for each particle index i. One repeats random selections (with replacement) of particles having probabilities p_ℓ^i given by the normalized selection weights

$$p_\ell^i := \frac{v_{s_\ell}^i}{\sum_{v=1}^{N} v_{s_\ell}^v}. \tag{16}$$

This is referred to as multinomial resampling. There are also procedures in which each individual particle continues to be selected with probability p_ℓ^i, but which exhibit a reduced overall variance, such as Stratified Resampling or Systematic Resampling, and these

should be chosen in preference to multinomial resampling (see [29, 33]). In any case, resampling defines a (random) selection function $\iota_\ell : I \to I$ on the index set $I := \{1, \ldots, N\}$.

The resampling step is then performed in two phases:

- Replacement of the state samples $(x^i_{s_\ell})_{i=1,\ldots,N}$ by the selected state samples $(x^{\iota_\ell(i)}_{s_\ell})_{i=1,\ldots,N}$.
- Replacement of the unnormalized weights $(w^i_{s_\ell})_{i=1,\ldots,N}$ by the corrected unnormalized weights $(w^{\iota_\ell(i)}_{s_\ell}/v^{\iota_\ell(i)}_{s_\ell})_{i=1,\ldots,N}$.

It is necessary to correct the weights in the final step in order to compensate for the bias introduced in the particle distribution by the selection process. This bias results from the following consideration: Before sampling, the selection probability for particle i (at each draw) is given by p^i_ℓ. The expected value for the number of times that particle i will actually be drawn after N samplings is therefore $Np^{\iota_\ell(i)}_\ell$.

As a result, each normalized weight $\tilde{w}^i_{s_\ell}$, for each selected particle i, must be corrected by replacing it with the weight

$$\frac{\tilde{w}^{\iota_\ell(i)}_{s_\ell}}{Np^{\iota_\ell(i)}_\ell} \Big/ \sum_{\nu=1}^{N} \frac{\tilde{w}^{\iota_\ell(\nu)}_{s_\ell}}{Np^{\iota_\ell(\nu)}_\ell} = \frac{w^{\iota_\ell(i)}_{s_\ell}}{v^{\iota_\ell(i)}_{s_\ell}} \Big/ \sum_{\nu=1}^{N} \frac{w^{\iota_\ell(\nu)}_{s_\ell}}{v^{\iota_\ell(\nu)}_{s_\ell}} \tag{17}$$

(using (16)).

Note that in the original particle filter, the selection weights $v^i_{s_\ell}$ at time s_ℓ are chosen so that they are given by the particle weights (before the replacement), that is,

$$v^i_{s_\ell} = w^i_{s_\ell} \quad \text{for } i = 1, \ldots, N,$$

so that after the resampling step, the unnormalized weights are all equal to 1. Nonetheless, in general, their choice is free and may be influenced by the system observations (measurements), for example (as used in the so-called Auxiliary Particle Filter [42]).

4.7.3 Particle Filter Algorithm

The particle filter calculates the state realizations and weights recursively through time. In its standard form, the particle filter can be specified as pseudo code, as in Algorithm 1.

Algorithm 1 Standard particle filter

{*At time* t_0*:*}
  Randomly sample N state realizations $(x^i_{t_0})_{i=1,\dots,N}$ of $\tilde{X}_{t_0}$ with large N.
  for all $i = 1, \dots, N$ **do**
    Set the weight $w^i_{t_0} = \varrho_{t_0}(x^i_{t_0})$.
  end for
for all times t_k, $k = 1, \dots, M$ **do**
  {*Resample the particles* $(x^i_{t_{k-1}})_{i=1,\dots,N}$*, if necessary (e.g., if the ESS drops below a threshold):*}
    Randomly generate a selection function ι according to certain selection weights $(v^i_{t_{k-1}})_{i=1,\dots,N}$.
    for $i = 1, \dots, N$ **do**
      Replace the state realization $x^i_{t_{k-1}}$ by the selection $x^{\iota(i)}_{t_{k-1}}$.
      Replace the unnormalized weight $w^i_{t_{k-1}}$ by the corrected weight $w^{\iota(i)}_{t_{k-1}} / v^{\iota(i)}_{t_{k-1}}$.
    end for
  for $i = 1, \dots, N$ **do**
    Randomly sample a realization $x^i_{t_k}$ from the Markov kernel

$$\tilde{K}_{t_{k-1},t_k}\big(x^i_{t_{k-1}}, \cdot\big).$$

    Update the weight by:

$$w^i_{t_k} = \varrho_{t_k|t_{k-1}}\big(x^i_{t_k} \mid x^i_{t_{k-1}}\big) g_k\big(y_k \mid x^i_{t_k}, t_k\big) w^i_{t_{k-1}}.$$

  end for
  For given suitable integrable functions h, calculate the estimates

$$\mathsf{E}_{t_k,N}\big[h(X_{t_k}) \mid Y_{1:k} = y_{1:k}, T_{1:M} = t_{1:M}\big] := \frac{\sum_{i=1}^N w^i_{t_k} h(x^i_{t_k})}{\sum_{i=1}^N w^i_{t_k}}.$$

end for

Note that, in choosing $\tilde{X}_{[t_0,\infty)} = X_{[t_0,\infty)}$ (in distribution), the identity

$$\varrho_{t_k|t_{k-1}}\big(x^i_{t_k} \mid x^i_{t_{k-1}}\big) \equiv 1$$

holds, and the updating of the weights simplifies to

$$w^i_{t_k} = g_k\big(y_k \mid x^i_{t_k}, t_k\big) w^i_{t_{k-1}}.$$

4.7.4 Data Likelihood

The model validation and discrimination are generally based on the data likelihood

$$\begin{aligned} Z_{t_k}(t_{1:M}) := f^{Y_{1:k}|T_{1:M}}(y_{1:k} \mid t_{1:M}) &= \int_{\mathscr{X}_{t_{0:k}}} f^{X_{t_{0:k}},Y_{1:k}|T_{1:M}}(x_{t_{0:k}}, y_{1:k} \mid t_{1:M}) \mathrm{d}\mathscr{L}_{X_{t_{0:k}}}(x_{t_{0:k}}) \\ &= \mathsf{E}\big[f^{X_{t_{0:k}},Y_{1:k}|T_{1:M}}(\cdot, y_{1:k} \mid t_{1:M})\big] \end{aligned} \tag{18}$$

for given observations $y_{1:k}$. Without resampling, the data likelihood could be approximated by the empirical average of the unnormalized weights, that is, by

$$\hat{Z}_{t_k}(t_{1:M}) := \frac{1}{N}\sum_{i=1}^{N} w_{t_k}^i, \tag{19}$$

since this is the empirical estimate of the above expected value. After a resampling step, this no longer holds true.

In any case (with or without resampling), the ratio estimator

$$Z_{t_k}(t_{1:M})/Z_{t_{k-1}}(t_{1:M})$$

can be used to recursively approximate the data likelihood:

$$\frac{\widehat{Z_{t_k}(t_{1:M})}}{Z_{t_{k-1}}(t_{1:M})} := \frac{\sum_{i=1}^{N} \varrho_{t_k|t_{k-1}}(x_{t_k}^i \mid x_{t_{k-1}}^i) g_k(y_k \mid x_{t_k}^i, t_k) w_{t_{k-1}}^i}{\sum_{i=1}^{N} w_{t_{k-1}}^i}, \tag{20}$$

with the initial estimator $\hat{Z}_{t_0}(t_{1:M}) = 1$ (see [25], for example).

4.8 Kalman Filter

As mentioned, the explicit formulaic calculation of the filter distributions is only possible in a very few instances. This is due to the recursively nested, high-dimensional integrals, which, in general, cannot be analytically solved. In two cases, however, this is still possible and practicable. In the first case, one assumes that the state-space can only jump between a finite number of discrete states (here, one is also usually working with a discrete-time model). The measurement disturbances have a normal distribution. One speaks here of a hidden Markov model in the narrower sense. One is dealing with a finite number of transitional probabilities, which can be directly calculated, and the filter distributions can be determined accordingly via a recursive procedure by means of direct computation. The second case, which is far more important for modeling, is given by linear systems with exclusively Gaussian disturbances. Gauss distributions are characterized solely by the first two moments (average and variance). Moreover, in linear systems, the Gaussian form is retained for all other relevant distributions. The appropriate state filter is the Kalman filter: The average and variance can be recursively calculated, simply and directly, using matrix operations.

Based on the terminology developed in the previous section, the Kalman filter can be derived as a special case without much effort. The Kalman filter is only correct as a state filter when we subject our system to certain restrictions: linearity and Gaussian normality

in the state and measurement processes. The assumption of linearity in the entire system means that all system disturbances remain normally distributed, regardless of whether one propagates them forward or backward through the system. This is evident by the corresponding property of the multivariate Gauss distribution.

Here, we consider only the discrete-time case.

4.8.1 Multivariate Normal Distribution and Linearity

A random variable X has a multivariate normal distribution (multivariate Gauss distribution), denoted $X \sim \mathcal{N}_d(\mu, \Sigma)$, with average $\mu \in \mathbf{R}^d$ and positive-semi-definite covariance matrix $\Sigma \in \mathbf{R}^{d\times d}$, when there is a random ℓ-vector Z with standard, normally-distributed coefficients and a matrix $A \in \mathbf{R}^{k\times \ell}$ with $AA^\top = \Sigma$, such that

$$X = AZ + \mu.$$

If the covariance matrix Σ is positive-definite rather than positive-semi-definite, then there exists a corresponding probability density, and it is given by

$$\frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu)^\top \Sigma^{-1}(x-\mu)\right).$$

An affine-linear transformation $Y = BX + c$, with $B \in \mathbf{R}^{m\times d}$ and $c \in \mathbf{R}^m$, leads to a variable Y, which also has a multivariate normal distribution:

$$Y \sim \mathcal{N}_m\big(B\mu + c, B\Sigma B^\top\big).$$

Special cases are given by:

- Marginalization: Let $X = (X_1, X_2)^\top$. Then X_1 (and X_2) are normally distributed, since

$$X_1 = BX \quad \text{with } B = \begin{pmatrix} I & 0 \\ 0 & 0 \end{pmatrix}.$$

- Let $Y = BX + V$, with $V \sim \mathcal{N}_m(0, R)$. It then follows that $Y \sim \mathcal{N}(BX, B\Sigma B^\top + R)$, since

$$Y = \begin{pmatrix} B & I \end{pmatrix} \begin{pmatrix} X \\ V \end{pmatrix}.$$

Note that, when

- $p(x)$ is normally distributed,

$$X \sim \mathcal{N}_d(\mu, \Sigma),$$

- $p(y \mid x)$ is conditionally normally distributed to a given x,

$$Y \mid (X = x) \sim \mathcal{N}_m\big(\hat{y}(x), R\big),$$

and

- the average $\hat{y}$ is affine-linearly dependent on x,

$$\hat{y}(x) = Bx + c,$$

then

- the joint distribution density $p(x, y) = p(y \mid x)p(x)$ is normal and
- the marginal distribution density $p(y) = \int p(x, y)dx$ is normal.

Here, the joint distribution density $p(x, y)$ is given by

$$\begin{aligned}\begin{pmatrix} X \\ Y \end{pmatrix} &\sim \mathcal{N}_{d+m}\left(\begin{pmatrix} 1 \\ B \end{pmatrix}\mu + \begin{pmatrix} 0 \\ c \end{pmatrix}, \begin{pmatrix} 1 \\ B \end{pmatrix}\Sigma\begin{pmatrix} 1 \\ B \end{pmatrix}^{\top} + \begin{pmatrix} 0 & 0 \\ 0 & R \end{pmatrix}\right) \\ &= \mathcal{N}_{d+m}\left(\begin{pmatrix} \mu \\ B\mu + c \end{pmatrix}, \begin{pmatrix} \Sigma & \Sigma B^{\top} \\ B\Sigma & B\Sigma B^{\top} + R \end{pmatrix}\right),\end{aligned}$$

and the marginal distribution density $p(y)$ is given by

$$Y \sim \mathcal{N}_m\big(B\mu + c, B\Sigma B^{\top} + R\big).$$

In the following treatment, we denote the average and covariance matrix of Y as

$$\mu_y = B\mu + c, \quad \text{and} \quad \Sigma_y = B\Sigma B^{\top} + R.$$

4.8.2 Bayes's Theorem for Normal Distributions: Kalman Gain

Let us consider Bayes's Theorem

$$p(y \mid x)p(x) = p(x, y) = p(x \mid y)p(y)$$

under the prerequisites listed above. All distribution densities that occur are thus normal. It remains to be shown that this is also correct for the posterior distribution $p(x \mid y)$. Let us take the approach

$$X \mid (Y = y) \sim \mathcal{N}_d(Ky + e, G),$$

and consider both the left and right sides of the previous equation. In so doing, we obtain the equation

$$\begin{aligned}&\mathcal{N}_{d+m}\left(\begin{pmatrix} \mu \\ B\mu + c \end{pmatrix}, \begin{pmatrix} \Sigma & \Sigma B^{\top} \\ B\Sigma & B\Sigma B^{\top} + R \end{pmatrix}\right) \\ &= \mathcal{N}_{d+m}\left(\begin{pmatrix} K\mu_y + e \\ \mu_y \end{pmatrix}, \begin{pmatrix} K\Sigma_y K^{\top} + G & K\Sigma_y \\ \Sigma_y K^{\top} & \Sigma_y \end{pmatrix}\right).\end{aligned}$$

Solving for K, G and e leads to:

$$\begin{aligned} K &= \Sigma B^\top \Sigma_y^{-1} \quad \text{(Kalman gain)}, \\ G &= \Sigma - K\Sigma_y K^\top = \Sigma - K\Sigma_y \Sigma_y^{-1} B\Sigma = (I - KB)\Sigma, \\ e &= \mu - K\mu_y = \mu - K(B\mu + c). \end{aligned}$$

