# A Nonlinear State Space Model for the Blood Glucose Metabolism of a Diabetic

## Ein nichtlineares Zustandsraummodell für den Blutglukosemetabolismus eines Diabetikers

Thomas Briegel and Volker Tresp

The blood glucose metabolism of a diabetic is a complex nonlinear process closely linked to a number of internal factors which are not easily accessible to measurement. Based on accessible information – such as occasional blood glucose measurements and information about food intake and physical exercise – the system appears highly stochastic and the quantity of main interest, the blood glucose concentration, is very difficult to model and to predict. In this paper we describe a stochastic nonlinear state space model for modeling the blood glucose concentration of a diabetic patient. The model structure is based on physiological prior knowledge and the main nonlinearities are modeled using artificial neural networks. Offline training of the model is performed using a newly developed Monte-Carlo generalized EM (expectation maximization) algorithm. Online prediction is performed using particle filters. Our experimental results show that our approach provides better prediction results than a number of competing approaches.

Der Blutglukosemetabolismus eines Diabetikers ist ein komplexer nichtlinearer Prozess, der sehr stark von einer Reihe interner Einflussfaktoren abhängt, welche Messungen nicht leicht zugänglich sind. Basierend auf zugänglicher Information – wie gelegentlicher Blutglukosemessungen und Information über Nahrungsaufnahme und physikalische Betätigung – erscheint das System hochgradig stochastisch und ist daher sehr schwierig zu modellieren. In dieser Veröffentlichung beschreiben wir ein stochastisches nichtlineares Zustandsraummodell zur Modellierung des Blutglukosemetabolismus eines Diabetikers. Die Modellstruktur basiert auf physiologischem Vorwissen, wobei die Hauptnichtlinearitäten durch ein künstliches Neuronales Netz modelliert werden. Offline Training wird mit Hilfe eines neu entwickelten Monte-Carlo EM-Algorithmus (expectation maximization) durchgeführt. Zur Online Vorhersage verwenden wir ein Partikel-Filter. Unsere experimentellen Resultate zeigen, dass unser Ansatz bessere Ergebnisse liefert als eine Vielzahl konkurrierender Verfahren.

## 1 Introduction

Diabetes mellitus is one of the most common chronic diseases. According to a recent study 5.6% of all women and 4.7% of all men in Germany in the age group between ages 18 and 70 years are affected [1]. In type I diabetes, the disease is caused by the failure of the pancreas to produce a sufficient amount of insulin which leads to an uncontrolled increase in blood glucose unless the patient administers insulin, typically by subcutaneous injection. Slowly acting (basal) insulin is administered to supply a baseline of insulin concentration whereas fast acting (normal) insulin is injected to accommodate for the increased demand of insulin after the intake of food. In consultation with the patient's physician and based on irregular measurements of the blood glucose, the exact time and amount of insulin injection is determined by the patient her- or himself. Although an experienced patient with a stable metabolism can achieve sufficient control of her or his blood glucose concentration, this is often not the case for unexperienced patients or elderly patients. Consequently, a number of computer-based approaches have been developed for supporting the insulin treatment of diabetic patients [2–4]. In addition to functions which are realized today, such as simple storage and display functions, computer-based support could realize systems which are able to analyze past therapy, to predict future blood glucose levels, and to provide therapy recommendations.

One can distinguish between approaches based on expert systems, model-based approaches, and black-box approaches. In expert systems, one tries to derive recommendations for improving the therapy of a patient based on expert prior knowledge [5; 6]. This is a direct approach in the sense that the expert system models the map between the current state of the patient and the required treatment. The other approaches are indirect in the sense that the state of the patient is evaluated based on a model and therapy improvements and predictions are based on this model as well.

In a purely model-based approach one constructs a physiological model of the blood glucose metabolism and derives predictions and recommendations based on the state of the patient using this physiological model [7–9]. In a black-box approach one would use a statistical model such as a linear predictive model, a nonparametric model or an artificial neural network (in the following referred to as neural networks) to model the relevant dependencies purely based on data obtained from patients [10; 11]. One might here make a distinction between static and dynamical models where the latter base their predictions at least partially on their previous estimates, i.e., they contain a feedback loop.

All these approaches have their particular problems: expert systems are typically inflexible in adapting to the particular physiology of a specific patient. In addition, extracting and maintaining expert knowledge is known to be a difficult process, particularly due to the fact that experts often base their recommendations on information which cannot easily be parameterized such as the general health state of the patient. In addition, the medical expert can take into account background information and context knowledge which is difficult to capture in an expert system. The problem with physiological models is that they are derived in general from measurements obtained in a clinical setting and the derived models are not necessarily applicable to the daily life of a patient. May be even more severe is the problem that those model contain a large number of compartments which might be accessible for measurement in a clinical setting but cannot be measured in daily life. The problem with black-box approaches is that they might be able to model the available data very well but often do not generalize sufficiently to situations which are not represented well in the data.

In this paper we pursue a hybrid "grey-box" approach in which the main nonlinear dependency of a physiologically motivated model is replaced with a neural network, thus providing additional modeling flexibility. The inputs to the model are all accessible to measurement in the normal daily life of a patient. The resulting model is a stochastic nonlinear state space model. The stochasticity in the model captures the unmodeled dynamics in the physiology of the patient as well as measurement errors.

A particular advantage of state space models is that they model the data generation process and as such can easily deal with missing data. Missing data are a major problem in this application since the main quantity of interest, the blood glucose concentration, is only measured a few times a day during the normal life of a diabetic patient. The model parameters are estimated based on a set of measurements of a diabetic patient using a newly developed Monte-Carlo generalized EM (expectation maximization) algorithm. Online prediction is performed using particle filtering which has become a popular approach for online estimation in nonlinear state space models.

The paper is organized as follows. In the next section we introduce stochastic nonlinear state space models and describe a newly developed approach to training stochastic nonlinear state space models based on a new Monte-Carlo generalized EM (expectation maximization) algorithm, reviewing results reported in [12], [13], and [14]. Furthermore, we describe the particle filter which we used for prediction. In Section 3 we briefly describe the main processes in the blood glucose physiology and develop a physiologically motivated compartment model and a hybrid neuro-compartment state space model. The new contribution of this paper is the application of the Monte-Carlo generalized EM algorithm and the particle filter in context of blood glucose modeling. In Section 4 we describe experimental results. Finally, in Section 5 we present an evalution and conclusions.

