

# Neural Network Models for the Blood Glucose Metabolism of a Diabetic

Volker Tresp, Thomas Briegel and John Moody

(to appear in *IEEE Transactions on Neural Networks*, 1999.)

V. Tresp is with Siemens AG, Corporate Technology, Department of Information and Communications, 81730 München, Germany. E-mail: Volker.Tresp@mchp.siemens.de.

T. Briegel is with the Department of Computer Science at the Technische Universität München, Germany. His work is supported by a grant from Siemens AG. E-mail: Thomas.Briegel@mchp.siemens.de.

J. Moody is with the Computer Science Department of the Oregon Graduate Institute, Portland, OR 97291-1000, USA. E-mail: Moody@cse.ogi.edu.

### Abstract

We study the application of neural networks to modeling the blood glucose metabolism of a diabetic. In particular we consider recurrent neural networks and time series convolution neural networks which we compare to linear models and to nonlinear compartment models. We include a linear error model to take into account the uncertainty in the system and for handling missing blood glucose observations. Our results indicate that best performance can be achieved by the combination of the recurrent neural network and the linear error model.

### Keywords

Diabetes Mellitus, Nonlinear Time Series Modeling, Missing Data, EM Algorithm, RTRL Algorithm, Time Series Convolution Neural Network

## I. INTRODUCTION

Diabetes mellitus is one of the most common chronic diseases with approximately 16 million affected people in the United States alone [1]. In type I diabetes, the disease is caused by the failure of the pancreas to produce sufficient 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. Currently, there are a number of computer based systems commercially available where the patient can enter insulin injections, blood glucose measurements and meals and which store and display this information. These systems are very useful for self-monitoring and for the consultation with the physician. Such computer-based systems are potentially more useful if they would be able to analyze past therapy, to predict future blood glucose levels and to provide therapy recommendations. So far, systems which provide therapy recommendations are either based on expert systems or are based on physiological models [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12]. A promising approach

is pursued by Andreassen, Hejlesen and co-workers [13], [14] who use a causal probabilistic network to model the glucose metabolism and to derive therapy recommendations. So far, none of the systems have gained widespread acceptance in therapy. Hejlesen, Andreassen, Hovorka, and Cavan [14] attribute this fact to the major problems associated with the many facets of evaluating a decision support system in diabetes.

The goal of the work presented in this paper is to study the application of neural networks to modeling the blood glucose metabolism. We expect that neural networks are particular suitable models since the physiological interactions are multi-dimensional, highly nonlinear, stochastic, time variant and patient specific. We will provide experimental results using different neural network architectures. For comparison, we also provide experimental results using linear models and a compartment model. Linear models supply a standard to which more sophisticated approaches must be compared and compartment models are the most common modeling technique for physiological systems.

In the next section we briefly describe some of the fundamentals of diabetes mellitus. In Section III we introduce the different models which were used in the experiments. In particular we describe the different neural network approaches and the compartment model. Furthermore, the importance of including a suitable noise model is discussed. In Section IV we describe the experiments and the experimental results and in Section V we present our conclusions.

## II. DIABETES MELLITUS

Fig. 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 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