The last equation yields:

$$Ky + e = \mu + K\big(y - (B\mu + c)\big).$$

4.8.3 Application to Recursive State Filtering: the Kalman Filter

As just shown, the posterior distribution $p(x \mid y)$ is given by

$$\mathcal{N}_d\big(\mu + K\big(y - (B\mu + c)\big), (I - KB)\Sigma\big) \quad \text{with } K = \Sigma B^\top \Sigma_y^{-1}.$$

We now consider the linear, normal, dynamic model:

$$x_0 \sim \mathcal{N}_d(\hat{x}_0, Q_0), \qquad x_t \sim \mathcal{N}_d\big(A_t x_{t-1} + b(u_{t-1}), Q_t\big), \qquad y_t \sim \mathcal{N}_m(C_t x_t, R_t).$$

We further assume that $p(x_{t-1} \mid y_{1:t-1})$ is recursively given by

$$\mathcal{N}_d(\hat{x}_{t-1|t-1}, P_{t-1|t-1}),$$

starting with $p(x_0)$, that is, $\hat{x}_{0|0} = \hat{x}_0$ and $P_{0|0} = Q_0$. We must now show that $p(x_t \mid y_{1:t})$ is also normally distributed, that is, it is given by

$$\mathcal{N}_d(\hat{x}_{t|t}, P_{t|t}).$$

The Kalman filter is calculated in two steps:

- Prediction:

$$p(x_t \mid y_{1:t-1}) = \int p(x_t \mid x_{t-1}) p(x_{t-1} \mid y_{1:t-1}) dx_{t-1}$$

  is given by $\mathcal{N}_d(\hat{x}_{t|t-1}, P_{t|t-1})$, with

$$\hat{x}_{t|t-1} = A_t \hat{x}_{t-1|t-1} + b(u_{t-1}), \quad \text{and} \quad P_{t|t-1} = A_t P_{t-1|t-1} A_t^\top + Q_t.$$

- Update:

$$p(x_t \mid y_{1:t}) = \frac{p(y_t \mid x_t)\, p(x_t \mid y_{1:t-1})}{p(y_t \mid y_{1:t})}$$

is given by $\mathcal{N}_d(\hat{x}_{t|t}, P_{t|t})$, with

$$S_t = C_t P_{t|t-1} C_t^\top + R_t, \qquad K_t = P_{t|t-1} C_t^\top S_t^{-1},$$
$$\hat{x}_{t|t} = \hat{x}_{t|t-1} + K_t\big(y_t - (C_t \hat{x}_{t|t-1})\big), \qquad P_{t|t} = (I - K_t C_t) P_{t|t-1}.$$

In the continuous-time case, we have a similar situation for the solution of the corresponding differential equations.

4.9 Extended Kalman Filter

As mentioned above, in linear systems with normally distributed disturbances, the normal distribution is also transferred to all other relevant distributions, including the filter distribution. This is not so for nonlinear systems, even when all disturbances are assumed to be normally distributed. The filter distribution, in particular, can be arbitrarily complex in these cases. Except for the simple cases, in which merely an additional asymmetry appears in the distribution, there can also be filter distributions with multiple local maxima (modes). Although such distributions can only be very poorly approximated by the single-mode Gauss distribution, this type of approximation is standard and is applied in the vast majority of cases. One does so by linearizing the nonlinear system for the current state values and then applying the Kalman filter to this linearized system. The resulting filter is then called an Extended Kalman Filter (EKF). However, due to the aforementioned poor approximation of Gauss distributions for multi-mode filter distributions, general convergence results for this filter are not to be expected. One often tries to improve at least the covariance values of the EKF estimator by increasing computational efforts, which occurs in the case of the Unscented Kalman Filter, for example. However, the fundamental problem of approximating complex filter distributions using unimodal Gauss distributions still remains.

4.10 MTU-PF: Accounting for Uncertainties in the Measurement Time-Points when State Filtering with the Particle Filter

We now return to the general case of the stochastic state-space and want to dispense with the assumption that the observation time-points (measurement time-points) t_j for $j = 1, \ldots, M$ are given and known deterministically. Instead, we want to assume that the

observation times t_j are realizations of random variables T_j. These variables thus model the uncertainty about the exact measurement time-points. In contrast to the observation variables Y_j themselves, the observation times T_j are never directly observed (measured). Instead, we assume that the only information we have at our disposal is their probability distribution on the half-axis $[t_0, \infty)$ itself, whereas, for the observations Y_j, we know both the densities $g_j(y_j \mid x_{t_j}, t_j)$ *and* the observed values y_j themselves. Consequently, we have here a significant conceptual difference.

We consider here only the simplest case, in which each time variable T_j is independent from each other time variable. Indeed, this contradicts the fact that measurement values typically follow a prescribed chronology, such as $T_1 < T_2 < T_3 < \cdots$, which would imply a stochastic dependency between the variables T_j. However, this would lead to substantially more complicated algorithms. Moreover, dependencies in the chronology can also be simply introduced via appropriate restrictions to the support $\operatorname{supp} T_j$ of the random variables T_j, for example, by requiring that all elements of $\operatorname{supp} T_j$ are smaller than all elements of $\operatorname{supp} T_{j+1}$. In this way, the independence of the variables from one another is preserved. In general, the probability distribution of each individual variable T_j should be given by a density $\gamma_j(t_j)$ relative to the Lebesgue measure $\lambda_{[t_0,\infty)}$ on the interval $[t_0, \infty)$.

Normally, the uncertainty in the measurement time-points is incorporated into the uncertainty of the measurement value by increasing the latter's variance (lumped measurement disturbances; see Fig. 3). This generally leads to parameters that can only be very conservatively estimated (with large uncertainties); for rapidly changing states, however, this procedure can also lead to really erroneous estimates (see Fig. 4(b)–(d)).

The standard particle filter can be extended appropriately (see [1, 8]). If there really are uncertainties in the measurement time-points, the resulting Measurement Time Uncertainty-Particle Filter (MTU-PF) delivers substantially better estimates than the standard particle filter (see Fig. 4 (a)).

The main difference between the MTU and the standard particle filter is that the weights are not just updated at discrete time-points (in the standard filter, at the exact measurement time-points); in principle, they are updated continuously, over all time-points. This results from the fact that the measurement time-points themselves are "smeared" across the time axis due to the densities $\gamma_j(t_j)$. This opens up a much broader range of possibilities for stabilizing the algorithm—for example, by choosing an adaptive increment control based on the development of the ESS estimator. With strongly decreasing ESS (i.e., high risk of algorithm degeneration), a smaller time increment (step size) can be selected so that early, repeated resampling can keep the particle set in good condition (at least from the standpoint of a high ESS value) (see Fig. 5). This is not possible in the standard case, since here, the algorithm's step size is fixed by the measurement intervals. More details can be found in [1] and [8]. In Sect. 7, we will introduce a biomedical application of this MTU particle filter and compare it to the standard particle filter.

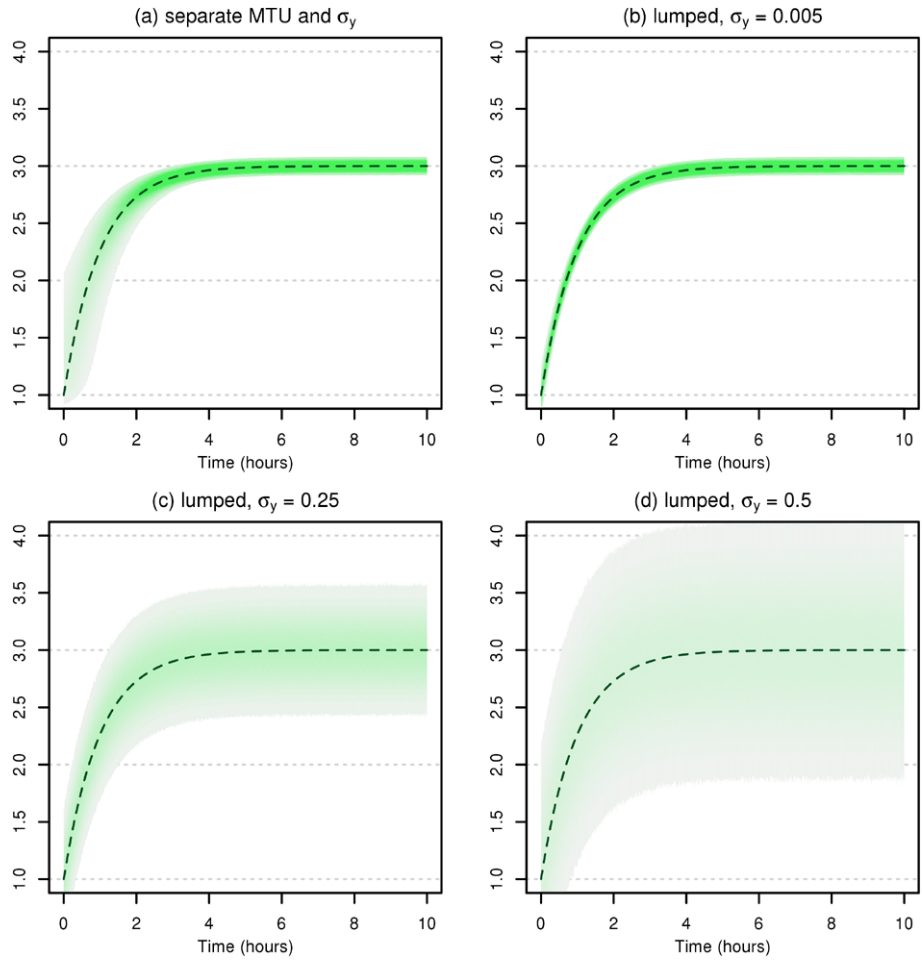

Fig. 3 Monte Carlo simulation of measurement values for a simple model consisting of one state with exponential growth and normally-distributed disturbances, as well as normally-distributed disturbances in the measurement values. The *dashed green lines* represent the nominal development (average) of the state over time. The *green shaded areas* show the distribution of the measurement times and values. (**a**) Separate modeling of uncertainty in measurement time-points (*horizontal*) and values (*vertical*); variance of the measurement values $\sigma_y = 0.005$. (**b**)–(**d**) Without special modeling of the uncertainty in the measurement time-points; scattering only in the measurement values (*vertical*). Here, to compensate, the variance σ_y of the measurement values is gradually increased. It shows that this compensation in (**b**)–(**d**) cannot adequately reproduce the distribution of the measurement values in case (**a**). This is especially visible for areas with steep increases (early time-points). Whereas in (**a**), the distributions in the y-direction scatter more here than where the state is constant (later time-points), in (**b**)–(**d**), the scattering in the y-direction is equally pronounced everywhere, as dictated by the model

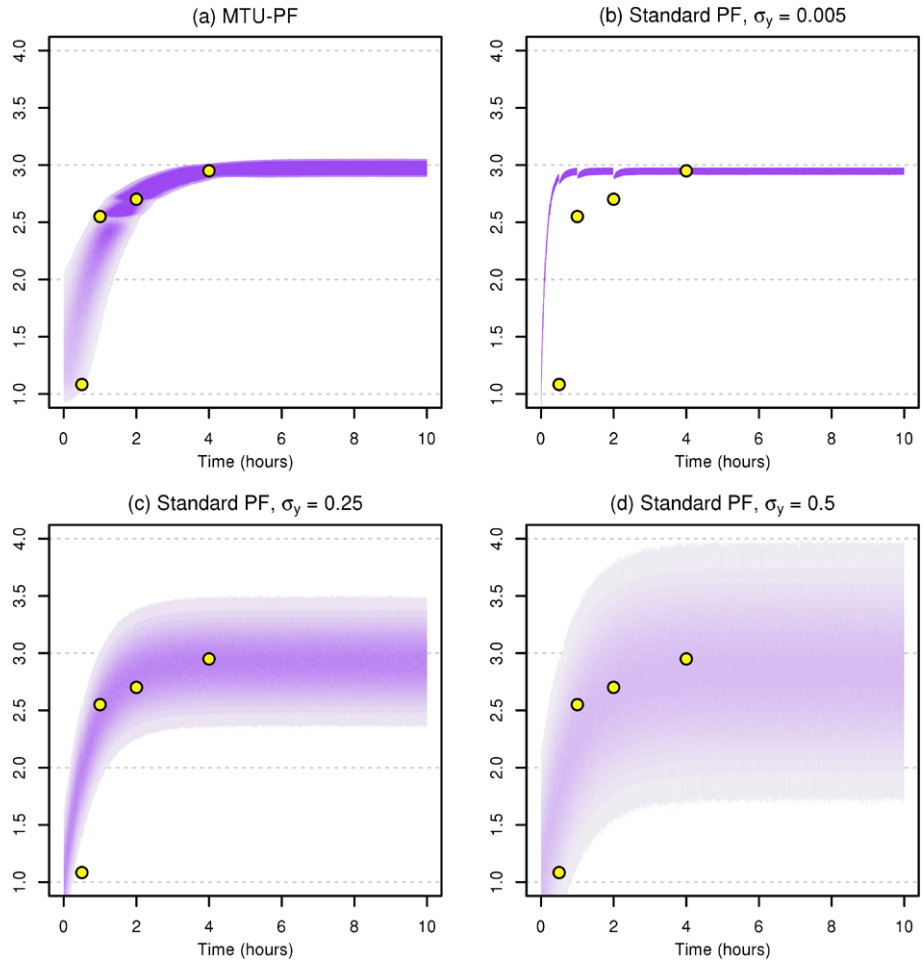

Fig. 4 Simulated distributions of measurement values after estimating parameters and states in the sample model with various filters. The *yellow points* correspond to the measurements used for the estimates, which were generated as random realizations of the distribution shown in Fig. 3(a). (**a**) MTU particle filter; (**b**)–(**d**) standard particle filter with different measurement value variances σ_y. The *purple shaded areas* show the distributions of the measurements that would result from the various estimates of the particle filters. The actual distribution of the measurement values can be seen in Fig. 3(a). The estimates with the MTU particle filter deliver clearly better matching distributions than the standard particle filter

4.11 PF-MPC as a Particle Filter-Based MPC Approach

At the ITWM, we developed a generalized, stochastic, nonlinear Model Predictive Control (MPC) approach based on a double application of the particle filter [3]. Model Predictive Control refers to a class of model-based controllers whose development began in the