## 2 The Stochastic Nonlinear State Space Model

### 2.1 Model Formulation

Nonlinear state space models (NSSM) are a general framework for representing nonlinear time series and processes. Mathematically, a NSSM is described by the system equation

$$x_t = f_w(x_{t-1}, u_t) + \epsilon_t \tag{1}$$

where $x_t$ denotes a hidden state variable, $\epsilon_t$ denotes zero-mean uncorrelated Gaussian noise with covariance $Q_t$ (written as $\epsilon_t \sim \mathcal{N}(0, Q_t)$) and $u_t$ is an exogenous known input vector. All variables are in general multidimensional. The time-series measurements $y_t$ are related to the unobserved hidden states $x_t$ through the observation equation

$$y_t = g_v(x_t, u_t) + v_t \tag{2}$$

where $v_t$ is uncorrelated Gaussian noise with covariance $V_t$. In the following we assume that the nonlinear mappings $f_w(.)$ and $g_v(.)$ are modeled as neural networks or a similar flexible nonlinear model with weight vectors $w$ and $v$, respectively.[1] The initial state $x_0$ is assumed to be Gaussian distributed with mean $a_0$ and covariance $Q_0$. The two challenges in NSSMs are the interrelated tasks of inference and learning. In inference we try to estimate the states of unknown variables $x_s$ given some measurements $y_1, \ldots, y_t$

[1] In our experiments, we used the well known multilayer perceptron [15].

(typically the states of past ($s < t$), present ($s = t$) or future ($s > t$) values of $x_t$) and in learning we want to adapt some unknown parameters in the model (e.g., the neural network weight vectors $w$ and $v$) given a set of measurements. In this paper we assume that training is performed based on a fixed data set, i.e., we do not consider online learning. This allows us to formulate the learning problem in a maximum likelihood framework. In the special case of linear state space models with Gaussian noise, efficient algorithms for inference and maximum likelihood learning exist. The latter can be implemented using EM update equations in which the E-step is implemented using forward-backward Kalman filtering [16]. Inference in NSSMs consists of calculating the distribution or the expected values of hidden variables. If the system is nonlinear, however, the problem of inference and learning leads to complex integrals which are usually considered intractable [17] unless suitable approximations are made as we will do in the next section.

## 2.2 The Gradients for Nonlinear State Space Models

Given our assumptions we can write the joint probability of the complete data for $t = 1, \ldots, T$ as[2]

$$p(X_T, Y_T, U_T) = p(U_T)\, p(x_0) \prod_{t=1}^{T} p(x_t|x_{t-1}, u_t) \prod_{t=1}^{T} p(y_t|x_t, u_t) \tag{3}$$

where $U_T = \{u_1, \ldots, u_T\}$ is a set of *known* inputs which means that $p(U_T)$ is irrelevant in the following. Since only $Y_T = \{y_1, \ldots, y_T\}$ and $U_T$ are observed, the log-likelihood of the model is

$$\log \mathrm{L} = \log \int p(X_T, Y_T|U_T)\, p(U_T)\, dX_T \propto \log \int p(X_T, Y_T|U_T) dX_T \tag{4}$$

with $X_T = \{x_0, \ldots, x_T\}$. By inserting the Gaussian noise assumptions we obtain the gradients of the log-likelihood with respect to the neural network weight vectors $w$ and $v$, respectively [13; 18]

$$\frac{\partial \log \mathrm{L}}{\partial w} \propto \sum_{t=1}^{T} \int\int (x_t - f_w(x_{t-1}, u_t))^\top Q_t^{-1} \times \frac{\partial f_w(x_{t-1}, u_t)}{\partial w} p(x_t, x_{t-1}|Y_T, U_T)\, dx_{t-1} dx_t \tag{5}$$

$$\frac{\partial \log \mathrm{L}}{\partial v} \propto \sum_{t=1}^{T} \int (y_t - g_v(x_t, u_t))^\top V_t^{-1} \times \frac{\partial g_v(x_t, u_t)}{\partial v} p(x_t|Y_T, U_T)\, dx_t\,. \tag{6}$$

[2] In the following, each probability density is conditioned on the current model. For notational convenience, we do not indicate this fact explicitly.

## 2.3 Monte-Carlo Generalized EM Learning

### 2.3.1 Approximating the E-step

The integrals in the previous equations can be solved using Monte-Carlo integration which leads to the following learning algorithm [12; 13].

1. Generate $S$ samples $x_T^s = \left\{x_{0|T}^s, \ldots, x_{T|T}^s\right\}_{s=1}^S$ from $p(X_T|Y_T, U_T)$ using the current model parameter estimates $w^{\text{old}}$ and $v^{\text{old}}$. (Monte-Carlo E-Step).
2. Treat those samples as real data and update

$$w^{\text{new}} = w^{\text{old}} + \eta \frac{\partial \log \mathrm{L}}{\partial w}$$

and

$$v^{\text{new}} = v^{\text{old}} + \eta \frac{\partial \log \mathrm{L}}{\partial v}$$

with stepsize $\eta$ and

$$\frac{\partial \log \mathrm{L}}{\partial w} = \frac{1}{S} \sum_{t=1}^{T} \sum_{s=1}^{S} \left(x_{t|T}^s - f_w\left(x_{t-1|T}^s, u_t\right)\right)^\top \times Q_t^{-1} \left.\frac{\partial f}{\partial w}\right|_{x_{t-1} = x_{t-1|T}^s} \tag{7}$$

$$\frac{\partial \log \mathrm{L}}{\partial v} = \frac{1}{S} \sum_{t=1}^{T} \sum_{s=1}^{S} \left(y_t - g_v\left(x_{t|T}^s, u_t\right)\right)^\top \times V_t^{-1} \left.\frac{\partial g}{\partial v}\right|_{x_t = x_{t|T}^s} \tag{8}$$

(generalized M-step). Go back to step one.