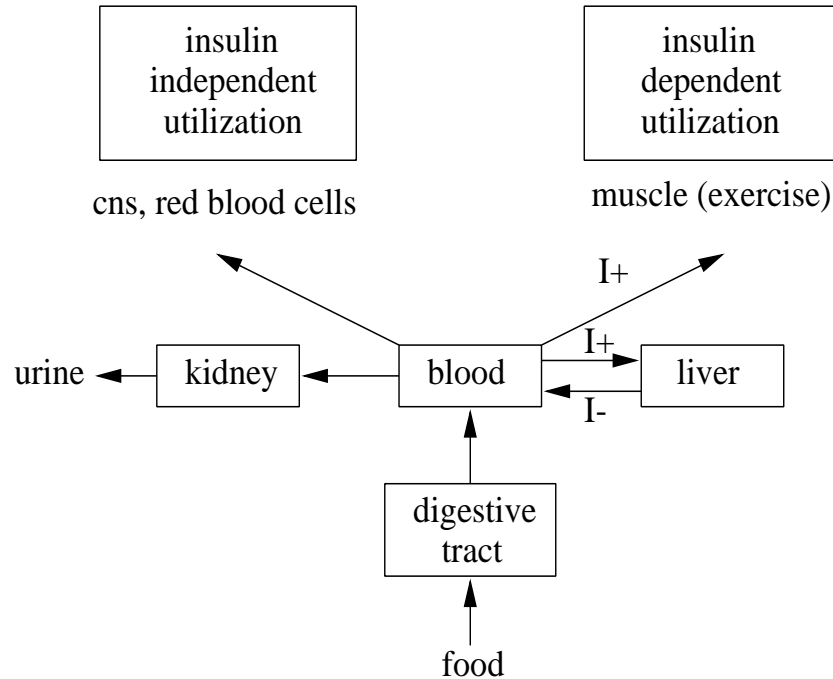


Fig. 1. A simplified model of the glucose metabolism. The arrows indicate the transport of glucose. I+ and I- indicate glucose transports which are promoted, respectively inhibited, by insulin in the blood.

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 Fig. 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  $y_t$  (measured a few times a day) were recorded with 15 minute time resolution. The  $u_{t,j}, j = 1, \dots, 7$  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. Fig. 2 shows a short time window (25 hours) of our data set.

### III. MODELING THE GLUCOSE METABOLISM

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

#### A. *Compartment Models (CM)*

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. We have formulated a compartment model using the different compartments depicted in the simplified model of the glucose metabolism of Fig. 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 [15], [16], [17]. Let  $v_{t,j}$  describe the

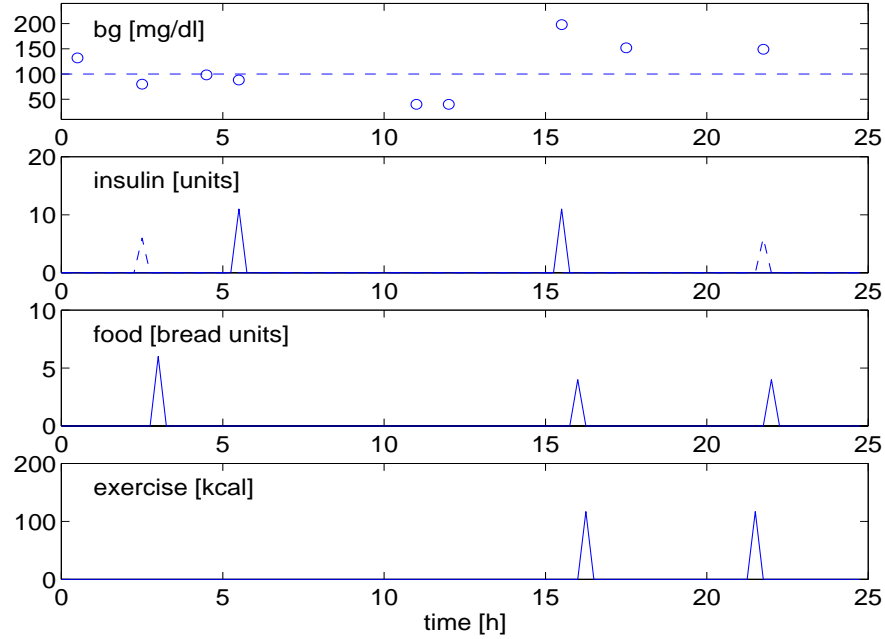


Fig. 2. Short time window (25 hours) of the data set. The top part shows the blood glucose measurements (unfilled circles) in mg/dl. The average blood glucose level of a normal person is indicated by the dashed line. The other plots show insulin injections (basal insulin  $u_{t,1}$  (dashed peaks) and 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 11h and 12h the patient stopped to administer insulin. As a consequence the patient had a case of hyperglycemia at 16h and he reacted with immediately injecting 20 units of insulin.