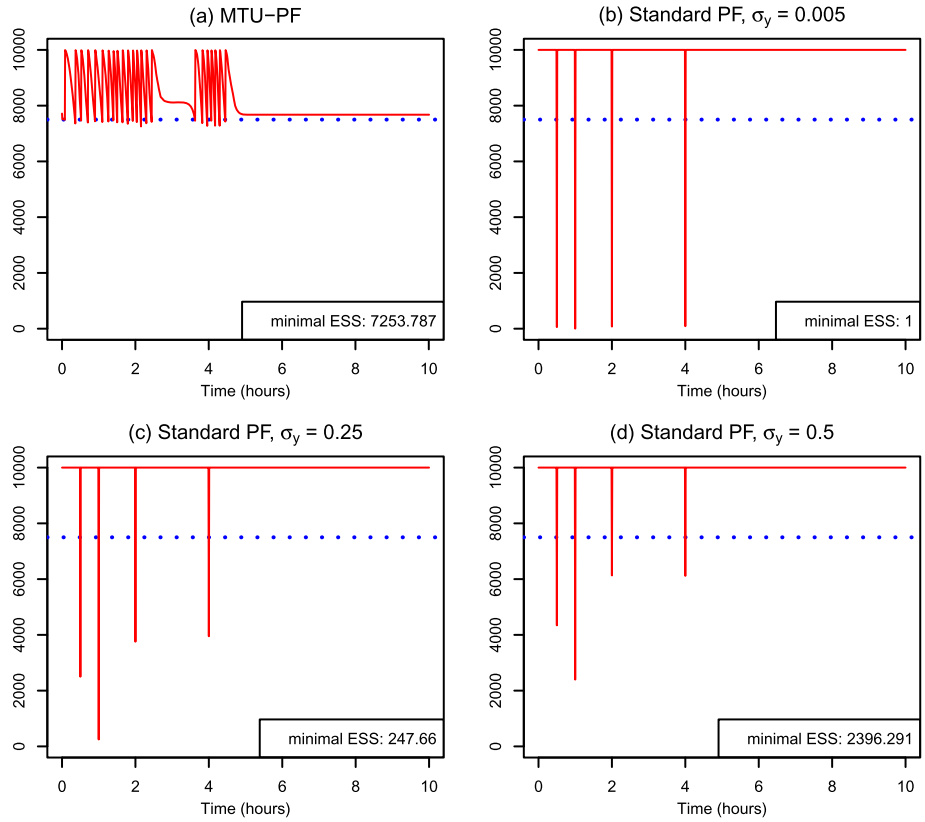

Fig. 5 Comparison of ESS estimates during filtering. (**a**) MTU particle filter; (**b**)–(**d**) standard particle filter with different measurement value variances σ_y. Clearly, the ESS in the standard filters (**b**)–(**d**) falls off significantly at the discrete time-points for which a measurement is available. In (**a**), this is prevented by early resampling

1970s. Unlike traditional control approaches, such as PID controllers, the control signal is not determined solely by the current state (or an estimate thereof). Instead, the MPC controller makes use of a system model to enable predictive calculation of the system's development under the influence of the control signal u_j. Based on these predictions, the control signal is defined over a particular time period (horizon) T_p, so as to minimize a given target function J. Then, the first value of this calculated control signal is delivered to the system as a control input. This procedure is continually repeated over time.

In the past, the particle filter was often used for state estimation in the context of an MPC approach. Our approach is new in the sense that our controller not only uses the particle filter for state estimation, but also for solving the optimization problem. This is accomplished by considering the control targets as virtual measurements. After adding the control variable to the state variables, the optimization problem reduces to a filter problem: the conditioned distributions of the control signal under given targets can thus

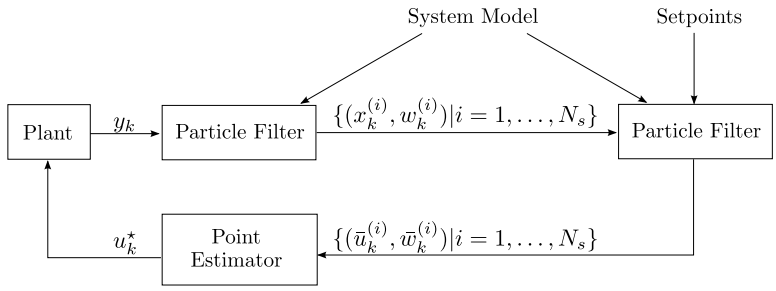

Fig. 6 Schematic diagram of the PF-MPC controller. Here, $x_k^{(i)}$ is the state vector of the i-th particle with weight $w_k^{(i)}$ at time k in the first particle filter for state estimation; y_k is the measurement at time k; $\bar{u}_k^{(i)}$ is the value of the control variable of the i-th particle with weight $\bar{w}_k^{(i)}$, as output of the second particle filter. This particle set serves to calculate the optimal control value u_k^*; N_s is the number of particles in each filter

be understood as a filter distribution and can therefore be calculated with a second particle filter. On the basis of these filter distributions for the control signal, the filtering trajectory most likely to lead to good system behavior can then be selected (see Fig. 6). The first value of this trajectory then serves as the next control input.

In the standard MPC approach, the target function $J = J(x_k, \bar{u}_{k:(k+T_p)}, T_p)$ is usually of the form

$$J = \sum_{j=k}^{k+T_p} \|\bar{u}_j - \bar{u}_{j-1}\|_Q^2 + \sum_{j=k+1}^{k+T_p} \|s_j - x_j\|_R^2.$$

Here, the norms refer to weighted Euclidian norms with the weighting matrices Q and R. The first term ensures that the difference between consecutive control values $\bar{u}_{j-1}$ and $\bar{u}_j$ remains small. The second term penalizes deviations of the system states x_j from the target states s_j; these target states describe those state trajectories that the system is to preferentially follow. The trick is that a minimization of J corresponds to a maximization of $\exp^{-1/2J}$, that is—except for a normalizing constant—to a multivariate Gauss distribution density. For given states x_j, this can be viewed as a joint distribution of the control signal transition probabilities and the observation probabilities of s_j. But the variant of the particle filter introduced above realizes just that. The treatment of the target function as, in its essence, a distribution density immediately opens the possibility of lifting the restriction to Gauss distributions and permitting general, complex probability densities. In this way, one can incorporate very complex strategies in the control system, along with, in a very free manner, restrictions and constraints. These only have to be appropriately reproduced in probability distributions, which can then be treated, so to speak, as the preferred distributions of the system states. Here, however, problems can arise from the possible degeneration of the particle filter algorithm; the control strategy must be well designed in order to keep the degeneration as small as possible. Real-time controlling on the basis of nonlinear models is then quite feasible. As an example, Fig. 7 shows the control system

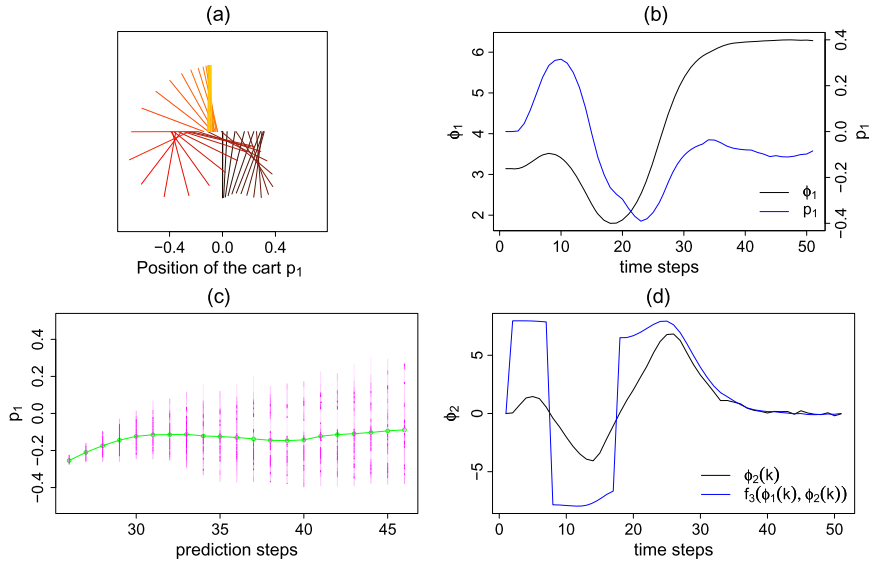

Fig. 7 Controlling a simulated inverted pendulum. (**a**) Movement of the pendulum over time, as indicated by color transition from black to red to yellow. *Black*: Start, with pendulum hanging below. *Red*: Swinging the pendulum into inverted position. *Yellow*: Inverted, balanced pendulum. The movement of the pendulum in the x-direction (cart movement) remains within narrow bounds. (**b**) Progression of the states p_1 (position, *blue*) and ϕ_1 (deflection angle, *black*) of the pendulum over time. (**c**) Prediction of the marginal distribution of the position p_1 (*magenta*) across the time horizon ($j = k, \ldots, k + T_p$ with $T_p = 21$) in the second particle filter at time-point $k = 25$. The *green line* describes the estimated average. (**d**) Progression of the state ϕ_2 (velocity of the deflection angle, *black*) in comparison with the specified progression of the corresponding set-point f_3 (*blue*), as a function of the deflection angle ϕ_1 and its velocity ϕ_2

for a nonlinear, inverted pendulum mounted on a cart that is sitting on a track. By moving the cart along the track, the pendulum is to be first swung into a vertical position and then held in this balanced state. As a supplementary constraint, the cart is not to drive across specified boundaries as it is moved along the track. It was possible to design the controls so simply that the computer-simulated system could be controlled in real-time (on a normal PC). Because the original, non-linearized model is used, both control tasks—swinging the pendulum into the vertical position and keeping it balanced there—can be accomplished with a single controller. More details can be found in [3].

5 Relationship to Simulation

As shown in the previous sections, one can estimate the hidden states of a process at prescribed measurement time-points by combining just a small amount of measurement data and a suitable mathematical system model. To do so, one performs a single system simulation step and then appropriately adapts the resulting calculated system state on the

basis of the new measurement information that has arrived during this time step. In addition to the actual process dynamic, another significant factor in choosing and implementing a state estimator is the computation time available for the algorithm and/or the level of real-time capability required for the application.

5.1 Requirements Relating to the Application

The amount of computation time available depends primarily on the application context of the state estimation. One must first decide whether it is to be performed online or offline, that is, in real-time or not.

For offline state estimations, the amount of computation time required is usually not a critical factor; the differential equation systems to be solved in the course of the simulation "simply" need to be solvable. The required computation time plays a subordinate role, since data acquisition and state estimation are not temporally linked to one another. This approach is frequently used when performing state estimations in connection with the identification of process parameters.

For online applications, however, such as safety systems, process monitoring and/or diagnosis, and process control, real-time capability is required for state estimating. Here, the real-time capability is assessed in relation to the updating time required for the state estimation. Depending on the process dynamic and the application, this can range from milliseconds to minutes. The requirements resulting from the three above-mentioned applications are discussed in the following sections:

- For critical situations in safety systems, the state estimation, the simultaneous process analysis for risk assessment, and the protective response trigger must be accomplished on the order of milliseconds. Even with very fast hardware systems, this speed is frequently impossible. Therefore, when developing such a system, the algorithms for risk analysis and protective response should be implemented on their own very high-speed hardware platform. For storage and diagnosis of events categorized as relevant by the safety system, one can then use a downstream monitoring system based on a state estimator on independent hardware. One can justify this separation, since, for system diagnosis, one is generally interested only in conspicuous and/or critical events and the upstream safety mechanism functions as a corresponding event detector.
- In addition to their downstream use in safety systems, independent monitoring systems are also frequently used for process behavior analysis or system diagnosis. Here as well, slight time delays in the analysis of the results are often permissible. In this case, one performs a delayed execution of the state estimation for measurements collected within a specified time window, while, in parallel, data for the next time window is being collected and stored. To prevent data loss, however, execution of both the state estimation and the diagnosis or evaluation algorithms must

be completed within the time required to store the measurement data. The ITWM's torque monitoring systems introduced in Sect. 6 also work according to this principle.

- In the course of process management and control, a controller must optimally adjust the performance of the process and, in particular, maintain process stability. In cases where there are no directly measureable performance variables, the optimal control inputs are calculated on the basis of the system state determined by a model-based state estimator. The execution of the state estimation must take into account the updating rate of the controller. Here, one must allow for the fact that additional computing capacity is needed for the controller to calculate the control inputs for the system. State estimation and calculation of the control signals must both take place within the available updating time. In connection with the BMWI project "Development of an energy-efficient furnace concept for the heat treatment of glass," this concept was used at the Fraunhofer ITWM on behalf of Schott AG to design a controller for the energy-efficient management of the glass cooling process. Here, in each time step, measurement information from a few air temperature sensors was used to estimate the temperature distribution in the entire furnace with a Kalman filter.

The computing time required for the state estimation depends on the time needed to perform the system simulation step and on the subsequent adaptation of the state. In many control engineering applications, system behavior is modeled by means of finite element approaches. Even at moderate resolution, these lead to high-dimensional models and, thus, to correspondingly long simulation times. Therefore, in many applications, regardless of the state estimator being used, one must initially reduce model complexity with mathematical model order reduction methods, so that the model-based state estimation is possible within the available computing time. The challenge with model order reduction is generating a less complex model that still approximates the dynamic of the real process in the relevant working areas as well as possible. The errors resulting from the model order reduction must then be accounted for in the state estimation in the form of process uncertainties. In this field, the Fraunhofer ITWM develops symbolic and numerical model reduction algorithms for parametric nonlinear systems.

Another component with a comparable impact on simulation time is the computational and storage capacity of the hardware platform being used. Along with the available working memory, the processing power must also be taken into consideration when implementing the selected filter. While today's typical PC processors or modern embedded systems exhibit no computation time problems for many applications when suitably reduced models are used, low cost processors with very low computing power and storage capacity are still often utilized for industrial mass-produced goods to save money.