The second step is simply a stochastic gradient step. The computational difficulties lie in the first step. Methods which produce samples from multivariate distributions such as Gibbs sampling and other Markov chain Monte-Carlo methods have (at least) two problems. Firstly, the sampling process has to "forget" its initial condition which means that the first samples have to be discarded and there are no simple analytical tools available to determine how many samples must be discarded. Secondly, subsequent samples are highly correlated which means that many samples have to be generated before a sufficient amount of independent samples is available. Since it is so difficult to sample from the correct posterior distribution $p(X_T|Y_T, U_T)$ the idea in this paper is to generate samples from an approximate distribution from which it is easy to draw samples. In the next sections we present two approximations using multivariate Gaussians. The first one is based on the extended Kalman filter and smoother and the second one on a Fisher scoring algorithm.

### 2.3.2 Approximate Mode Estimation Using the Extended Kalman Filter

Whereas the Kalman filter and smoother is an optimal state estimator for linear state space models, the extended Kalman filter and smoother is a suboptimal state estimator for NSSMs based on local linearizations of the non-linearities.[3]

[3] Note that we do not include the parameters in the NSSM as additional states to be estimated as done by other authors, e.g. [19].

The *extended Kalman filter and smoother* (EKFS) algorithm is a forward-backward algorithm and can be derived as an approximation to posterior mode estimation for Gaussian error sequences [20]. Its application to our framework amounts to approximating $x_t^{\text{mode}} \approx x_{t|T}^{\text{EKFS}}$ where $x_{t|T}^{\text{EKFS}}$ is the smoothed estimate of $x_t$ obtained from forward-backward extended Kalman filtering over the set of measurements $Y_T$ and $x_t^{\text{mode}}$ is the mode of the posterior distribution $p(x_t|Y_T, U_T)$. We use $x_{t|T}^{\text{EKFS}}$ as the center of the approximating Gaussian. The EKFS also provides an estimate of the error covariance of the state vector at each time step $t$ which can be used to form the covariance matrix of the approximating Gaussian. The EKFS equations can be found in [17]. To generate samples we recursively apply the following algorithm. Given $x_{t-1|T}^s$ is a sample from the Gaussian approximation of $p(x_{t-1}|Y_T, U_T)$ at time $t-1$, draw a sample $x_{t|T}^s$ from $p(x_t|x_{t-1} = x_{t-1}^s, Y_T, U_T)$. The last conditional density is Gaussian with mean and covariance calculated from the EKFS approximation and the lag-one error covariances derived in [16], respectively.

#### 2.3.3 Exact Mode Estimation Using the Fisher Scoring Algorithm

If the system is highly nonlinear, however, the EKFS can perform badly in finding the posterior mode due to the fact that it uses a first order Taylor series expansion of the nonlinearities $f_w(.)$ and $g_v(.)$. A useful – and computationally tractable – alternative to the EKFS is to compute the "exact" posterior mode by maximizing $\log p(X_T|Y_T, U_T)$ with respect to $X_T$. A suitable way to determine a stationary point of the log posterior, or equivalently, of $p(X_T, Y_T|U_T)$ (derived from (3) by dropping $p(U_T)$) is to apply *Fisher scoring*. With the current estimate $X_T^{\text{FS,old}}$ we get a better estimate $X_T^{\text{FS,new}} = X_T^{\text{FS,old}} + \eta\delta$ for the unknown state sequence $X_T$ where $\delta$ is the solution of

$$\mathcal{J}\left(X_T^{\text{FS,old}}\right)\delta = s\left(X_T^{\text{FS,old}}\right) \tag{9}$$

with the score function

$$s(X_T) = \frac{\partial \log p(X_T, Y_T|U_T)}{\partial X_T}$$

and the expected information matrix[4]

$$\mathcal{J}(X_T) = \mathrm{E}\left[-\frac{\partial^2 \log p(X_T, Y_T|U_T)}{\partial X_T \partial X_T^\top}\right].$$

By extending the arguments given in [21] to nonlinear state space models it turns out that solving equation (9) – i.e., to compute the inverse of the expected information matrix – can be performed by Cholesky decomposition in one forward and backward pass.[5] The forward-backward steps can be implemented as a fast EKFS-like algorithm which has to be iterated to obtain the maximum posterior estimates $x_t^{\text{mode}} = x_{t|T}^{\text{FS}}$ [12; 13]. Our experiments have shown that Fisher scoring is successful in finding the "exact" mode, the EKFS algorithm is not. Samples of the approximating Gaussian are generated in the same way as described in the last section.

### 2.4 Prediction: Particle Filters

Particle filtering (a.k.a. condensation algorithm or sampling importance resampling) is currently a very popular method for prediction in nonlinear state space model. In particle filtering, the conditional distribution $p(x_{t-1}|Y_{t-1}, U_{t-1})$ is represented by $S$ samples drawn from this distribution. Given those samples, one draws samples according to the state transition probability and weights them with the probability of the new measurement given the sample. One then produces $S$ new samples by resampling using those weights. We implemented a version of particle filtering introduced by [22]:

Algorithm sequential importance resampling
*Initialization: draw S samples from*
$x_0^s \sim \mathcal{N}(a_0, Q_0), \quad \tilde{c}_0^s = 1/S$
$t = 1, 2, \ldots$
*Predictor step: draw S samples*
$x_{t|t-1}^s \sim \mathcal{N}\left(f_w(x_{t-1|t-1}^s, u_t), Q_t\right)$
*Corrector step:*
$s = 1, \ldots, S$
*Evaluate the importance weights:*
$c_t^s = c_{t-1}^s \mathcal{N}\left(y_t|g_v(x_{t|t-1}^s, u_t), V_t\right)$
*Normalize the importance weights:*
$\tilde{c}_t^s = \frac{c_t^s}{\sum_{\tilde{s}=1}^S c_t^{\tilde{s}}}$
*Resampling stage:*
*if* $S^{\text{eff}} > threshold$
$x_{t|t}^s = x_{t-1|t}^s$
*else*
*resample index* $\tilde{s}$ *from* $\{x_{t|t-1}^s, \tilde{c}_t^s\}$
$x_{t|t}^s = x_{t-1|t}^{\tilde{s}}$
$c_t^s = 1/S$
*end*
*end*
*end*

where $\mathcal{N}(z|\mu, \Sigma)$ is a multivariate Gaussian density with mean $\mu$ and covariance matrix $\Sigma$ evaluated at $z$. In [22] a mathematical proof is given that the above algorithm creates samples according to the state prediction density $p(x_t|Y_{t-1}, U_{t-1})$ and the filtering density $p(x_t|Y_t, U_t)$.