effect at time  $t$  of past inputs  $\{u_{\tau,j}\}_{\tau=0}^t$ . The response  $v_{t,2}$  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 [15]

$$v_{t,2} = \sum_{\tau=0}^t g_2(t - \tau) u_{\tau,2} \quad (1)$$

with

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

Fig. 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

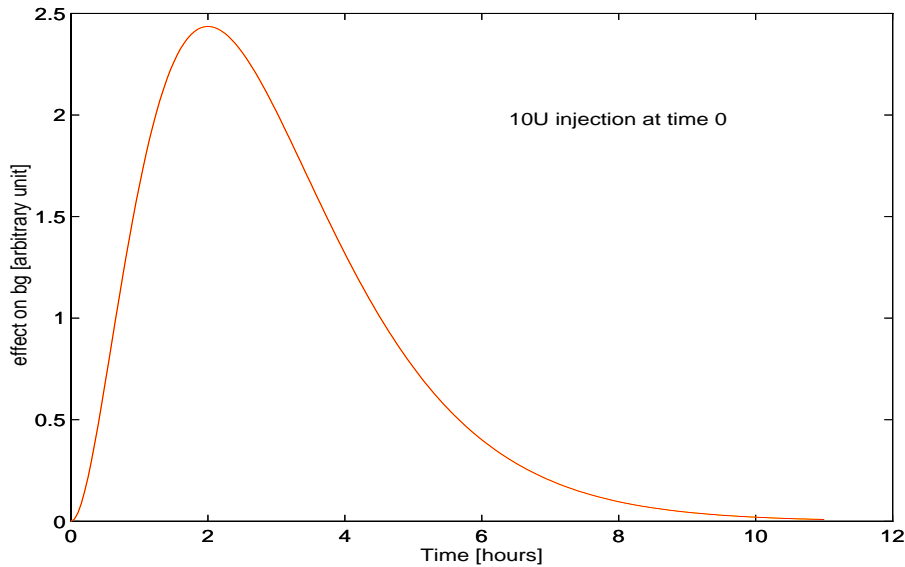


Fig. 3. The plot shows the effect of insulin on the blood glucose level as a consequence of injecting 10 units of normal insulin at time  $\tau = 0$  as simulated by our preprocessing module using impulse response  $g_2(t - \tau)$ . This time behavior is following approximately the concentration of insulin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

time-dependent effects for the digestive tract and for exercise are less well known. In our experiments we followed Detschew [15] 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  $u_t = (u_{t,1}, \dots, u_{t,7})^\top$  to  $v_t = (v_{t,1}, \dots, v_{t,7})^\top$  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 difference equation of the form

$$y_t = y_{t-1} + c_1(v_{t,3} + v_{t,4} + v_{t,5}) + c_2(e^{-c_3(v_{t,1} + v_{t,2})} - c_4 y_{t-1}) - c_5(v_{t,1} + v_{t,2})(y_{t-1} + c_6) - c_7 \sqrt{y_{t-1}} - c_8 y_{t-1}^3 - c_9(v_{t,6} + v_{t,7}) \quad (3)$$

where  $y_t$  is the blood glucose at time  $t$ .<sup>1</sup> This nonlinear difference equation was derived from parameterizing published data describing the dependencies [13], [18], [15], [16], [19]. 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

<sup>1</sup>As mentioned above, the time resolution of our system is 15 minutes.

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 describes the blood glucose lowering effect of exercise. Fig. 4 shows how the functional forms in equation (3) were obtained by fitting nonlinear maps to published data.