5.2 Implementations at the ITWM

5.2.1 Linear State Estimators

Various linear state estimators have already been put to use at the Fraunhofer ITWM for diverse industrial projects and products. The starting point for the following treatment is a linear, time-variant state-space model of the form

$$\begin{aligned} \dot{x}(t) &= A(t)x(t) + B(t)u(t) + Q(t)w(t) \\ z(t) &= C_1(t)x(t) \\ y(t) &= C_2(t)x(t) + M(t)v(t) \end{aligned} \tag{21}$$

for the technical or biological process under consideration. Here, $A(t) \in \mathbb{R}^{n\times n}$ is the state matrix and $B(t) \in \mathbb{R}^{n\times q}$ is the input matrix, with which the measured system inputs $u(t)$ are assigned to the states and, where necessary, also converted into the needed physical quantities. The outputs of actual interest $z(t)$ are calculated from the states with the matrix $C_1(t) \in \mathbb{R}^{k\times n}$. These outputs can be any of the states themselves, that is, $k = n$ and $C_1 = I_{n\times n}$, or physical quantities converted from one or more states, such as torque calculated from the twisting angles (states $x(t)$) for a shaft. The outputs $z(t)$ thus calculated are frequently used as virtual sensors for process analysis or control. State estimation also requires a comparison of the simulation result with the real measurement information in order to calculate the correction terms. Therefore, one must determine from the model the physical quantities corresponding to the sensor measurement values. This is done by transforming the states $x(t)$ with the matrix $C_2(t) \in \mathbb{R}^{p\times n}$. Moreover, in (21), $w(t)$ and $v(t)$ are stochastic disturbances that are modeled by the matrices $Q(t)$ and $M(t)$ and which impact the system and/or the measurements. Selection of the appropriate state estimator now depends on the assumptions made and/or on the process characteristics.

Discrete Kalman Filter Due to its simple iterative calculation scheme, the linear discrete Kalman filter, along with its diverse variations, is the most widely used algorithm for state estimations of linear systems (see Sect. 4.8). In particular, it can also be used in cases of non-steady-state noise or with time-variant systems. In order to apply it to the continuous model being treated here (21), one must first discretize it, reduce it to an appropriate dimension, and implement the iterative, discrete Kalman filter specified in Sect. 4.8. One such application at the ITWM involved the aforementioned BMWI project "Development of an energy-efficient furnace concept for the heat treatment of glass," where we estimated the temperature distribution in a passive glass annealing furnace on the basis of the time variance of the underlying linear model.

Continuous-Time Filter The Kalman-Bucy filter is the continuous form of the Kalman filter. Analogously to the discrete Kalman filter, the disturbances are handled purely by means of the expectation value and covariances, so that one can assume $M(t) = I$ for the continuous state-space system (21) without any restrictions. Here, we let $w(t)$ represent

normally-distributed process noise with expectation value 0 and covariance matrix $R_w(t)$, and we let $v(t)$ represent normally-distributed measurement noise with expectation value 0 and covariance matrix $R_v(t)$.

Using the previous assumptions, the continuous Kalman filter is given by

$$\dot{\hat{x}}(t) = A(t)\hat{x}(t) + B(t)u(t) + K(t)\big(y(t) - C_2\hat{x}(t)\big) \tag{22}$$

where $K(t) = P(t)C_2(t)^T R_v^{-1}(t)$ and $P(t)$ is the solution of the differential equation

$$\dot{P} = A(t)P + PA(t)^T - PC_2^T R_v^{-1}(t)C_2(t)P + QR_w(t)Q^T \tag{23}$$

with $P(0) = E\{x(0)x^T(0)\}$.

Due to the time-dependency of the noise and the time-variance of the model, calculating the solution of the differential equation (23) for each time-step is very computationally intensive and, in many cases, cannot be done online. This is not so for a time-invariant system, that is, one in which the state matrices A, B and C_2 in (21) are constant. For the underlying process, one frequently assumes steady-state measurement and process noise, along with time-invariance. In this case, the Kalman gain K converges to a constant matrix and is given by

$$K = PC_2^T R_v^{-1},$$

where P is the stabilizing solution of the Riccati equation

$$AP + PA^T - PC_2^T R_v^{-1}C_2P + QR_wQ^T = 0. \tag{24}$$

The resulting filter can then be represented as a linear state-space system, with (A_{KF}, B_{KF}, C_{KF}) given by

$$\begin{aligned} A_{KF} &= A - PC_2^T R_v^{-1}C_2 \\ B_{KF} &= \begin{bmatrix} PC_2^T R_v^{-1} & B \end{bmatrix} \\ C_{KF} &= C_1. \end{aligned} \tag{25}$$

To implement this filter, one can now perform a single *a priori* offline calculation of the Kalman gain and the error covariance estimation before the actual state estimation. Then, one determines the desired states online for each time-step by solving the differential equation system. Here, one can either discretize the system *a priori* or calculate the solution of the differential equation system stepwise using a suitable algorithm. This separation into

offline and online calculation steps makes implementation possible even for short updating times.

As described in Sect. 4, the Kalman filter offers, in the form of the error covariance matrix, a confidence measure for the estimated states under the assumed stochastic influences. From a systems theory perspective, the Kalman filter is the state estimator that delivers optimal estimates, in the sense that it averages across all frequencies. An estimator that focuses on critical frequencies is the H_∞-filter [48], which is based on the H_∞-norm. For a well-defined, stable, time-invariant system G, this is defined by

$$\left\| G(s) \right\|_\infty := ess \sup_\omega \bar{\sigma}\left(G(j\omega)\right)$$

with the maximum singular value

$$\bar{\sigma}\left(G(j\omega)\right) := \max_{u(\omega)\neq 0} \frac{\|z(\omega)\|_2}{\|u(\omega)\|_2}$$

and $z(\omega) = G(j\omega)u(\omega)$. For linear systems with one input and one output, the norm describes the maximum gain factor across all frequencies. Choosing the ∞-norm shifts the focus from the simultaneous minimization of the energy of the transfer functions for all frequencies to the most critical frequency of the system. In other words, it deals with a worst-case scenario.

As the starting point for calculating the filter, we take a time-invariant linear state-space system; that is, the matrices A, B, C_1, and C_2 in (21) are constant. With H_∞-filter problems, one assumes that the disturbances have limited energy. The actual information about the intensity of the disturbances is captured by the time-invariant matrices in (21), Q and M [2]. On the basis of the H_∞-norm, the H_∞-filter problem can be formulated as follows [48]:

H_∞-Filter Problem For a given $\gamma > 0$, find a causal filter $F(s) \in \Re H_\infty$, where $\Re H_\infty$ is the set of all well-defined and real-rational, stable transfer functions, so that

$$\sup_{w\in L^2[0,\infty)} \frac{\|\tilde{z} - \hat{z}\|_2^2}{\|w\|_2^2} < \gamma^2.$$

Here $\tilde{z}$ denotes the real system output and $\hat{z}$, the estimated filter output.

One then obtains the desired gain-matrix for a linear, robust H_∞-filter by solving the following algebraic Riccati equation:

$$PA^T + AP + P\left(\gamma^{-2}C_1^T C_1 - C_2^T M M^T C_2\right)P + QQ^T = 0. \tag{26}$$

If the positive, semi-definite stabilizing solution P exists, then the desired filter $F(s) \in \Re H_\infty$ can be represented in the state-space representation and the system matrices (A_{HF}, B_{HF}, C_{HF}) are given by

$$\begin{aligned} A_{HF} &= A - PC_2^T \left(MM^T\right)^{-1} C_2 \\ B_{HF} &= \left[PC_2^T (MM^T)^{-1}\ B\right] \\ C_{HF} &= C_1. \end{aligned} \tag{27}$$

The difference between the Riccati equation (26) and the Riccati equation (24) for calculating the Kalman gain is essentially the additional term $\gamma^{-2} C_1^T C_1$ resulting from the robustness requirement. Here, the existence of a solution to the Riccati equation (26) is not guaranteed for each γ. To obtain the most robust estimator possible, γ is iteratively reduced until the solution of the algebraic Riccati equation exists. Therefore, as with the Kalman filter (25), implementation of the robust linear H_∞-filter (27) also involves first solving an algebraic Riccati equation offline for the underlying linear, time-invariant state-space model. The resulting H_∞-filter is also given in the form of a dynamic continuous-time state-space system (27). However, minimizing the H_∞-norm results in more robustness relative to unstructured disturbances and/or model uncertainties than exists for the Kalman filter. The μ-synthesis, also used at the ITWM, delivers extensions relating to structured uncertainties. The decision whether to use the H_∞-filter or the Kalman filter depends on the model uncertainties and the resulting robustness requirements.

The torque detection and analysis system TorAn described in Sect. 6 is an ITWM monitoring system based on an online-capable, robust H_∞-filter or the continuous Kalman filter. On the basis of the measured mechanical torque signals of the energizing drive train components, such as the motor or generator torque, and a direct torsion measurement with a torque sensor, TorAn uses the selected filter to estimate the states given in the form of the twisting angle of the rotating shaft. TorAn also uses the estimated states to predict and analyze the torque characteristics for other critical, inaccessible shaft components. The algorithms for data acquisition and state estimation and the criteria monitoring and fatigue analysis were implemented as a real-time-capable C-Library with a link to an analog-digital converter.

5.2.2 Nonlinear State Estimator

Constrained Extended Kalman Filter The extended Kalman filter is the extension of the discrete Kalman filter to nonlinear systems. As with the previously described linear state estimator, these filters do not initially allow for consideration of physical constraints. However, there are some approaches that do make this possible, such as the Moving Horizon Estimation or the Constrained Extended Kalman Filter (CEKF) proposed in [47]. The

basic idea of the CEKF is to initially perform a general state estimation with a first extended Kalman filter. Then, in a second extended Kalman filter, a correction to the first estimation is undertaken so that the states lie within the permissible value range. Particularly for nonlinear systems, one can frequently limit the states' solution space by means of physically motivated restrictions and, thus, prevent a possible divergence in the state estimation. The ITWM uses this filter to compensate rpm-dependent, periodic disturbances in connection with torque measurements using an inductive torque sensor based on the magnetostrictive effect (see Sect. 6.2).

Particle Filter Algorithm As described in detail in Sect. 4, the standard particle filter works on discrete-time, nonlinear, non-Gaussian models and can be easily adapted for use with continuous-time systems with discrete-time measurements. The particle filter algorithm has already been used in diverse application areas in the form of prototype implementations. Among the applications are the measurement-based model identification of a shock absorber with hysteresis effects, the estimation of gene copies (copy number) in tumor DNA, and the estimation of parameters in the biomedical field (see Sect. 7). There are implementations in Java and in the statistical programming language R, which is often used in biomedicine. In addition, the particle filter algorithm has been implemented as an element of a system-biology toolbox being developed at the Fraunhofer Chalmers Centre (FCC) in Gothenburg on the basis of the symbolic programming environment Mathematica. Extensions to the particle filter algorithm have also been developed at the ITWM, in particular, the adaptation of the algorithm to uncertainties in measurement time-points and its double-use in a new nonlinear Model Predictive Control (MPC) approach.

6 Online Monitoring of Torsional Vibrations in Power Plant Turbine Generator Shaft Lines

6.1 Problem Description

The first of what have become many endeavors involving state estimation in the System Analysis, Prognosis, and Control Department began with a project to develop a state observer for power plant turbine sets. These turbine sets consist of a long shaft on which are mounted a generator for electricity generation and one or more turbines to drive the shaft (see Fig. 8). Grid malfunctions or operational errors can trigger torsional vibrations in a turbine set, which lead to fatigue in the shaft components and may even result in serious mechanical damage. In some cases, the latter can induce additional, permanent torsional vibrations at the rotational frequency and its harmonics. It is therefore necessary to ensure continuous monitoring of turbine generator shaft lines for torsional vibrations. For many years, the Fraunhofer ITWM has worked to develop methods for online monitoring of torsional vibrations and has developed procedures for model-based prognosis of torsional vibrations and run-out compensation of inductive magnetostrictive sensors. These results

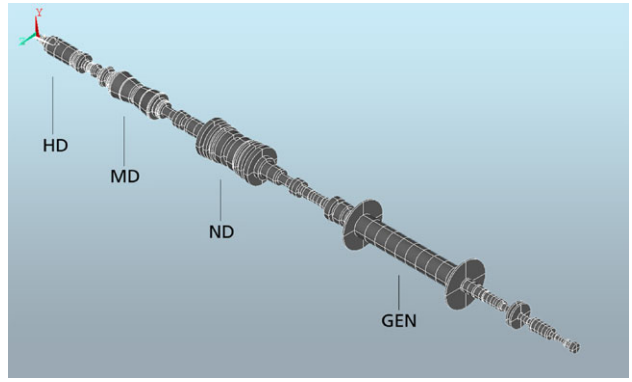

Fig. 8 Schematic of a power plant turbine set with generator (GEN), low-pressure turbine (ND), intermediate-pressure turbine (MD), and high-pressure turbine (HD)

have been put into service around the world by such customers as E.ON Anlagenservice, Siemens Energy, and ABB Utilities.

The challenges associated with the torsion monitoring of power plant turbine generator shaft lines are diverse. The methods and products developed at the Fraunhofer ITWM to meet these challenges include systems for the following:

- Experimental torsional analysis with which, for example, torsional natural frequencies from turbine generator shaft lines can be determined (TorStor),
- Measurement-data-based detection and assessment of critical torsional vibrations, such as sub-synchronous oscillations (TorFat),
- Targeted monitoring of at-risk locations in a turbine set—the shaft couplings, for example—using model-based state estimators (TorAn), and
- Detection and classification of shaft damage.