## 3 Modelling the Blood Glucose Metabolism

### 3.1 Diabetes Mellitus

Figure 1 shows a very simplified model of the glucose metabolism. The digestive tract breaks down most of the carbohydrates in the food into glucose and releases it into the blood stream. Glucose is stored in the liver as glycogen and released again if the blood glucose drops too

[4] Note that the difference between the Fisher scoring and the Gauss-Newton update is that in the former we take the expectation of the information matrix.

[5] The expected information matrix is a positive definite block-tridiagonal matrix.

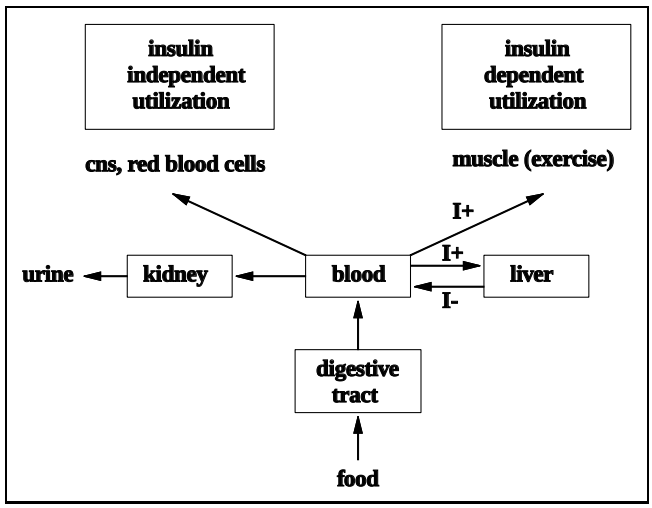


**Figure 1:** A simplified model of the glucose metabolism. The arrows indicate the transport of glucose. I+ and I- indicate glucose transports which are promoted, respectively inhibited, by insulin in the blood.

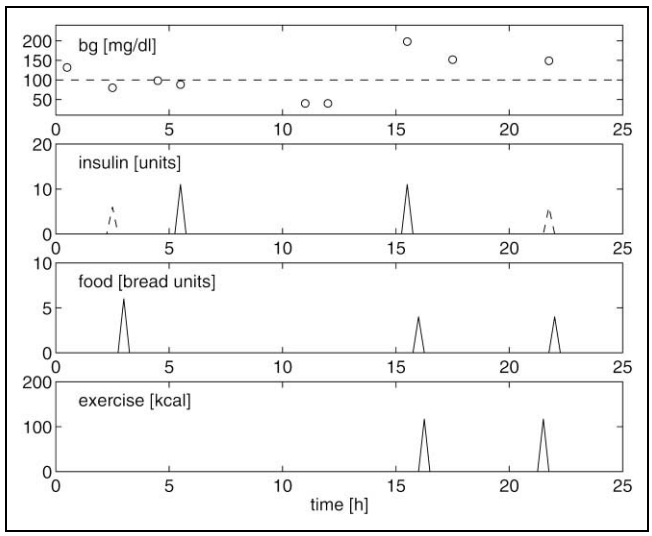


**Figure 2:** Short time window (25 h) of the data set. The top part shows the blood glucose measurements (unfilled circles) in mg/dl. The average blood glucose level of a normal person is indicated by the dashed line. The other plots show insulin injections (basal insulin $u_{t,1}$ (dashed peaks) and normal insulin $u_{t,2}$ (continuous peaks), respectively), food intake ($u_{t,3}+u_{t,4}+u_{t,5}$) and exercise ($u_{t,6}+u_{t,7}$), respectively, as singletons in the lower part of the picture. As a reaction to the severe hypoglycemia at 11 h and 12 h the patient stopped to administer insulin. As a consequence the patient had a case of hyperglycemia at 16 h and he reacted with immediately injecting 10 units of insulin.

low. The extraction of glucose from the blood stream by the liver requires insulin, which suppresses indirectly the inverse process, the release of glucose by the liver. Most cells – including muscle cells – need insulin to absorb glucose from the blood stream. The central nervous system and the red blood cells rely completely on glucose for their energy supply, but fortunately do not require insulin to metabolize it. Glucose is lost in urine (renal clearance) if the blood glucose level increases above the renal threshold. The glucose metabolism and blood glucose level in a healthy person are kept within tight tolerances and are controlled by the secretion of insulin by the beta-cells of the pancreas. In a person with type I diabetes mellitus, the insulin production is either largely reduced or ceases completely due to impairment or death of the beta cells. Treatment consists of administering insulin by subcutaneous injection, or more rarely, through an external or implanted insulin pump. In modern "intensive therapy," the patient attempts to lead as normal a life as possible, adjusting her or his therapy to the daily schedule of meals and exercise by monitoring her or his blood glucose a few times a day. Essentially, the patient has to replace an internal feedback mechanism with an external control which can be done only imperfectly. Consequently, a patient's blood glucose can often be outside of the desired range.

Even the simplified depiction in Figure 1 illustrates why the glucose metabolism of a diabetic is so unstable. If the insulin concentration drops too low, glucose cannot be removed from the blood sufficiently fast and the liver even releases additional glucose. Blood glucose can rise far above its normal level, a state of hyperglycemia. In the opposite case, if too much insulin is present in the blood (i.e., after an injection of a high dosage of insulin), cells continue to absorb glucose rapidly from the blood and the liver glucose production is blocked, leading to hypoglycemia. Note that the blood normally only contains less than 6 g of glucose, which approximately corresponds to three cubes of sugar. Since the blood is not a large reservoir of glucose, several times the amount of glucose normally present in the blood can be added by the digestive tract and the liver and/or removed by the cells of the body every hour. These rapid changes in the blood glucose level make control and prediction so difficult.

Our data set consists of the protocol of a male type I diabetic patient over a period of 63 days. During that time period, times and dosages of insulin injections (basal insulin $u_{t,1}$ and normal insulin $u_{t,2}$), the times and amounts of food intake (fast $u_{t,3}$, intermediate $u_{t,4}$ and slow $u_{t,5}$ carbohydrates), the times and durations of exercise (regular $u_{t,6}$ or intense $u_{t,7}$) and the blood glucose level $x_t$ (measured a few times a day) were recorded with 15 minute time resolution. The $u_{t,j}$, $j = 1, \ldots, 7$ are equal to zero except if there is an event, such as food intake, insulin injection or exercise. In the time period of 63 days, we only had 463 blood glucose measurements available in total which means that at 92% of the time steps, the blood glucose is unknown. Figure 2 shows a short time window (25 hours) of our data set.