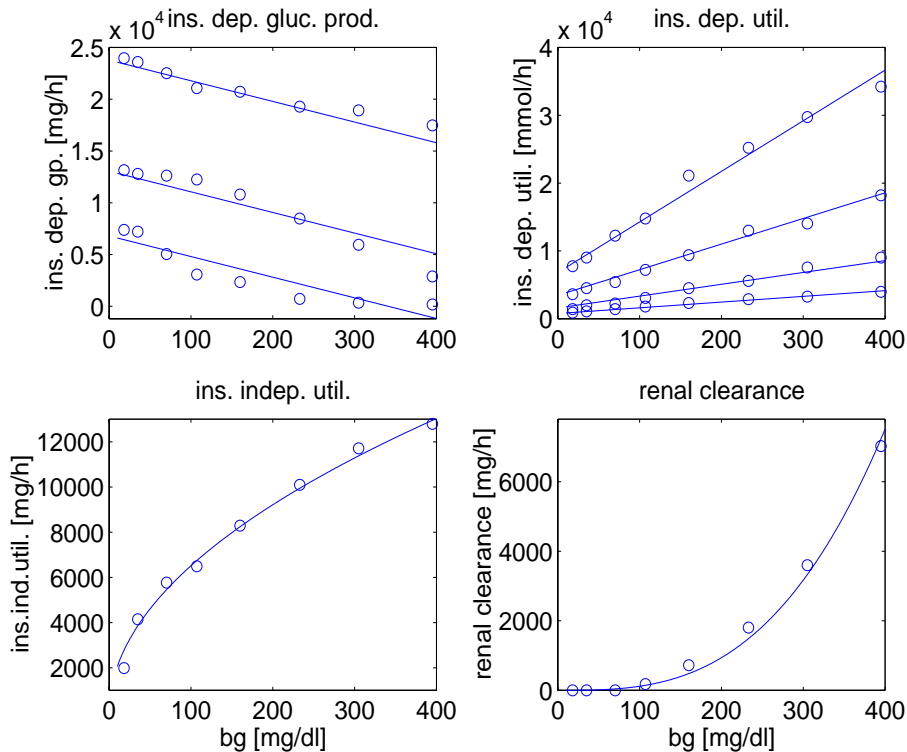


Fig. 4. The plots show the rate of change of the blood glucose as a result of the insulin dependent glucose production (term three in equation (3)), the insulin dependent utilization (term four), the insulin independent glucose removal (term five), and renal clearance (term six). The small circles indicate published data [13] and the solid lines show the fitted parameterization used in equation (3). The plots for insulin dependent glucose production and utilization, respectively are shown for three (from top to down:  $0\mu\text{U/ml}$ ,  $10\mu\text{U/ml}$ ,  $20\mu\text{U/ml}$ ), respectively four (from top to down:  $80\mu\text{U/ml}$ ,  $40\mu\text{U/ml}$ ,  $20\mu\text{U/ml}$ ,  $10\mu\text{U/ml}$ ) different levels of active insulin in the blood.

The functional block described by equation (3) is considered a second compartment module  $\mathcal{M}_n$ . Fig. 5 shows the overall compartment system containing the functional blocks  $\mathcal{M}_d$  and  $\mathcal{M}_n$ .

In the experiments we 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 [13], [18], [15], [16], [19] (Fig. 4 shows the parametrization for some of the terms). Improved performance could be achieved by adapting all parameters using the training data.<sup>2</sup> More details about the training process follow in Section IV.

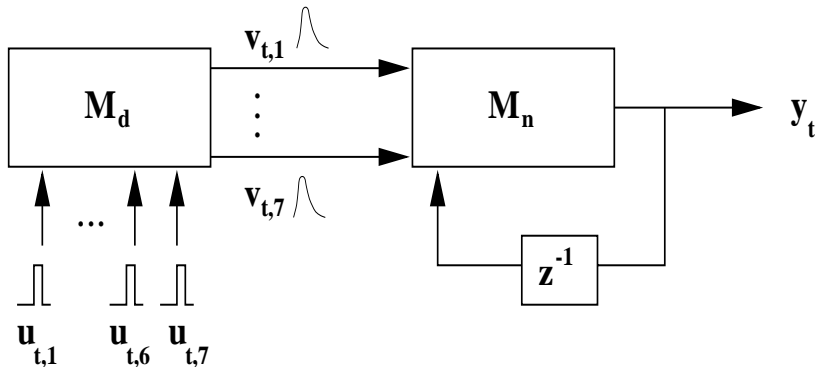


Fig. 5. A compartment model approach, consisting of two compartment modules  $\mathcal{M}_d$  and  $\mathcal{M}_n$  ( $z^{-1}$  indicates a delay of one time step).

### B. Recurrent Neural Network Models (RNN)

Based on standard medical literature on diabetes [21] 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  $a_j, b_j$ .