All systems require torque measurements from at least one position on the drive train. Here, one can make use of the magnetostrictive effect, which describes the relationship between the magnetic permeability change and a change in strain when external loads are applied. The measurement systems function with no shaft contact and can be positioned flexibly. On the basis of the inverse magnetostrictive effect of ferromagnetic shafts, they detect changes in the torsional stress on the shaft surface by means of induction and magnetic field measurements. Thus, they represent a useful alternative to traditional torque sensing technology. The great advantage of these sensors is that their use requires no structural modifications to the shaft itself. They can therefore be flexibly implemented without influencing the dynamics of the shaft. The magnetostrictive sensor developed on behalf of the Fraunhofer ITWM for torsion monitoring in power plants inductively measures the difference in magnetic permeability, which is proportional to the torsional stress on the shaft surface across a large measurement range. Here, a primary coil in the center

of the measurement head is excited with high frequency alternating current, thus producing a magnetic field. The magnetic field passes through the air gap between sensor and shaft and penetrates the shaft's surface. Depending on the magnetic permeability, the field spreads out over the shaft surface and is assessed by four measurement coils within the sensor head, which are positioned at a 45-degree angle to the main axis of the shaft. With this measurement arrangement, the signal resulting from the measurement coil circuitry is proportional to the torsional stress on the shaft surface. The sensor's output voltage/current $S_V(t)$ is converted into torque $S_M(t)$ by means of

$$S_M(t) = gain * \big(S_V(t) - offset\big).$$

The quantities *offset* and *gain* needed for the conversion must be determined in a calibration step using measurements from two known load points. After calibration, the sensor can then be used for torque measurements.

6.2 Run-out Compensation Using the Constrained Extended Kalman Filter

In addition to the torsional stresses resulting from external loads, there are always permanent, frozen stresses on the shaft surface that arise during the manufacturing process. These so-called inhomogeneities vary locally, and it is impossible to make any general *a priori* statement about their shape and size. Therefore, when performing torsional measurements on a rotating shaft with an inductive magnetostrictive sensor, one must take into consideration that these inhomogeneities along the measurement track lead to varying magnetic flows around the entire circumference. However, the rotation of the shaft causes the signals resulting from the inhomogeneities along the measurement track around the shaft circumference to always recur in the same sequence. Thus, one obtains a deterministic, periodic disturbance signal $y_{Inhom}(t)$, referred to as the run-out signal. Because one is dealing with localized characteristics distributed around the circumference, the frequencies of the various elements of the disturbance signal correspond to whole-numbered multiples of the shaft's rotational frequency $f(t)$. However, for state evaluations of industrial turbines, for example, it is often exactly these frequencies that are of interest. Thus, the run-out masks the significant system information, and determinations of the torsional load made without signal correction always contain errors. Due to their characteristics, run-out signals can be modeled with the time-dependent rotational frequency $f(t)$ as the basic frequency of a Fourier sum at time t_k as follows:

$$y_{Inhom}(t_k) = \sum_{l=1}^{n} a_l(t_k) \sin\big(l2\pi f(t_k)t_k\big) + b_l(t_k) \cos\big(l2\pi f(t_k)t_k\big). \tag{28}$$

The amplitudes $a_l(t_k)$ and $b_l(t_k)$ do not change relative to a fixed reference point on the shaft's circumference, even when the rotational speed varies. For a non-changing measurement track, they can be assumed to be constant. When the shaft is loaded with an

external torque $y_T(t_k)$, the disturbance signal is then superimposed on the torque signal. The measurement noise of the measurement system $v(t)$ must also be considered, so that the measurement signal $y_{Mess}(t_k)$ of the torque sensor at time t_k thus becomes

$$y_{Mess}(t_k) = y_T(t_k) + y_{Inhom}(t_k) + v(t_k). \tag{29}$$

In some applications, one can resort to classical signal filters, such as high-pass, low-pass, or notch filters, to compensate the run-out signal. In the context of detecting critical torsional frequencies for turbine sets, for example, one encounters mostly an excitation of the critical excitable torsional natural frequencies, which do not correspond to the shaft rpm and its harmonics for standard grid operation. By means of frequency-selective analysis or band-pass filtering of the relevant frequency zones, one can therefore isolate the torsional vibrations of interest $y_T(t_k)$ from the measurement signal $y_{Mess}(t_k)$ for subsequent analysis. If a disturbance frequency should correspond to one of the critical torsional frequencies of interest, however, these torsional vibration signals would either be completely eliminated with the aforementioned filter, or be at least strongly distorted. Therefore, one should use a filter that eliminates the run-out signal, but leaves the actual relevant vibration information unchanged—also for the shaft rpm and its whole-numbered multiples.

In a compensation method developed at the ITWM, the run-out signal $y_{Inhom}(t_k)$ is estimated online relative to the circumference for each time-step t_k and subtracted from the measurement value $y_{Mess}(t_k)$. Here, however, the influences of all transfer functions, such as phase shifts and amplitude attenuations from signal filters, must be taken into consideration within the measurement chain.

The parameters to be determined in the run-out function (28) are described by the dynamic discrete state-space model

$$x(t_{k+1}) = Ax(t_k) + w(t_k) \tag{30}$$

$$y(t_k) = h\big(x(t_k)\big) + v(t_k) \tag{31}$$

with $x(t_k) = [a_0(t_k), \ldots, a_n(t_k), b_1(t_k), \ldots, b_n(t_k), f(t_k), \Theta(t_k)]^T \in \mathbb{R}^{2n+3}$ and

$$A = \begin{bmatrix} I_1 & 0 & 0 & 0 \\ 0 & I_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 2\pi\Delta t & 1 \end{bmatrix} \in \mathbb{R}^{(2n+3)\times(2n+3)}$$

where $I_1 \in \mathbb{R}^{(n+1)\times(n+1)}$, $I_2 \in \mathbb{R}^{n\times n}$ are unit matrices. Moreover, $w(t_k)$ and $v(t_k)$ in Eqs. (30) and (31) are normally-distributed, white process and measurement noise, respectively, while $h(x(t_k))$ represents the use of the states $x(t_k)$ in Eq. (28) and the subsequent signal filtering. The states $x(t)$ are estimated by a mathematical state observer online from the measurements by comparing $y_{Mess}(t_k)$ and $y(t_k)$. Here, in particular, constraints resulting from the technical and physical boundary conditions of the measurements are taken into account in the form of upper and lower bounds for each physical quantity. As

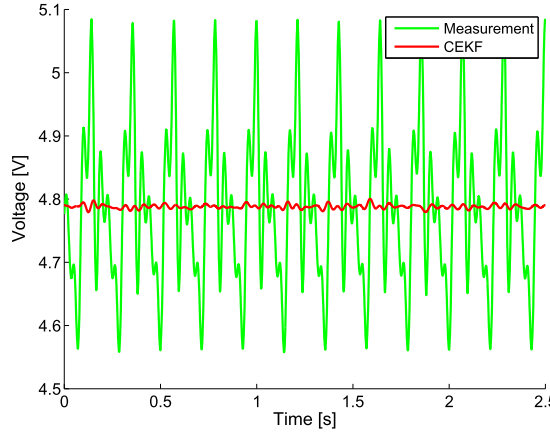

Fig. 9 Run-out compensation result with Constrained Extended Kalman Filter (CEKF)

the state estimator, the Constrained Extended Kalman Filter (CEKF) proposed in [47] was adapted to the given run-out compensation problem. The CEKF allows one to distinguish between the run-out signal and torsional vibrations, even when the torsional natural frequency corresponds to the rotational frequency. Figure 9 shows the results of a CEKF run-out compensation for a measurement with constant load. After the compensation, one obtains the load value with sensor noise, which in this case is less than 0.5 %. Along with the run-out compensation, the method also estimates the rotational frequency $f(t_k)$ and the current position $\Theta(t_k)$ of the shaft in relation to the sensor. Both this and other methods have been used successfully for run-out filtering on magnetostrictive torque measurements in industrial installations.

6.3 Prognosis of Torsional Vibrations for Inaccessible Components on a Turbine Generator Shaft Line

During state monitoring of torsional vibrations on a power plant turbine generator shaft line, one must be able to guarantee uninterrupted surveillance of the drive train's critical components. However, technical restrictions and cost concerns sometimes prevent placement of torque sensors on all the critical shaft components one would like to observe. In order to nonetheless be able to make a statement about the torsional oscillations and their influence on any given shaft component, one must use a suitable, model-based prognosis system. Older torsional oscillation monitoring systems are based on a pure system simulation, in which modeling errors, estimated initial conditions, and model uncertainties during the simulation can lead unavoidably to deviations from true system behavior. An overview of the existing systems can be found in [35]. Use of a mathematically robust, online-capable state estimator represents an extension of the pure simulation approach. This approach was first introduced by the Fraunhofer ITWM in collaboration with the Electrical Drives and Mechatronics Chair of the Technical University of Dortmund, under the supervision of Professor Stephan Kulig, for the torsion monitoring of power plant

turbine generator shaft lines. Along with the measurement data needed for the simulation, the state estimator receives a torque measurement for one component as an additional input quantity. The mathematical state estimator then implicitly compares the real, measured data with the time-series predicted by the simulation for the measurement site. Information obtained on the difference between the measured torque signal and the system simulation is integrated—as described in Sect. 5—into the prediction of the torsional vibration behavior of the remaining components in the form of a correction term for error compensation. Depending on the quality of the available physical information, either a Kalman filter or a robust H_∞-filter (see Sect. 5.2.1) is used for the torsion monitoring. The filter design is accomplished at the ITWM by following the steps outlined below.

On the basis of the geometric and physical information available for the given drive train, the finite element method is used to generate the Newtonian equations of motion—a system of 2nd order ordinary differential equations—for the torsional behavior of the drive train:

$$J\ddot{\varphi}(t) + K\varphi(t) = \bar{B}u(t); \qquad y(t) = \bar{C}\varphi(t). \tag{32}$$

Here, $0 < J^T = J \in \mathbb{R}^{n\times n}$ is the matrix of the mass moment of inertia and $0 \leq K^T = K \in \mathbb{R}^{n\times n}$, the torsional stiffness matrix of the system. Moreover, $\bar{B} \in \mathbb{R}^{n\times q}$ is the input matrix, which contains information about the position and conversion factors for the externally applied torques of the generator and the turbines $u(t)$. The matrix $\bar{C} \in \mathbb{R}^{p\times n}$ transforms the angular displacements $\varphi(t)$ into torques $y(t)$ for the targeted system components, for example, the coupling between the turbine trains. The torque at the sensor's measurement site is another output, which is then used in the state estimators to compare simulation and measurement. One knows that the matrices J and K can be diagonalized by an equivalence transformation with the modal matrix $V = [v_1 \cdots v_n]$ and a suitable norming of the modes $v_i \in \mathbb{R}^n$, $i = 1, \ldots, n$. This then yields:

$$V^T J V = I, \qquad V^T K V = \Lambda, \tag{33}$$

where $I \in \mathbb{R}^{n\times n}$ is the unit matrix. The diagonal matrix $\Lambda \in \mathbb{R}^{n\times n}$ contains the generalized eigenvalues of the undamped system, that is, $KV = JV\Lambda$.

To account for the damping neglected up to this point in Eq. (32), one usually has only rough approximations regarding the modal damping. Therefore, Eq. (32), as will be now demonstrated, is first modally transformed and then augmented by a modal damping term $D\dot{x}(t)$ with the modal damping matrix $D \in \mathbb{R}^{n\times n}$. Because of this simplification, and also because the modal damping coefficients are usually not known exactly for turbine generator shaft lines, one must treat these as model uncertainties during the subsequent filter design. With these damping assumptions, and with the substitution of $\varphi(t) = Vx(t)$, one obtains, taking into account the specified orthogonality conditions, the following system of decoupled 2nd order differential equations:

$$\ddot{x}(t) + D\dot{x}(t) + \Lambda x(t) = V^T \bar{B}u(t); \qquad y(t) = \bar{C}Vx(t) \tag{34}$$

If one now also substitutes $z(t) = \left[\begin{smallmatrix} x(t) \\ \dot{x}(t) \end{smallmatrix}\right]$ in (34), one obtains the following state-space model:

$$\begin{aligned} \dot{z}(t) &= Az(t) + Bu(t); \\ y(t) &= Cz(t); \end{aligned} \tag{35}$$

with

$$A = \begin{bmatrix} 0 & I \\ -\Lambda & -D \end{bmatrix} \in \mathbb{R}^{2n \times 2n}, \qquad B = \begin{bmatrix} 0 \\ V^T \bar{B} \end{bmatrix} \in \mathbb{R}^{2n \times q} \quad \text{and}$$
$$C = \begin{bmatrix} \bar{C} V & 0 \end{bmatrix} \in \mathbb{R}^{p \times 2n}.$$

In order for the state estimator to achieve real-time capability, that is, in order to be able to calculate one time-step of the state estimation within the real time sampling interval, the dimension of the state-space model (35) is reduced using a model order reduction method. Here, the system model is transformed using the appropriate projection matrices $T_R, T_L \in \mathbb{R}^{n \times s}$ with $s \ll n$. This yields

$$\begin{aligned} \dot{z}_r(t) &= A_r z(t) + B_r u(t) \\ y(t) &= C_r z_r(t); \end{aligned} \tag{36}$$

with

$$z(t) = T_R z_r(t), \qquad A_r = T_L^T A T_R, \qquad B_r = T_L^T B, \qquad C_r = C T_R.$$

Especially for the reduction of weakly damped, 2nd order systems—as we have with the torsion model for power plant turbine shaft lines —the Fraunhofer ITWM has developed efficient methods for an approximated, frequency-weighted, balanced reduction [9]. With regard to the quality of the approximation, special emphasis is placed here on specifying a frequency range in advance. For the torsion monitoring of turbine generator shaft lines, this is the low-frequency range from 0 to 200 Hz, since most malfunctions in the electrical grid result primarily in an excitation of the torsional natural frequency of the power plant turbine shaft line below the grid frequency.