## 3.2 Nonlinear State Space Models for the Blood Glucose Metabolism

In the following we discuss various approaches for modeling the glucose metabolism of a diabetic patient.

### 3.2.1 Compartment Models

Compartment models are typically used for modeling complex systems with dynamics that can be approximated

well by a number of discrete subsystems which interact by exchanging materials [23]. We have formulated a compartment model using the different compartments depicted in the simplified model of the glucose metabolism of Figure 1, i.e., blood, liver, kidney, digestive tract, insulin independent utilization and insulin dependent utilization.

The effects of the inputs insulin, food and exercise on the blood glucose are delayed and can be approximated by linear response functions [24–26]. Let $d_{t,j}$ describe the effect at time $t$ of past inputs $\{u_{\tau,j}\}_{\tau=0}^{t}$. The response $d_{t,2}$ of normal insulin after injection is determined by the diffusion of the subcutaneously injected insulin into the blood stream and can be modeled by three first order compartments in series or, as we have done, by a response function of the form [24]

$$d_{t,2} = \sum_{\tau=0}^{t} g_2(t-\tau) u_{\tau,2} \tag{10}$$

with

$$g_2(z) = a_2 z^2 e^{-b_2 z} . \tag{11}$$

Figure 3 shows a typical impulse response $g_2(t-\tau)$ to an injection of 10 units of normal (soluble) insulin at time $\tau = 0$ on the blood glucose level. The same response curve with different parameters were used also for basal insulin. The time-dependent effects for the digestive tract and for exercise are less well-known. In our experiments we followed [24] and used response functions of the above form as well.

We consider in the following the functional block of the response functions which perform the mapping from $u_t = (u_{t,1}, \ldots, u_{t,7})^\top$ to $d_t = (d_{t,1}, \ldots, d_{t,7})^\top$ as one compartment module $\mathcal{M}_d$. The effects of the inputs now influence the blood glucose concentration in a nonlinear way. We model the dynamics of the blood glucose using a nonlinear autoregressive model with exogenous inputs of the form

$$\begin{aligned} x_t =& x_{t-1} + c_1(d_{t,3} + d_{t,4} + d_{t,5}) + \\ & c_2\left[\exp\left(-c_3(d_{t,1} + d_{t,2})\right) - c_4 x_{t-1}\right] \\ & - c_5(d_{t,1} + d_{t,2})(x_{t-1} + c_6) - c_7\sqrt{x_{t-1}} - c_8 x_{t-1}^3 \\ & - c_9(d_{t,6} + d_{t,7}) + \epsilon_t \end{aligned} \tag{12}$$

where $x_t$ is the blood glucose at time $t$ and $\epsilon_t$ denotes zero-mean uncorrelated Gaussian noise terms with variance $\sigma_\epsilon^2$. The nonlinearity in the above time-series model was derived from parameterizing published data describing the dependencies [24; 25; 27]. The second term on the right side of the difference equation describes the increase in blood glucose due to carbohydrates in the food, the third term approximates the insulin dependent glucose production of the liver (if this term becomes negative, it is set to zero), the fourth term describes the insulin dependent usage of blood glucose, the fifth term describes the insulin independent usage of blood glucose, the sixth term describes the renal clearance (this term is set to zero if blood glucose is below the renal threshold) and finally the last term term describes the blood glucose lowering effect of exercise. Figure 4 shows how the functional forms in equation (12) were obtained by fitting nonlinear maps to published data. The functional block described by equation (12) is considered a second compartment module $\mathcal{M}_n$.

Figure 5 shows the overall compartment system containing the functional blocks $\mathcal{M}_d$ and $\mathcal{M}_n$. In the experiments we

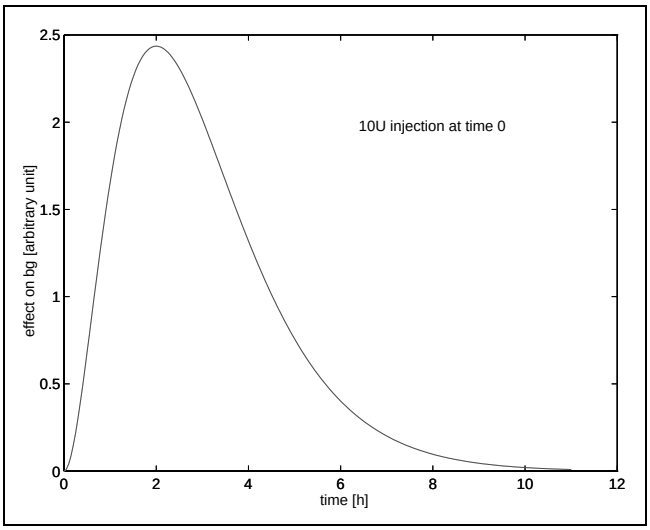


**Figure 3:** The plot shows the effect of insulin on the blood glucose level as a consequence of injecting 10 units of normal insulin at time $\tau = 0$ as simulated by our preprocessing module using impulse response $g_2(t-\tau)$. This time behavior is following approximately the concentration of insulin in the blood stream as measured experimentally.

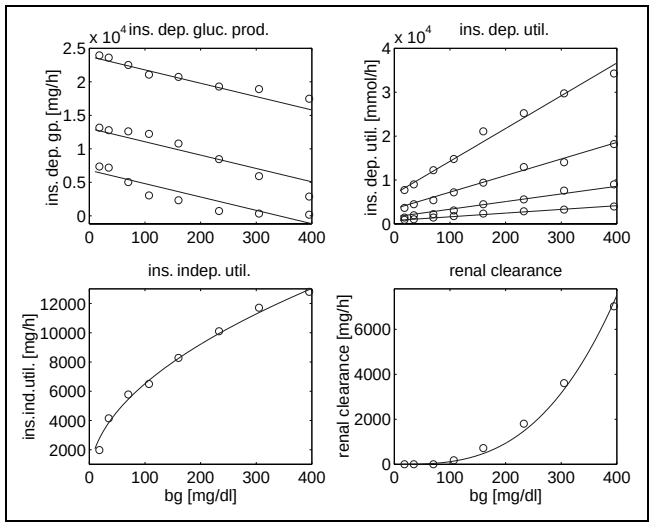


**Figure 4:** The plots show the rate of change of the blood glucose as a result of the insulin dependent glucose production (term three in equation (12)), the insulin dependent utilization (term four), the insulin independent glucose removal (term five), and renal clearance (term six). The small circles indicate published data [27] and the solid lines show the fitted parameterization used in equation (12). The plots for insulin dependent glucose production and utilization, respectively are shown for three (from top to down: 0 µU/ml, 10 µU/ml, 20 µU/ml), respectively four (from top to down: 80 µU/ml, 40 µU/ml, 20 µU/ml, 10 µU/ml) different levels of active insulin in the blood.