The nonlinear difference equation (3) 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 as shown in Fig. 6. The five inputs to the network are insulin ( $in_{t,1} = v_{t,1} + v_{t,2}$ ), food ( $in_{t,2} = v_{t,3} + v_{t,4} + v_{t,5}$ ), exercise ( $in_{t,3} = v_{t,6} + v_{t,7}$ ) and the current and previous estimates of the blood glucose. To be specific, the second order <sup>3</sup>

<sup>2</sup>Note that since the blood glucose prediction is fed back as an input to the system, a recurrent learning rule has to be used for adaptation. We applied the real time recurrent learning rule (RTRL) [20] using a quadratic error-function, see Appendix A.

<sup>3</sup>The second order RNN model outperformed higher order RNN models in the experiments.

nonlinear neural network model is

$$y_t = y_{t-1} + f_w(y_{t-1}, y_{t-2}, in_{t,1}, in_{t,2}, in_{t,3}) \quad (4)$$

where  $f_w$  in our experiments was a feedforward multi-layer perceptron (MLP) with weight vector  $w$ . Note that the neural network is used in a recurrent fashion since previous predictions are used as inputs.

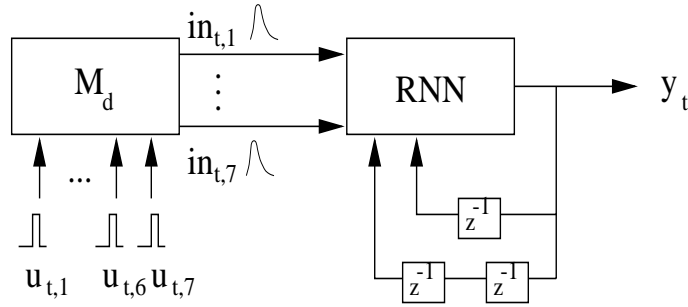


Fig. 6. A neuro-compartment model approach, consisting of a recurrent neural network and the compartment module  $\mathcal{M}_d$ .

In the first experiment with a recurrent neural network (RNN-FR) we estimate the blood glucose at time  $t$  as the output of the neural network  $\hat{y}_t = y_t$ . The neural network is used in the free running mode in which the network predictions are iterated for training and prediction. We use RTRL to both adapt the weights in the neural network as well as all parameters in  $\mathcal{M}_d$ .

In the second experiment we used the teacher forced model (RNN-TF). This experiment is identical to the previous one except that —both for training and prediction— measurements of the blood glucose concentration are substituted for the predicted values whenever available.

Details of the learning algorithm are described in Appendix A.

### C. Time Series Convolution Neural Network Models (TSCNN)

To motivate the following approach, consider the output of a single-input ( $x_t$ ) and single-output ( $y_t$ ) linear time-invariant causal system which can be described by the convolution

$$y_t = \sum_{\tau=0}^{\infty} g_{\tau} x_{t-\tau} \approx \sum_{\tau=0}^T g_{\tau} x_{t-\tau}. \quad (5)$$

The latter approximation is referred to as the finite impulse response (FIR) filter [22] in linear filtering theory. Similarly, a suitable model for a nonlinear time-invariant causal system is

$$y_t = \text{NN}_t(x_t, x_{t-1}, \dots) \approx \text{NN}(x_t, \dots, x_{t-T}) \quad (6)$$

where  $\text{NN}_t(\cdot)$  would be a (unrealizable) neural network with an infinite number of inputs and  $\text{NN}(\cdot)$  is neural network with  $T + 1$  inputs. Note that the latter is *not* the well known nonlinear autoregressive model since the inputs to the neural network are previous inputs to the system and are not past values of the time series itself. The latter approximation is justified if the system does not depend sensitively on inputs which occurred a long time period in the past, i.e. if it has finite memory (a counter example would be a chaotic system). We shall call this system the time series convolution neural network (TSCNN). The name is derived from a related neural network architecture for classification problems known as the convolution neural network (LeCun *et al.* [23]). Our approach is appropriate for time series modeling purposes. Note that past measurements of  $y_t$  are not represented as inputs of the system, such that this approach is particularly suitable if these are not always available (as in our application) and one wants to avoid recurrent learning rules.

A problem with this approach is that relevant inputs might have occurred a long time period in the past (in our application, up to 24 hours). Since the time resolution of our system is relatively fine (15 minutes) we apply a suitable preprocessing step in the next section to avoid ending up with a system with a large input space resulting from the approximation in equation (6). Otherwise, training such a system would easily lead to overfitting.