When designing the state estimator, one must then consider both the above-mentioned modeling assumptions and also the uncertainties resulting from the model reduction. Therefore, one adapts the model using suitable methods based on measurements, such as the modal data of the real system. Here, one must consider that the torsional natural frequencies cannot be excited experimentally at will for power plant turbine generator shaft lines. Thus, the modal data must often be determined from actual grid disturbances by means of permanent monitoring and analysis. Moreover, one only has torsion measurements for a few shaft components. All told, one has usually measured only a few natural frequencies from the low-frequency range, and measurement values are available

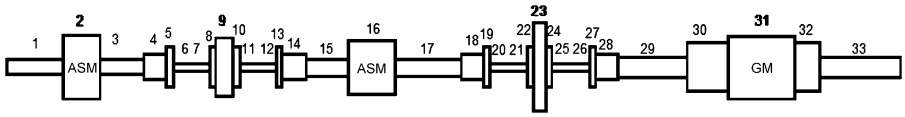

Fig. 10 Schematic of the test rig showing the division into finite elements

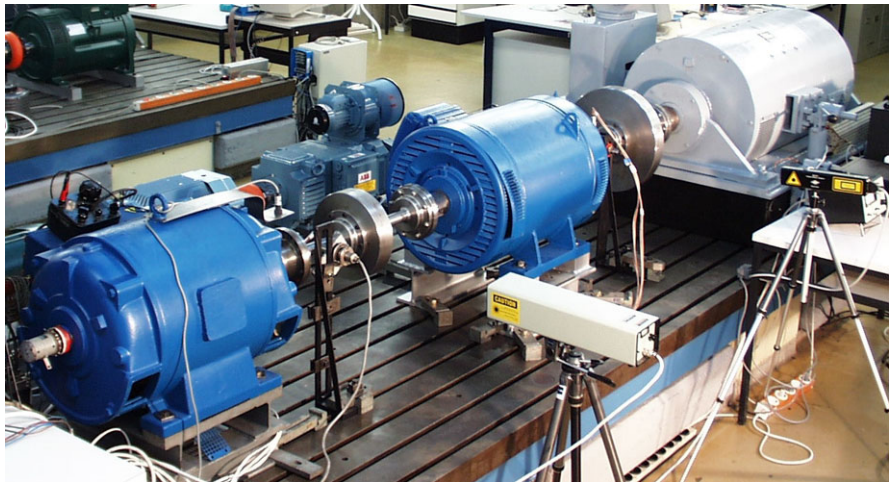

Fig. 11 Torque test rig (©Chair Electrical Drives and Mechatronics, TU Dortmund)

for few of the nodes for the corresponding modes. In contrast, the analytical model contains significantly more natural frequencies, depending on the model dimension. However, many of these "extra" natural frequencies are in the high-frequency range. To the extent that one has modal information, that is, measurements of natural frequencies and modes, one then uses model updating methods to improve model quality. Using the iterative model updating methods [10] developed for this problem at the Fraunhofer ITWM, the model parameters are adapted so as to achieve the desired correspondence between the model's modal data and the measured data. The reduced and adapted model then serves as the basis for the filter design (e.g., Kalman filter or H_∞-filter; see Sect. 5.2.1), which is then used in the monitoring system TorAn for the state estimation and torsional vibration prognosis. Extensive descriptions of the design steps outlined here can be found in [4–6].

The functionality of the prediction of the mathematical, robust state observer for torsional vibration prognosis is demonstrated using the example of the test stand from the Electrical Drives and Mechatronics Chair of the TU Dortmund. In contrast to a real power plant turbine generator shaft line, it was a simple procedure to install extra sensors here to assess the quality of the state estimator. Figure 10 shows a schematic diagram of the test rig; Fig. 11, a photo; and Fig. 12, the torque sensor positioned at node 7. The test rig was designed to exhibit the typical natural frequency and mode data of a turbine generator

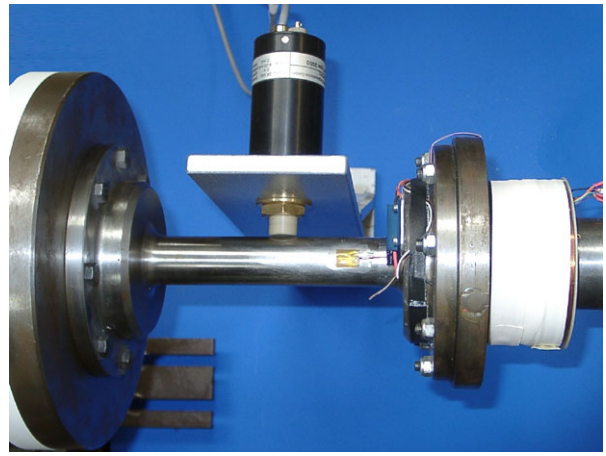

Fig. 12 Contact-free torque sensor on element 7 (©Chair Electrical Drives and Mechatronics, TU Dortmund)

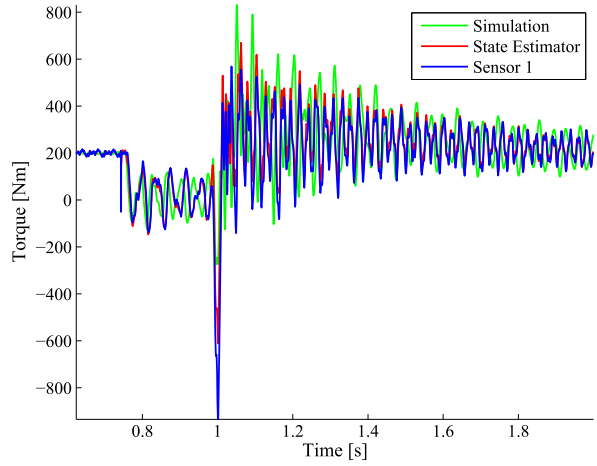

Fig. 13 Comparison of measurement (*blue*), state estimation (*red*), and pure simulation (*green*) at node 7

shaft line, and thus a similar dynamic. According to the previously described modeling steps, an online-capable, robust H_∞-filter was designed for the test rig. As measurement quantities, that is, as input for the filter, we used the mechanical moment of the asynchronous drive machine (obtained via power and rpm measurements)—node 2—and the DC machine—node 31—along with a measurement from the contact-free torque sensor—node 7. As explained earlier, the state observer, unlike a pure simulation, uses the comparison of measurement signal and simulation as a central quantity for determining the torque estimates.

For the disturbance scenario referred to as a "short interruption," the torsional vibrations for nodes 7 (Fig. 13) and 21 (Fig. 14) were predicted on the basis of the pure finite element model and also using the filter generated from this model. In order to be able to assess the results of the pure system simulation and the state estimation, the torsional vibrations on node 21 were measured with another sensor. Unlike the measurement from node 7,

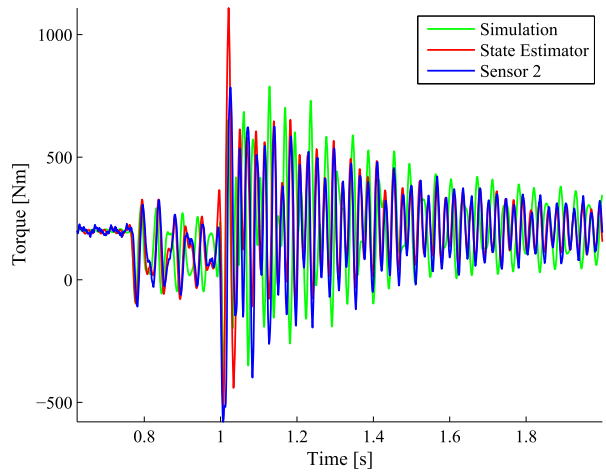

Fig. 14 Comparison of measurement (*blue*), state estimation (*red*), and pure simulation (*green*) at node 21

the measurement from node 21 is not used as an input quantity for the state estimator. The correspondence between measurement and state estimation is, as expected, clearly better than that between system simulation and measurement. The cause of the overall poor correspondence for the system simulation is the erroneous modeling of the drive train damping. With the state estimation, this leads to short-lived errors at the beginning of the short interruption. However, this prediction error is corrected by the filter within one to two cycles, which highlights the capability of a robust mathematical state estimator to compensate for model uncertainties.

7 Application of the MTU Particle Filter to a Plasma-Leucine Model with Population Data

In this section, we describe an application of the MTU-PF algorithm, a version of the particle filter developed at the ITWM that allows for inclusion of uncertainties in the measurement time-points (MTU stands for Measurement Time Uncertainties; see Sect. 4.11). The results are based on a collaboration between the ITWM and Mats Jirstrand, from the Fraunhofer Chalmers Center (FCC) in Gothenburg, Sweden, Martin Adiels, from the Sahlgrenska Center for Cardiovascular Research in Gothenburg, and Marja-Riitta Taskinen, from the medical faculty of the University of Helsinki, Finland [1, 8]. In this project, we apply the MTU-PF to a study that analyzes the kinetics of the amino acid leucine in blood plasma by means of so-called tracer/tracee experiments. This plasma-leucine is a component of certain lipoproteins, which serve as fat transporters in blood and play an important role in cardiovascular disorders. Specifically, in the course of our project, we were asked to confirm a hypothesis about the deviation of a rate parameter for diabetes patients, in comparison with test subjects from a control group. The difficulties in performing the corresponding estimates result, on the one hand, from the assumption of hierarchically arranged parameters (global, group-specific, individual parameters), which lead to so-called

mixed effects in the models, and, on the other hand, from uncertainties in the measurement values (blood samples), missing measurements, and, in particular, from uncertainties in the measurement time-points.

Tracer/tracee experiments were carried out in the study to analyze the plasma-leucine kinetics. The kinetics of the actual substance of interest—plasma-leucine—in this case referred to as the tracee, are determined by observing a labeled leucine added in the experiment, referred to as the tracer. The underlying model for the plasma-leucine kinetics comes from Demant et al. [26] and is based, in turn, on Cobelli et al. [23]. The data is taken from a clinical study on diabetes patients [12, 13]. In [1] and [8], both model and data are used to perform a Bayesian population-based parameter estimation, and were already used earlier for a maximum likelihood estimation [19]. In this case, the original model, based on ordinary differential equations (ODE), had to be supplemented with a stochastic element, with the result that the kinetics are now modeled using stochastic differential equations (SDE). The approach in [19] differs from ours also in that it assumes the stochastic fluctuations to be in the plasma-leucine (tracee), whereas we place the variability in the labeled leucine (tracer). In point of fact, stochastic variability should be assumed for both concentrations; for simplicity's sake, however, we limit ourselves to just one.

The negative effects on the estimates of uncertainties or inaccuracies in determining the measurement time-points are to be expected primarily at the beginning of the measurement series, since it is here, directly after addition of the tracer, that the concentrations change most abruptly. Our algorithm has the ability to counteract this problem.

7.1 The Leucine Model

In [13] (see [11] also), a new, combined multi-compartmental model for apolipoprotein B-100 (apoB) and triglyceride metabolism in very low density lipoprotein (VLDL) subfractions was developed (see Fig. 15). VLDL serve as transporters of triglycerides and cholesterol from the liver to the periphery. Elevated values are associated with an increased risk of cardiovascular disorders. Each VLDL particle contains exactly one apoB molecule, which makes apoB a suitable marker for triglyceride transport. The secreted particles become denser and denser as more triglycerides are delivered to target sites, such as muscles and adipose tissue, so that the relative protein content increases. As the density increases, the VLDL becomes an intermediate density lipoprotein (IDL) and, finally, a low density lipoprotein (LDL).

For our purposes, we use only the portion of the model that concerns the leucine pool, that is, compartments 1–4 (see Fig. 16). The fluxes exiting the subsystem are located in compartments 1 and 2. The flux entering compartment 1 is designated U_1.

The data is obtained from tracer/tracee experiments. Here, the tracee (i.e., the concentration we are actually interested in) consists of the leucine amino acids as components of the apoB molecule. Additional, labeled leucine (the tracer) is injected as a bolus infusion. Knowledge about the kinetics (fluxes between the compartments) of the tracee can be gained by studying the kinetics of the tracer.

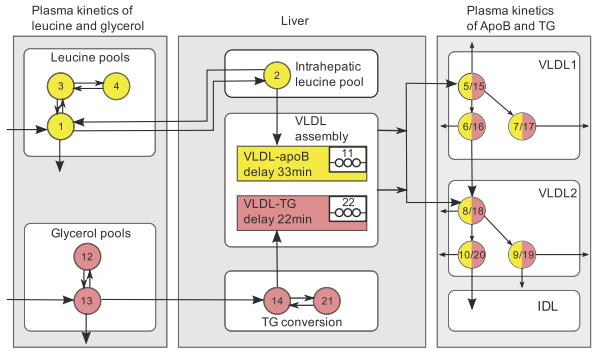

Fig. 15 Multi-compartmental model for the metabolism of apolipoprotein B-100 (apoB) and triglycerides (TG) in very low density lipoprotein (VLDL) subfractions. This multi-compartment model was developed in [13]

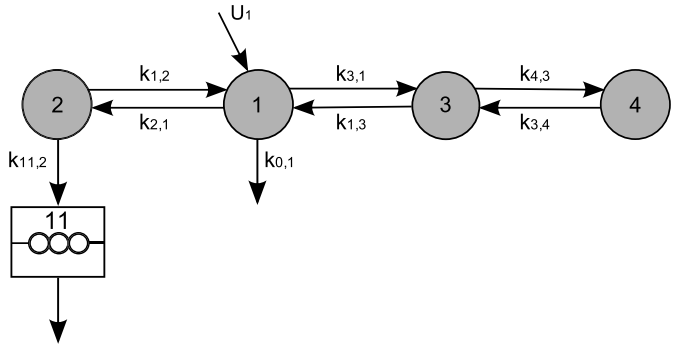

Fig. 16 Schematic depiction of the restricted model (leucine pool) [11]. This scheme is a sub-scheme of Fig. 15. *Circles* depict compartments. *Arrows* depict fluxes between compartments and are labeled with the corresponding fractional transfer coefficients. Compartment 1 is the plasma-leucine compartment, into which the leucine is injected. Compartment 2 is an intrahepatic compartment and source of the apoB synthesis. Compartments 3 and 4 are body protein pools. The output is from compartment 1. Compartment 11 is a delay compartment, used here only as an output from compartment 2