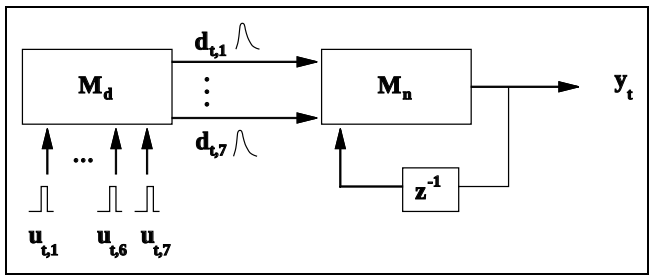


**Figure 5:** A compartment model approach, consisting of two compartment modules $\mathcal{M}_d$ and $\mathcal{M}_n$. $z^{-1}$ indicates a unit time delay.

initialized all parameters in $\mathcal{M}_d$, $\{a_i, b_i\}_{i=1}^7$, and in $\mathcal{M}_n$, $\{c_i\}_{i=1}^9$, with values derived from literature [24; 25; 27].

### 3.2.2 Neuro-Compartment State Space Models

Based on standard medical literature on diabetes [28] and based on consultation with a physician, we conclude that the functional form of the response functions in $\mathcal{M}_d$ is sufficient to capture the various delays of the inputs and can be tuned to the physiology of the patient by varying the parameters $a_i, b_i$. The nonlinear equation (12) in the second compartment module $\mathcal{M}_n$, on the other hand, is based on a number of uncertain physiological assumptions, and we cannot necessarily expect that the true interactions can be approximated by just adapting equation parameters. To be able to capture more complex interactions, we replace $\mathcal{M}_n$ by a neural network (multi-layer perceptron). The five inputs to the network are insulin ($in_{t,1} = d_{t,1} + d_{t,2}$), food ($in_{t,2} = d_{t,3} + d_{t,4} + d_{t,5}$), exercise ($in_{t,3} = d_{t,6} + d_{t,7}$) and the current and previous estimates of the blood glucose. We obtain now

$$x_t = x_{t-1} + f_w(x_{t-1}, x_{t-2}, in_{t,1}, in_{t,2}, in_{t,3}) + \epsilon_t \qquad (13)$$

where $f_w(.)$ in our experiments was a multi-layer perceptron with weight vector $w$.

### 3.2.3 Training

Both compartment model and the neuro-compartment state space model contain parameters which are adapted using the training data set. Although for the compartment model, parameter values could have been derived from prior physiological knowledge, significantly improved performance could be achieved by adapting all parameters using the training data here as well.

We applied the framework of the stochastic nonlinear state space model approach[6] from Section 2 and included the measurement equation

$$y_t = x_t + v_t \qquad (14)$$

to be able to handle noisy and missing blood glucose measurement outputs where $v_t$ is zero-mean Gaussian measurement noise with variance $\sigma_v^2$. For missing measurement we set $\sigma_v^2 \to \infty$.

[6] Equations (12) and (13) can easily be transformed into the state space formulation (1).

Training was performed by applying the EM-type algorithm from Section 2.3.1 with the Gaussian approximations obtained from the extended Kalman filter and smoother (RNN-EKFS) and the Fisher scoring algorithm (RNN-FS) of Section 2.3.2. Correspondingly, we adapted the compartment model using the Fisher scoring algorithm (CM-FS). The training process was started with $S = 50$ samples per time step and the sample size was increased to $S = 250$ when the EM-type algorithm started to converge. Fisher scoring was performed with a stepsize parameter $\eta = 0.07$ and typically needed about $13 - 20$ iterations to converge to the posterior mode estimate. The measurement variance was initialized from prior knowledge about the measurement error levels of typical blood glucose meters which is about 15% [28]. The variance of the noise terms $\epsilon_t$ was set to the above "patient noise" level.

## 4 Experiments

Our data set consists of the protocal of a male type∼I diabetic patient over a period of 63 days with a total of 463 blood glucose measurements. We used the first 42 days of the data set for training the models (containing 312 measurements of the blood glucose) and the following 21 days for testing (containing 151 blood glucose measurements). The blood glucose prediction on the test set was obtained by using the particle filter described in Section 2.4 such that the prediction of a blood glucose value is based on all previously measured blood glucose values in the test set. The particle filter used $S = 500$ samples per time step. Figure 6 shows the explained variance on the test set for different predictive models. The explained variance on the test set is defined [in percent] as $100 \times (1 - \mathrm{MSE_M}/\mathrm{MSE_{mean}})$. Here, $\mathrm{MSE_M}$ is the mean squared prediction error on the test set of the specific model $M$ and $\mathrm{MSE_{mean}}$ is the mean squared one-step ahead prediction error of a trivial model which simply predicts the mean value of the time series.

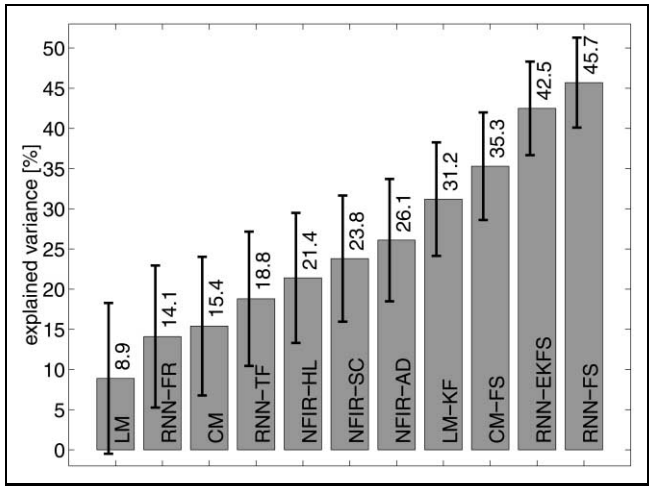


**Figure 6:** The explained variance on the test set for the various approaches. The error bars are calculated based on the $\chi^2$-distribution with 151 degrees of freedom. They indicate the region with 68 % of the probability mass corresponding to the region within two standard deviations in the normal distribution.