### C.1 Hard Limited Time Windows (TSCNN1-HL)

Our first approach (TSCNN-HL) is to divide the input space into different time windows as indicated in Fig. 7. First we group the *raw* inputs in  $(U_{t,1} = u_{t,1})$ , normal (soluble) insulin injections  $(U_{t,2} = u_{t,2})$ , food  $(U_{t,3} = u_{t,3} + u_{t,4} + u_{t,5})$  and exercise  $(U_{t,4} = u_{t,6} + u_{t,7})$ . Let  $t - E_{i,j}$  be the end of the  $j$ -th time window for grouped input  $i$ . We integrate the

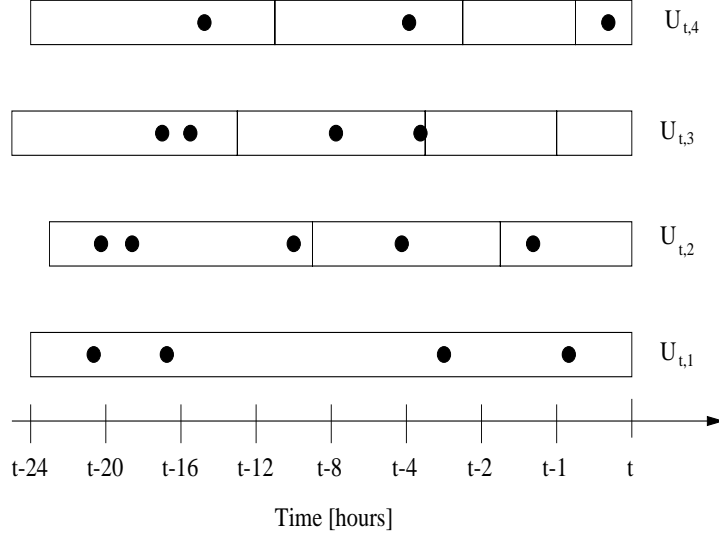


Fig. 7. The figure shows the hard limited time windows for a blood glucose measurement at time  $t$ . As illustrative example, the filled circles indicate at which instances in time an input  $U_{t,i}$  might be measured.

inputs in the respective time windows to form

$$act_{i,j}(t) = \sum_{\tau=t-E_{i,j+1}}^{t-E_{i,j}} U_{\tau,i}. \quad (7)$$

The sizes of the time windows are indicated in Fig. 7 and were chosen following the consultation with a physician. Note that we choose only one time window for basal insulin which corresponds to its slow and constant effect. Also note that the sizes of the time windows corresponding to recent events are shorter.

Overall, we end up with a system with 12 inputs.

The  $act_{i,j}(t)$  form the inputs to the time series convolution neural network (TSCNN) which is realized as an MLP parameterized with weight vector  $w$  with one hidden layer. The training was performed with the Quasi-Newton BFGS-method [24] using the quadratic error-function

$$E = \frac{1}{2} \sum_m \left( y_m^{bg} - \text{TSCNN}_w(act_{i,j}(t_m)) \right)^2 \quad (8)$$

on the training set with an additional weight-decay term  $-\alpha \|w\|^2$  which was tuned to avoid overfitting using cross validation.  $\text{TSCNN}_w(\cdot)$  is the response of the TSCNN and the  $y_m^{bg}$  are the blood glucose measurements at times  $t_m$ .

## C.2 Soft Competitive Fixed Time Windows (TSCNN1-SC)

If we make the assumption – like in the last section – that the *precise* time of a past input is irrelevant and it is, for example, sufficient to specify if the past input occurred *very recently*, *recently*, *in the intermediate past*, or *long ago*, then it is natural to introduce basis functions which “fuzzify” the exact instance in time when a past input occurred [25]. The basis functions are implemented as Gaussians of the form

$$b_{i,j}(u) = \exp\left[-\frac{1}{2}\left(\frac{u - \mu_{i,j}}{\sigma_{i,j}}\right)^2\right]. \quad (9)$$