For each compartment i, with $i = 1, \ldots, 4$, Q_i and q_i now refer to the concentrations of tracee and tracer, respectively. Similarly, U_i and u_i refer to the input of tracee and tracer, respectively. For tracer/tracee experiments, a steady-state is generally assumed for the concentration Q_i of the tracee. If the concentration of the labeled injection is small compared with the overall concentration levels, and if the model is linear, than the following holds approximately:

$$\frac{\mathrm{d}q(t)}{\mathrm{d}t} = K(t)q(t) + u(t)$$

with $q(t) = (q_i(t))_{i=1,2,3,4}^{\mathrm{T}}$, $u(t) = (u_1(t), 0, 0, 0)^{\mathrm{T}}$, and

$$K(t) = (k_{j,i})_{j,i=1,2,3,4}.$$

Here, $k_{j,i}$ for $i \neq j$ is the transfer coefficient of the tracer from compartment i to compartment j. Compartment 0 is, in general, the output compartment (not shown in the figures). Here, compartment 11 is also considered to be an output compartment. Moreover, for each $i = 1, \ldots, 4$,

$$k_{i,i} := - \sum_{\substack{j=0,1,2,3,4,11 \\ j \neq i}} k_{j,i}.$$

As time unit, we always assume 1 hour (h); all transfer coefficients are given in the unit h^{-1}, and the amount of material in the compartments, in mg. In our model, only $k_{0,1}$, $k_{1,2}$, $k_{1,3}$, $k_{2,1}$, $k_{3,1}$, $k_{3,4}$, $k_{4,3}$, and $k_{11,2}$ are assumed to be non-zero, while the following dependencies between the transfer coefficients are also assumed to be valid:

$$k_{1,2} = k_{2,1},$$
$$k_{3,4} = 0.1 \cdot k_{4,3}.$$

The transfer coefficient $k_{11,2}$ must be fixed and specified in order for the system to be identifiable. We set $k_{11,2} = 0.01\ \mathrm{h}^{-1}$ as an estimated average from earlier measurements. We generate stochastic differential equations (SDE) based on the resulting ordinary differential equations by adding stochastic noise terms. These are given by standard Wiener processes $\mathscr{W}_{1,t}, \ldots, \mathscr{W}_{4,t}$, multiplied by the corresponding diffusion parameters $\sigma_1, \ldots, \sigma_4$. All fluxes that occur within the subsystem should follow the principle of mass conservation. We therefore depart from the usual procedure and add the stochastic terms in the following manner:

$$\mathrm{d}q(t) = K(t)q(t)(\mathrm{d}t + \Sigma \mathrm{d}\mathscr{W}_t) + u(t)\mathrm{d}t$$

with $\Sigma = \mathrm{diag}(\sigma_1, \sigma_2, \sigma_3, \sigma_4)$ and $\mathscr{W}_t = (\mathscr{W}_{1,t}, \ldots, \mathscr{W}_{4,t})^{\mathrm{T}}$. We fix the diffusion parameters thus: $\sigma_1 = \sigma_2 = \sigma_3 = \sigma_4 = 3$. The initial conditions are given by

$$q_2(0) = q_3(0) = q_4(0) = 0.$$

The test subjects receive a bolus injection with labeled leucine, so that we can fix the initial condition

$$q_1(0) = u_{1,0}$$

and simultaneously assume $u_1(t) = 0$ in the differential equation.

We assume identical differential equations, without the stochastic noise terms, however, for the states Q_i and the input U_1 of the tracee:

$$\frac{\mathrm{d}Q(t)}{\mathrm{d}t} = K(t)Q(t) + U(t)$$

with $Q(t) = (Q_i(t))_{i=1,2,3,4}^{\mathrm{T}}$, $U(t) = (U_1(t), 0, 0, 0)^{\mathrm{T}}$. Here, the tracee input $U_1(t) = U_1$ is presumed to be constant, but unknown. We therefore want to estimate this value along with the transfer parameters. Because a steady-state is presumed for the tracee (i.e., $\mathrm{d}Q_i(t)/\mathrm{d}t = 0$), we can solve the equations for $Q_1(t)$ and thus obtain:

$$Q_1(t) = \frac{(k_{11,2} + k_{1,2})U_1}{k_{0,1}(k_{11,2} + k_{1,2}) + k_{11,2}k_{1,2}}.$$

Each measurement is given by a value that is proportional to the ratio of tracer and tracee, with additional log-normal disturbances:

$$y_1(t) = p_1 \frac{q_1(t)}{Q_1(t)}\xi_t, \quad \xi_t \sim \text{Log-}\mathcal{N}\left(0, \sigma_{y_1}^2\right) \text{ independently for each } t,$$

where we assume the value of the variance parameter (this denotes the variance of $\log \xi_t$) to be $\sigma_{y_1}^2 = 0.5^2$. The parameter p_1 denotes the unknown proportion of plasma-leucine that is actually in the plasma. Since the parameters p_1 and U_1 are not jointly identifiable, we specify $p_1 = 0.65$ (on the basis of previous knowledge). More details concerning the deterministic model (without stochastic disturbances) can be found in [13] and [11]. Note that the stochastic disturbances are not part of the original model, but are our subsequent enhancements.

7.2 The Mixed-Effects Model

The model, as it was presented in the previous paragraphs, contains only flux parameters $k_{j,i}$ that are the same for each individual. In this form, the model does not account for individual differences between the various persons, nor does it consider group-specific differences between the patients and the control group. In the latter case, differences in flux parameters may arise when the persons examined belong in part to a group whose members are affected by a disease or have received a special treatment, while other persons belong to a control group. In order to account for these differences, we now introduce group-specific and patient-specific parameters into the model. Specifically, we split the transfer coefficients $k_{0,1}$ into a group-dependent and a patient-dependent part. In this fashion, we introduce so-called mixed effects into the model. Mixed effects generally make it more difficult to perform the estimates, since they not only increase the number of parameters to be estimated, but also result in a hierarchical ranking among the parameters. For the following estimates, we use measurement data collected in a study involving 34 persons—data that was already used in another context [12, 13]. From these 34 persons, 15 belong to the group of diabetes patients, while the other 19 belong to the control group. From earlier experiments, one sees that the degradation rate $k_{0,1}$ of the plasma-leucine differs significantly for persons with and without diabetes. We therefore assume that the expected value of $k_{0,1}$ is different in each group, that is, has either a value $k_{0,1}^d$ or a value $k_{0,1}^c$, depending on whether the person belongs to the diabetes or the control group. Moreover, we

also assume patient-dependent random factors ζ_p that reflect the parameter uncertainties among the individuals. In the end, we obtain

$$k_{0,1}^{(p)} = \begin{cases} \zeta_p k_{0,1}^{\mathrm{d}} & \text{if patient } p \text{ belongs to the diabetes group,} \\ \zeta_p k_{0,1}^{\mathrm{c}} & \text{if patient } p \text{ belongs to the control group,} \end{cases}$$

where all ζ_p are assumed to be static and independently log-normally distributed:

$$\zeta_p = \exp(\eta_p) \quad \text{with } \eta_p \sim \mathcal{N}\left(0, \sigma_{\eta_p}^2\right) \text{ independently for all } p$$

for $p = 1, \ldots, 34$. Consequently, each state $q_1, \ldots, q_4$ must be considered separately for each patient p. We indicate this in the notation by means of indices $q_1^{(p)}, \ldots, q_4^{(p)}$, $p = 1, \ldots, 34$.

The goal of our investigations, apart from estimating the remaining parameters, is thus to show that the group-dependent parameters $k_{0,1}^{\mathrm{d}}$, $k_{0,1}^{\mathrm{c}}$ are indeed different. To do so, we use the Bayesian approach for parameter estimation. For this reason, we treat the parameters essentially like state variables in the particle filter. Because the estimation of constant parameters is problematic with particle filter methods, it is standard to introduce a small, artificial stochastic dynamic so that the parameters can change slightly over time. This is done by allowing normally or log-normally distributed increments with decaying variances for the parameters in each time-step [36]. We also introduce corresponding dynamics for the static individual parameters η_p, which must also be estimated. Our process X_t is therefore given as an augmented state vector

$$X_t = \left(q_{1:4}^{(1:34)}(t), k_{0,1}^{\mathrm{c}}(t), k_{0,1}^{\mathrm{d}}(t), k_{1,2}(t), k_{1,3}(t), k_{3,1}(t), k_{4,3}(t), U_1(t), \eta_{1:34}(t)\right)^{\mathrm{T}}.$$

The complete model is thus a nonlinear mixed-effects model with three levels of effects (parameters), namely, global parameters, group-dependent parameters $k_{0,1}^{\mathrm{d}}$, $k_{0,1}^{\mathrm{c}}$, and individual parameters ζ_p.

7.3 Estimation Results

In this section, we compare the results of parameter estimations with the MTU particle filter and the standard particle filter. We performed estimations and subsequent test runs with the estimated parameters, using the data from all 34 patients (19 in the control group and 15 in the diabetes group). Specifically, we carried out the following computer experiments: In the first phase, we estimated the parameters with the MTU particle filter; for comparison purposes, we also separately estimated the parameters with the standard particle filter—under the same conditions and with the same seed value for the random number generator. The initial distribution of the particles was thus the same in both cases. For each run, we also calculated estimators for the effective sample size (ESS) and the data likelihood over time. These estimators allow a performance comparison for the MTU and the standard particle filters. In a second phase, we then used the empirical medians of the final parameter

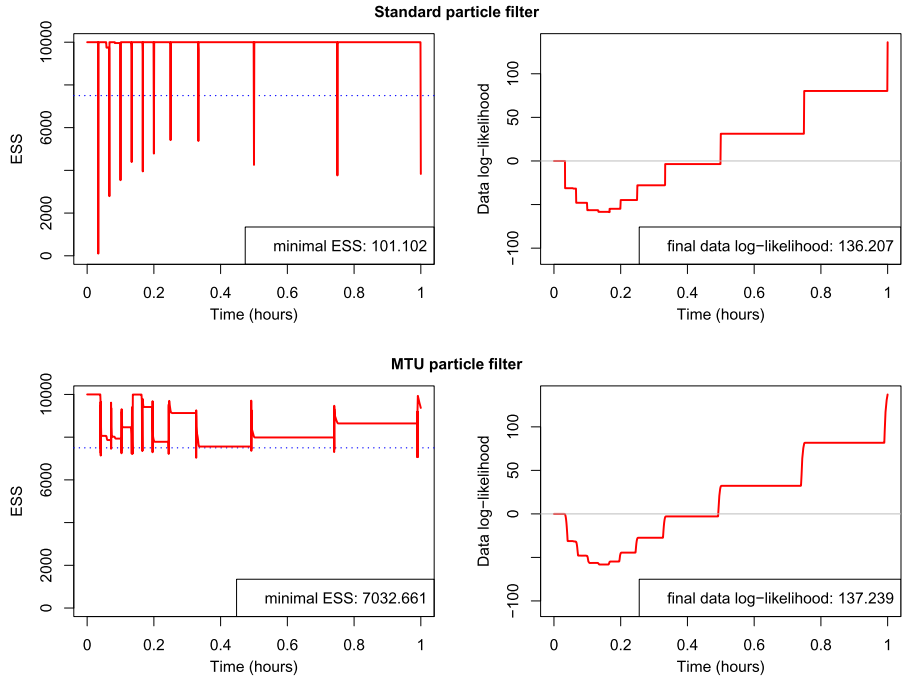

Fig. 17 Development of the Effective Sample Size (ESS) and the data likelihood over time during the parameter estimation. Standard particle filter (*top*) and MTU particle filter (*bottom*)

distributions in test runs, separately for each case. Both versions of the particle filter were used in these runs for state filtering and calculating the data likelihood—this time with fixed parameters given by the estimated values. In this manner, the resulting simulated distributions of the measurement values can be compared with the actual measurement values, both visually and quantitatively, by examining the data likelihood.

We performed our calculations with 10 000 particles and a resampling threshold of 7500. The step-size in the MTU filter was between 10^{-7} h and 10^{-3} h, adaptively calculated on the basis of the ESS estimate. In the standard filter, we used a fixed step-size of 10^{-3} h. Although the data contains measurements up to time $t = 8$ h, we only used values up to time $t = 1$ h for our estimates and test runs, mainly to save computing time. In any event, after time $t = 1$ h, the tracer concentrations are quite small and almost static, so it is not to be expected that including the later data would alter the estimates significantly. In our implementation of the particle filters, we sample directly from the (augmented) states X_t (i.e., $\tilde{X}_{[t_0,\infty)} = X_{[t_0,\infty)}$ in distribution), with the aid of the Euler–Maruyama method for discretizing the SDE over time [37].

Figure 17 shows the development over time t of the estimated value of the effective sample size ESS and the estimated data likelihood. Figures 18 and 19 are box plots showing the posterior distributions of the global/group parameters and the individual parameters, respectively, in the final time-step of the estimation.