The model RNN-EKFS corresponds to the neural state space model trained with the Monte Carlo generalized EM-type algorithm using a Gaussian approximation where center and width of the Gaussian posterior are derived from the extended Kalman filter and smoother algorithm (Section 2.3.2). This model achieved 42.5% explained variance when used for prediction. The RNN-FS model, which uses Fisher scoring instead of the extended Kalman filter (Section 2.3.3), achieved the best prediction performance with an explained variance of 45.7%.

The compartment model shows somewhat worse performance when adapted within a state space model approach using the Fisher scoring posterior approximation (CM-FS): it explains about 10% less variance than its neural network based counterpart.

For comparison, we also show the performance of a *linear* model (LM-KF) which was embedded into a linear state space framework and trained with the EM algorithm given in [16]. This model achieved an explained variance of 31.2%.

We have also included results using techniques not introduced here, but are described in detail elsewhere [26; 29–31].

LM, CM, RNN-FR, and RNN-TF correspond to *deterministic* linear, compartment and neural network models, i.e., neither measurement noise nor process noise is included in the model. In RNN-TF measured blood glucose values are substituted whenever available, otherwise predicted values are used. The bad performance of those models illustrates the need for the inclusion of appropriate error models.

NFIR-HL, NFIR-SC, and NFIR-AD show the results from various approaches where we used a neuro-fuzzy preprocessing step approximating the dynamics of the system by solely relying on past inputs thus avoiding a feedback loop in the model. The results in Figure 6 show that the various recurrent models are superior to these models. The particular neuro-fuzzy architectures and the corresponding training algorithms can be found in [31].

## 5 Evaluation and Conclusions

Nonlinear time-series models with exogenous inputs, embedded into a state space representation and trained with the Monte Carlo generalized EM-type algorithm from Section 2 are powerful tools for blood glucose prediction and outperformed neuro-fuzzy time series models, linear, and nonlinear autoregressive time-series models. The better performance of the neural network based model in comparison to the compartment model might be attributed to the greater flexibility obtained using a neural network model which relies less on prior physiological assumptions than a compartment model. The better results in comparison to the neuro-fuzzy time series model approach might be explained by the fact that autoregressive or recurrent neural networks better represent dynamical systems than models whose predictions solely depend on past inputs. The explained variance of the best model is approximately 46% which corresponds to 50 mg/dl unexplained variations and is slightly better than the inherent patient noise of 54 mg/dl mentioned in [32]. This is a promising result but, by itself, says little about the usefulness of our model.

Future work will concentrate on the question if useful therapy recommendations can be derived from our model. Another important aspect is non-stationarity. There is evidence that insulin sensitivity changes over the course of the day and is typically at its high point in the morning. Furthermore, it is generally assumed that the physiological parameters change with a time constant on the order of days. This might be modeled by slowly changing hidden parameters. Whereas fast changing hidden states are typically modeled as noise (as in our approach) slowly changing hidden states are taken into account by having either an online adaptive system or by explicitly introducing hidden states as in a hidden Markov model. Recent work on switching dynamical systems might be an interesting direction as well. Of course, both for an online adaptive system and for a system with hidden states, the small number of measurements of the blood glucose concentrations might pose a problem. Further work will also focus on developing error models for the input measurements (for example, the calory intake is typically estimated with great uncertainty).

Potential future aims are, first, the development of a system for making recommendations to optimize the patient's therapy, second, or a system which is able to warn the patient of dangerous metabolic states, and third, a system which can be used in the design of a stabilizing control system for blood glucose regulation, a so-called "artificial beta cell" [24].

### References

[1] Thefeld W., „Prävalenz des Diabetes mellitus in der erwachsenen Bevölkerung Deutschlands", *Gesundheitswesen*, Vol. 61, 1999.

[2] Schiffrin A., Mihic M., Leibel B. S., Albisser M., "Computer assisted insulin dosage adjustment", *Diabetes Care*, Vol. 8, 1985.

[3] Chao S. C., Albisser A. M., "The diabetes simulator", in *Decision support for patient management*, British medical informatics society, 1989.

[4] Deutsch T., Carson E. R., Harvey F. E., Lehmann E. D., Sonksen P. H., Tamas G., Whitney G., Williams C. D., "Computer-assisted diabetic management: a complex approach", *Computer methods and programs in biomedicine*, Vol. 32, 1990.

[5] Harvey F. E., Carson E. R., "Diabeta-an expert system for the management of diabetes", in *Objective Medical Decision-Making: System Approach in Disease*, Tsiftsis D. D., Ed., 1986, Springer-Verlag.

[6] Schneider J., Piwernetz K., Engelbrecht R., Renner R., "Diacons-a consultation system to assist in the management of diabetes", in *Expert systems and decision support in medicine*, Rienhoff O. et al., Eds., 1988, Springer.

[7] Salzsieder E., Albrecht G., Fischer U., Rutscher A., Thierbach U., "Computer-aided systems in the management of type I diabetes: the application of a model-based strategy", *Computer methods and programs in biomedicine*, Vol. 32, 1990.

[8] Leaning M. S. Boroujerdi M. A., "A system for compartmental modeling and simulation", *Computer methods and programs in biomedicine*, Vol. 35, 1991.

[9] Cobelli C., Mari A., "Validation of mathematical models of complex endocrine-metabolic systems. A case study on a model of glucose regulation", *in Med. & Biol. Eng. & Comput.*, Vol. 21, pp. 390–399, 1983.

[10] Berger M. P., Gelfand R. A., Miller P. L., "Combining statistical, rule-based and physiologic model-based methods to assist in the management of diabetes mellitus", *Comput. biomed res.*, Vol. 23, 1990.