We derived the widths  $\sigma_{i,j}$  and centers  $\mu_{i,j}$  of the basis functions from the time windows used in the previous section (Section III-C.1). The overlapping basis functions “compete” for the input leading to the normalized weighting function

$$n_{i,j}(u) = \frac{b_{i,j}(u)}{\sum_{k=1}^{M_i} b_{i,k}(u)}. \quad (10)$$

The input  $act_{i,j}(t)$  to the TSCNN at time  $t$  is obtained by weighting the amplitudes of the past (raw) input time series  $U_{t,i}$  at time  $\tau$  by  $n_{i,j}(t - \tau)$  over a certain time window  $[t - M_{i,j}^b, t - M_{i,j}^e]$ :

$$act_{i,j}(t) = \sum_{\tau=t-M_{i,j}^b}^{t-M_{i,j}^e} U_{\tau,i} n_{i,j}(t - \tau) \quad (11)$$

where  $t - M_{i,j}^b$  is the beginning,  $t - M_{i,j}^e$  is the end<sup>4</sup> of the  $j$ -th soft time window for grouped input  $U_{t,i}$ . Fig. 8 shows an example of the preprocessing step using four normalized weighting functions  $n_{3,1}(t - \tau), \dots, n_{3,4}(t - \tau)$  for grouped input  $U_{t,3}$ .

If compared to the hard limited time windows in the last section, using soft time windows has two potential benefits. First, soft time windows provide the TSCNN with better knowledge about the time of an input event. For example, if neighboring basis functions are active, the event must have occurred somewhere close to the border of the two time windows. Secondly, since the time windows now overlap, a larger number of inputs to the TSCNN are unequal to zero at any given time providing the neural network with a more informative and distributed view on the inputs. In the following the preprocessing step using the soft competitive time windows in combination with the TSCNN is referred to

<sup>4</sup>We choose  $M_{i,j}^b > M_{i,j}^e \geq 0$  so, that  $b_{i,j}(t - M_{i,j}^b) < \epsilon$  and  $b_{i,j}(t - M_{i,j}^e) < \epsilon$  with a sufficiently small  $\epsilon > 0$ .

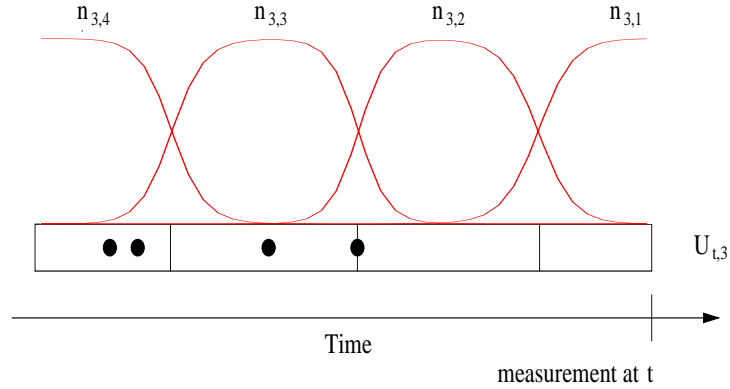


Fig. 8. Soft competitive time windows  $n_{3,1}(t - \tau) = \text{“very recently”}$ ,  $\dots$ ,  $n_{3,4}(t - \tau) = \text{“long ago”}$  implemented using normalized RBFs for time series  $U_{t,3}$  (food).

as TSCNN1-SC. Recall that the only difference to the approach described in the previous section (TSCNN1-HL) is that the preprocessing to the inputs of the neural network has changed.

### C.3 Soft Competitive Adaptive Time Windows (TSCNN1-AD)

This is the same approach as in the last section except that the parameters of the time windows  $\{\sigma_{i,j}, \mu_{i,j}\}$  are also adapted using backpropagation (TSCNN1-AD). The computation of the gradients with respect to the widths  $\sigma_{i,j}$  and centers  $\mu_{i,j}$  can be found in [25].

### C.4 Including Past Blood Glucose Measurements (TSCNN2)

The TSCNN2 model is identical to the TSCNN1 model, except that two additional inputs are used,

- the blood glucose concentration of the last measurement  $y_{m-1}^{bg}$ ,
- the time which has passed since the last measurement occurred.