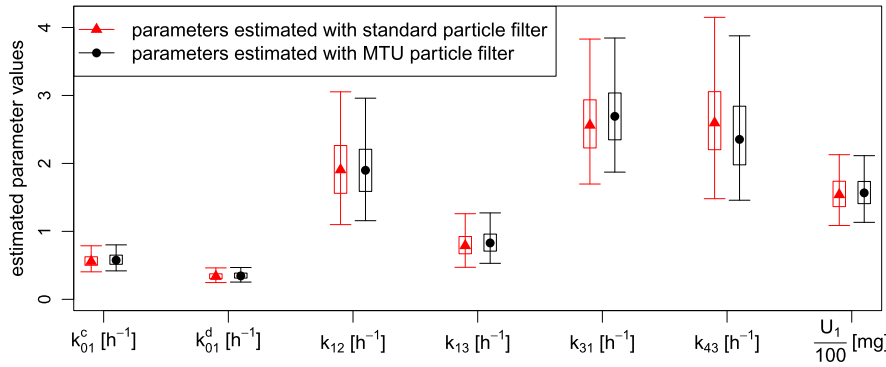

Fig. 18 Estimated global and group parameters. Box plots of the estimated posterior distributions for the global parameters and the group-dependent parameters. The medians are depicted with *triangles* (standard particle filter) or *circles* (MTU particle filter). The bottom and top of the box are the 0.25-quantile and the 0.75-quantile; i.e., 50 % of the values lie within the box. The whiskers mark the 0.025-quantile and the 0.975-quantile; i.e., 95 % of the values lie between the whiskers. The values for U_1 have been scaled by a factor of 0.01

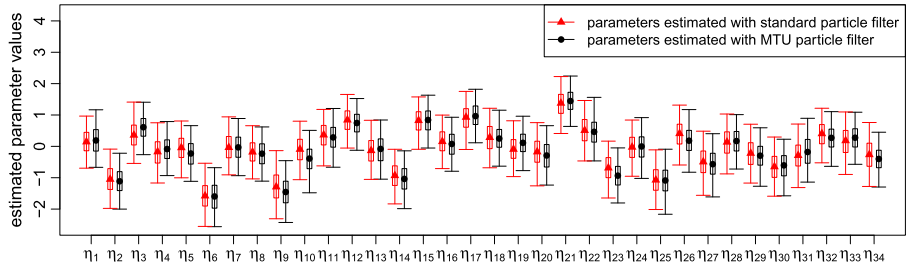

Fig. 19 Estimated individual parameters. Box plots of the estimated posterior distributions for the individual parameters. The medians are depicted with *triangles* (standard particle filter) or *circles* (MTU particle filter). The *bottom* and *top* of the box are the 0.25-quantile and the 0.75-quantile; i.e., 50 % of the values lie within the box. The whiskers mark the 0.025-quantile and the 0.975-quantile; i.e., 95 % of the values lie between the whiskers

A comparison of the results of the MTU-PF and the standard particle filter shows that both algorithms deliver very similar performance with regard to the quality of the estimated parameters; in each case, the development of the data likelihood is very similar, both during the estimation and the test run. The estimated log-likelihood of the data in the final estimation step is 137.239 for the MTU particle filter and 136.207 for the standard case; in other words, for all practical purposes, they are equal. The computation time for the MTU-PF is only slightly longer than that of the standard filter. A visual inspection of the test runs shows that the predicted distribution of the measurement values based on parameters estimated by both filters fits the data equally well. This impression is reinforced by the values of the estimated data likelihood. In the final step, the MTU particle filter deliv-

ers a log-likelihood value of 157.622, which is very similar to the 155.952 value delivered by the standard filter. The difference is insignificant; the uncertainty in the measurement time-points thus appears not to lead to differences in the actual estimation results, at least in this case.

In contrast to the insignificant differences in the likelihoods between the MTU-PF and the standard PF, the development of the ESS estimate in the estimation runs differs remarkably. The runs with the MTU particle filter deliver an ESS estimate with very high values throughout the estimation, and a minimum of 7032.661. This value lies just slightly under the resampling bound of 7500 (see Fig. 17, top). In contrast, the standard particle filter shows a substantially worse performance. One can see from the bottom of Fig. 17 that the ESS drops repeatedly to very low values, with a minimum of 101.102. Here, the MTU particle filter avoids degeneration of the particle cloud by holding the ESS at a high value at all time-points. This indicates that these results have been obtained on a sound basis and may be considered more reliable than those delivered by the standard algorithm.

A glance at the estimated values of the group parameters $k^c_{0,1}$ and $k^d_{0,1}$ (see Fig. 18) shows that, in both estimation cases (standard PF and MTU-PF), the rate $k^d_{0,1}$ for diabetes patients is only about 60 % of the rate $k^c_{0,1}$ for the control group (standard PF: 0.337 h^{-1} vs. 0.557 h^{-1}; MTU-PF: 0.346 h^{-1} vs. 0.577 h^{-1}). The good performance of the MTU particle filter, in particular, strengthens one's confidence in the results obtained and leads to the conclusion that the secretion rate $k_{0,1}$ is, in fact, lower for the group of diabetes patients than for the control group.

References

Publications of the Authors

1. Krengel, A., Hauth, J., Taskinen, M.R., Adiels, M., Jirstrand, M.: A continuous-time adaptive particle filter for estimations under measurement time uncertainties with an application to a plasma-leucine mixed effects model. BMC Syst. Biol. **7**, 8 (2013). doi:10.1186/1752-0509-7-8
2. Lang, P., Prätzel-Wolters, D., Kulig, S.: Modellreduktion und dynamische Beobachter für Torsionsschwingungen in Turbosätzen. In: Hoffmann, K.H., Jäger, W., Lohmann, T., Schunk, H. (eds.) Mathematik-Schlüsseltechnologie für die Zukunft, pp. 491–501. Springer, Berlin (1997)
3. Stahl, D., Hauth, J.: PF-MPC: Particle Filter-Model Predictive Control. Syst. Control Lett. **60**(8), 632–643 (2011)
4. Wirsen, A.: Monitoring von Torsionsschwingungen in Kraftwerksturbosätzen. In: Turbogeneratoren in Kraftwerken, Technik–Instandhaltung–Schäden. Haus der Technik, Essen (2011)
5. Wirsen, A., Humer, M.: Online Monitoring von Torsionsschwingungen in Wellensträngen von Kraftwerksturbosätzen. In: Symposium Schwingungsdiagnose–Schwingungsdiagnostische Überwachung von Kraftwerksturbosätzen–Methoden, Nutzen, Erfahrung. Potsdam Sanscoussi, Germany (2006)
6. Wirsen, A., Lang, P., Humer, M.: Systems for monitoring and analysing torsional vibrations in turbine generator shaft lines. In: Conference Proceedings of the 16th International Conference on Electrical Machines, Krakau, Polen (2004)

7. Wirsen, A., Mohring, J.: Methods for H_2 optimal actuator placement and controller design based on high dimensional parametric models of mechanical structures. In: Conference Proceedings IV European Conference on Computational Mechanics (ECCM) (2010)

Dissertations on the Topic at the Fraunhofer ITWM

8. Krengel, A.: A Modified Particle Filter with Adaptive Stepsize for Continuous-Time Models with Measurement Time Uncertainties (2013). Verlag Dr. Hut
9. Lang, P.: Model Reduction, Sensor Placement and Robust H_∞-Filter Design for Elastomechanical Systems (1998). Shaker Verlag
10. Wirsen, A.: Sensitivitätsanalyse und modaldatenbasierte Modelladaption bei elastomechanischen Systemen. dissertation.de (2002)

Further Literature

11. Adiels, M.: A compartmental model for kinetics of apolipoprotein B-100 and triglycerides in $VLDL_1$ and $VLDL_2$ in normolipidemic subjects. Licentiate thesis, Chalmers University of Technology, Göteborg (2002)
12. Adiels, M., Borén, J., Caslake, M.J.J., Stewart, P., Soro, A., Westerbacka, J., Wennberg, B., Olofsson, S.O.O., Packard, C., Taskinen, M.R.R.: Overproduction of VLDL1 driven by hyperglycemia is a dominant feature of diabetic dyslipidemia. Arterioscler. Thromb. Vasc. Biol. **25**(8), 1697–1703 (2005). doi:10.1161/01.ATV.0000172689.53992.25
13. Adiels, M., Packard, C., Caslake, M.J., Stewart, P., Soro, A., Westerbacka, J., Wennberg, B., Olofsson, S.O., Taskinen, M.R., Borén, J.: A new combined multicompartmental model for apolipoprotein B-100 and triglyceride metabolism in VLDL subfractions. J. Lipid Res. **46**, 58–67 (2005)
14. Andersen, K.E., Hojbjerre, M.: A population-based Bayesian approach to the minimal model of glucose and insulin homeostasis. Stat. Med. **24**(15), 2381–2400 (2005)
15. Andrieu, C., De Freitas, N., Doucet, A.: Sequential MCMC for Bayesian model selection. In: Proceedings of the IEEE Signal Processing Workshop on Higher-Order Statistics, pp. 130–134. IEEE, Caesarea (1999). doi:10.1109/HOST.1999.778709
16. Andrieu, C., Doucet, A., Holenstein, R.: Particle Markov chain Monte Carlo methods. J. R. Stat. Soc., Ser. B, Stat. Methodol. **72**(3), 269–342 (2010)
17. Ashyraliyev, M., Fomekong-Nanfack, Y., Kaandorp, J.A., Blom, J.G.: Systems biology: parameter estimation for biochemical models. FEBS J. **276**(4), 886–902 (2009). doi:10.1111/j.1742-4658.2008.06844.x
18. Beal, S., Sheiner, L.: NONMEM User's Guides. NONMEM Project Group. University of California, San Francisco (1994)
19. Berglund, M., Sunnåker, M., Adiels, M., Jirstrand, M., Wennberg, B.: Investigations of a compartmental model for leucine kinetics using non-linear mixed effects models with ordinary and stochastic differential equations. Math. Med. Biol. **29**(4), 361–384 (2011)
20. Bohlin, T.: Practical Grey-Box Process Identification: Theory and Applications. Advances in Industrial Control. Springer, London (2010)
21. Cappé, O., Godsill, S., Moulines, E.: An overview of existing methods and recent advances in sequential Monte Carlo. Proc. IEEE **95**(5), 899–924 (2007)
22. Chou, I.C.C., Voit, E.O.: Recent developments in parameter estimation and structure identification of biochemical and genomic systems. Math. Biosci. **219**(2), 57–83 (2009). doi:10.1016/j.mbs.2009.03.002

23. Cobelli, C., Saccomani, M.P., Tessari, P., Biolo, G., Luzi, L., Matthews, D.E.: Compartmental model of leucine kinetics in humans. Am. J. Physiol. **261**(4 Pt 1), E539–50 (1991)
24. Crisan, D., Doucet, A.: A survey of convergence results on particle filtering methods for practitioners. IEEE Trans. Signal Process. **50**(3), 736–746 (2002)
25. Del Moral, P., Doucet, A., Jasra, A.: Sequential Monte Carlo samplers. J. R. Stat. Soc., Ser. B, Stat. Methodol. **68**(3), 411–436 (2006)
26. Demant, T., Packard, C.J., Demmelmair, H., Stewart, P., Bedynek, A., Bedford, D., Seidel, D., Shepherd, J.: Sensitive methods to study human apolipoprotein B metabolism using stable isotope-labeled amino acids. Am. J. Physiol. **270**(6 Pt 1), E1022–36 (1996)
27. Donnet, S., Samson, A.: EM algorithm coupled with particle filter for maximum likelihood parameter estimation of stochastic differential mixed-effects models. http://hal.archives-ouvertes.fr/hal-00519576. Preprint version 2–21 Jul. 2011
28. Donnet, S., Samson, A.: Parametric inference for mixed models defined by stochastic differential equations. ESAIM Probab. Stat. **12**, 196–218 (2008)
29. Douc, R., Cappé, O., Moulines, E.: Comparison of resampling schemes for particle filtering. In: Proceedings of the 4th International Symposium on Image and Signal Processing and Analysis, 2005 ISPA, pp. 64–69. IEEE, Zagreb (2005)
30. Doucet, A., de Freitas, N., Gordon, N. (eds.): Sequential Monte Carlo Methods in Practice. Statistics for Engineering and Information Science. Springer, New York (2001)
31. Fearnhead, P.: MCMC, sufficient statistics and particle filters. J. Comput. Graph. Stat. **11**(4), 848–862 (2002)
32. Gordon, N., Salmond, D., Smith, A.: Novel approach to nonlinear/non-Gaussian Bayesian state estimation. IEE-Proc.-F **140**(2), 107–113 (1993)
33. Hol, J.D., Schön, T.B., Gustafsson, F.: On resampling algorithms for particle filters. In: Proceedings of Nonlinear Statistical Signal Processing Workshop (NSSPW), pp. 79–82. IEEE, Cambridge (2006)
34. Hu, X.L., Schön, T., Ljung, L.: A basic convergence result for particle filtering. IEEE Trans. Signal Process. **56**(4), 1337–1348 (2008)
35. Humer, M.: Erfassung und Bewertung von Torsionsschwingungen in Wellensträngen von Kraftwerksturbosätzen. Ph.D. thesis, TU Dortumund (2004)
36. Hürzeler, M., Künsch, H.R.: Approximating and maximizing the likelihood for a general state-space model. In: Sequential Monte Carlo Methods in Practice. Springer, New York (2001)
37. Kloeden, P.E., Platen, E.: Numerical Solution of Stochastic Differential Equations. Springer, Berlin (1999)
38. Moles, C.G., Mendes, P., Banga, J.R.: Parameter estimation in biochemical pathways: a comparison of global optimization methods. Genome Res. **13**(11), 2467–2474 (2003). doi:10.1101/gr.1262503
39. Neubauer, M.: Aktive Bedämpfung von Drehschwingungen im Fahrzeugantriebstrang. Ph.D. thesis, TU, Braunschweig (2011)
40. Nielsen, J., Madsen, H., Young, P.: Parameter estimation in stochastic differential equations: an overview. Annu. Rev. Control **24**, 83–94 (2000)
41. Overgaard, R.V., Jonsson, N., Tornøe, C.W., Madsen, H.: Non-linear mixed-effects models with stochastic differential equations: implementation of an estimation algorithm. J. Pharmacokinet. Pharmacodyn. **32**(1), 85–107 (2005)
42. Pitt, M.K., Shephard, N.: Filtering via simulation: auxiliary particle filter. J. Am. Stat. Assoc. **94**, 590–599 (1999)
43. Racine-Poon, A., Wakefield, J.: Statistical methods for population pharmacokinetic modelling. Stat. Methods Med. Res. **7**(1), 63–84 (1998)
44. Sheiner, L., Wakefield, J.: Population modelling in drug development. Stat. Methods Med. Res. **8**(3), 183 (1999)

45. Storvik, G.: Particle filters for state-space models with the presence of unknown static parameters. IEEE Trans. Signal Process. **50**(2), 281–289 (2002)
46. Tornøe, C.W., Overgaard, R.V., Agersø, H., Nielsen, H.A., Madsen, H., Jonsson, E.N.: Stochastic differential equations in nonmem: implementation, application, and comparison with ordinary differential equations. Pharm. Res. **22**(8), 1247–1258 (2005)
47. Ungarala, S., Dolence, E., Li, K.: Constrained extended Kalman filter for nonlinear state estimation. In: 8th International IFAC Symposium on Dynamics and Control of Process Systems, vol. 2. Cancun, Mexico (2007)
48. Zhou, K., Doyle, J.C., Glover, K.: Robust and Optimal Control. Prentice Hall, New York (1996)