[11] Prank K., Jürgens C., von der Mühlen A., Brabant G., "Predictive Neural Networks for Learning the Time Course of Blood Glucose Levels from the Complex Interaction of Counterregulatory Hormones", Neural Computation 10, pp. 941–953, 1998.

[12] Briegel T., Tresp, V., "Fisher Scoring and a mixture of Modes Approach for Approximate Inference and Learning in Nonlinear State Space Models", in Kearns M. J., Solla S. A., Cohn D., Eds., *Advances in Neural Information Processing Systems 11*, pp. 403–409, 1999.

[13] Briegel. T., "Neuro-statistical time-series models", Shaker Verlag, Aachen, Berichte aus der Informatik, Dissertation at the Technical University of Munich, 2000.

[14] Briegel. T., Tresp, V., "Dynamic Neural Regression Models", *Discussion Paper 181 des Instituts für Statistik der Ludwig-Maximilians-Universität München*, 2000.

[15] Rumelhart, D., Hinton, G. Williams, R., "Learning internal representations by error backpropagation", in Rumelhart, D. and McClelland, J., Eds., *Parallel Distributed Processing*, MIT Press, 1986.

[16] Shumway R. H., Stoffer D. S., "Time series smoothing and forecasting using the em algorithm", *Technical Report No. 27*, Division of Statistics, UC Davis, 1981.

[17] Anderson B., Moore J., "Optimal Filtering", Prentice-Hall, Englewood Cliffs, NJ, 1979.

[18] Tresp V., Hofmann R., "Missing and Noisy Data in Nonlinear Time-Series Prediction", in Girosi F., Makhoul J., Manolakos E., Wilson E. (Eds.), *Neural Networks for Signal Processing 5*, IEEE catalog number: 95TH8094, 1995.

[19] Puskorius G. V., Feldkamp L. A., "Neurocontrol of nonlinear dynamical systems with Kalman filter trained recurrent networks", *IEEE Transactions on Neural Networks*, Vol. 5, pp. 279–297, 1994.

[20] Sage A., Melsa J., "Estimation Theory with Applications to Communications and Control", McGraw-Hill, New York, NY, 1971.

[21] Fahrmeir L., Kaufmann H., "On Kalman Filtering, Posterior Mode Estimation and Fisher Scoring in Dynamic Exponential Family Regression", *Metrika*, Vol. 38, pp. 37–60, 1991.

[22] Gordon N. J., Salmond D. J., Smith A. F. M., "Novel approach to nonlinear/non-Gaussian Bayesian state estimation", *IEE Proceedings-F*, Vol. 140, pp. 107–113, 1993.

[23] Seber G. A. F., Wild C. J., "Nonlinear Regression", John Wiley & Sons, New York, NY, 1989.

[24] Detschew W., „Theoretische und experimentelle Beiträge zur Messwertverarbeitung und Reglerentwicklung für die künstliche Betazelle", TH Ilmenau, 1990.

[25] Oranien S., „Das Beratungssytem Diabetex – Ein Beitrag zur Verbesserung der ambulanten Betreuung von Diabetikern mit kontinuierlicher subkutaner Insulininfusion", TH Ilmenau, 1989.

[26] Tresp V., Moody J., Delong W.-R., "Neural modeling of physiological processes", *Computational Learning Theory and Natural Learning Systems*, Vol. 2, Hanson S. J., Petsche T., Kearns M., Rivest L. R., Eds., MIT Press, 1994.

[27] Andreassen S., Benn J., Hovorka R., Olesen K., Carson E., "A probabilistic approach to glucose prediction and insulin dose adjustment: description of metabolic model and pilot evaluation study", *Computer Methods and Programs in Biomedicine*, Vol. 41, pp. 153–165, Elsevier Sc. Publ. Ireland, 1994.

[28] Berger M., Jörgens V., „Praxis der Insulintherapie", Springer, 1990.

[29] Tresp V., Briegel T., "A solution for missing data in recurrent neural networks with an application to blood glucose prediction", *Advances in Neural Information Processing Systems*, Vol. 10, M. Jordan, M. Kearns, S. Solla, Eds., MIT Press, 1998.

[30] Tresp V., Briegel T., Moody J., "Neural Network Models for the Blood Glucose Metabolism of a Diabetic", *IEEE Transactions on Neural Networks*, Vol. 10, Nr. 5, pp. 1204–1213, 1999.

[31] Briegel T., Tresp V., "A neuro-fuzzy approach to feedforward modeling of nonlinear time series", *Proceedings of the 5.th International Conference Fuzzy-Neuro Systems '98*, Munich, 1998.

[32] Hejlesen O. K., Andreassen S., Hovorka R., Cavan D. A., "Dias-the diabetic advisory system: an outline of the system and the evaluation results obtained so far", *Computer methods and programs in biomedicine*, Vol. 54, 1997.

Manuscript received: 20th December 2001.

**Dr. Thomas Briegel** has worked in the Department Neural Computation, Siemens AG, Corporate Technology, Munich-Neuperlach between 1996 and 2000. His research was mainly focused on statistical inference and learning methods for analytically intractable time series and regression problems. Since 2000 he is with McKinsey & Company, Inc. He has received a diploma and a doctoral degree in Computer Science from Technische Universität München in 1996 and 2000, respectively. Furthermore, he received diploma degrees in Applied Mathematics and Business Administration from FernUniversität Hagen in 1996 and 1999, respectively.

Address: McKinsey & Company, Inc., Prinzregentenstr. 22, D-80538 München, Fax: + 49-(0)89-5594-8495, E-Mail: Thomas_Briegel@mckinsey.com

**Dr. Volker Tresp** received a Diploma degree in physics from the University of Göttingen, Germany, in 1984 and the M.Sc. and Ph.D. degrees from Yale University, New Haven, CT, in 1986 and 1989, respectively. He joined the central research and development unit of Siemens AG in 1989 where he currently is the head of a research team. In 1994 he was a visiting scientist at the Massachusetts Institute of Technology's Center for Biological and Computational Learning. His main interests include learning systems, in particular neural networks and graphical models.

Address: Siemens AG, Corporate Technology, Department of Information and Communications, Otto-Hahn-Ring 6, D-81739 München, Fax: + 49-(0)89-636-49767,
E-Mail: Volker.Tresp@mchp.siemens.de