The neural network now has 14 inputs. Note that the neural network is now provided with information about past measurements of the blood glucose.

### D. Including a Linear Error Model

Two of the major problems in this application are that first, blood glucose is only measured a few times every day such that most blood glucose measurements are missing

and second, that the system is highly stochastic: it has been estimated that the standard deviation of the residual error which cannot be explained by the inputs (“patient noise”) is around 54 mg/dl, which is substantial if one considers that the average blood glucose level of a healthy person is around 100 mg/dl [14]. This estimate was achieved by observing the variations in blood glucose levels of a diabetic on different days with identical diets, insulin injections and activities. It is therefore reasonable to assume that the inclusion of a proper error model might lead to better predictions. In our approach we added a linear error model which is described in the following paragraphs. The linear error model allows us to apply EM learning rules such that we can avoid to solve the complex integrals required for more general stochastic nonlinear dynamic models [26]. Consider the model with state updates <sup>5</sup>

$$y_t^* = f_w(y_{t-1}^*, \dots, y_{t-N}^*) \quad (12)$$

$$x_t = \sum_{i=1}^K \theta_i x_{t-i} + \epsilon_t \quad (13)$$

$$y_t = y_t^* + x_t = f_w(y_{t-1}^*, \dots, y_{t-N}^*) + \sum_{i=1}^K \theta_i x_{t-i} + \epsilon_t \quad (14)$$

and with measurement equation

$$z_t = y_t + \delta_t \quad (15)$$

where  $\epsilon_t$  and  $\delta_t$  denote uncorrelated additive noise. The variable of interest  $y_t$  is now the sum of the deterministic response of the deterministic model  $y_t^* = f_w(\cdot)$ <sup>6</sup> and the output of a linear system error model  $x_t$ .  $z_t$  is a noisy measurement of  $y_t$ . In particular we are interested in the special cases that  $y_t$  can be measured with certainty (variance of  $\delta_t$  is zero) or that a measurement is missing (variance of  $\delta_t$  is infinity). The nice feature is now that  $y_t^*$  can be considered a deterministic input to the state space model consisting of the equations (13)–(14). This means that for optimal one-step or multiple-step prediction, we can use the *linear* Kalman filter for equations (13)–(14) and measurement equation (15) by treating  $y_t^*$  as deterministic input. Similarly, to train the parameters in the linear part of the system (i.e.  $\{\theta_i\}_{i=1}^N$ ) we can use an EM adaptation rule, implemented using

<sup>5</sup>For simplicity, we leave out in the description the input measurements which are irrelevant for the discussion.

<sup>6</sup> $f_w(\cdot)$  stands for any of the deterministic approaches described in the previous sections.

forward-backward Kalman filter equations (see the Appendix B). The deterministic model is adapted with the residual error which cannot be explained by the linear model, i.e.  $target_{t_m}^{RNN} = y_m^{bg} - \hat{y}_{t_m}^{LEM}$  where  $y_m^{bg}$  is a blood glucose measurement time  $t_m$  and where  $\hat{y}_{t_m}^{LEM}$  is the estimate of the linear model at time  $t_m$ . After the deterministic model is adapted, the linear model can be retrained using the residual error which cannot be explained by the neural network, then again the deterministic model is retrained and so on until no further improvement can be achieved. The advantage of this approach is that all of the nonlinear interactions are modeled by a deterministic model which can be trained deterministically. The linear model is responsible for the noise model which can be trained using powerful learning algorithms for linear systems. The constraint is that the error model cannot be nonlinear which often might not be a major limitation.

#### IV. EXPERIMENTAL RESULTS

The data set consists 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).

Fig. 9 shows the explained variance of the test set for different predictive models. The explained variance on the test set is defined [in percent] as  $100 \cdot \left(1 - \frac{MSPE(model)}{MSPE(mean)}\right)$ . Here,  $MSPE(model)$  is the mean squared prediction error on the test set of the specific model and  $MSPE(mean)$  is the mean squared prediction error of predicting the mean.

In the first experiment (CM) we estimate the blood glucose at time  $t$  as the output of the compartment model depicted in Fig. 5 and described in Section III-A. We use RTRL for both training the parameters in the compartment model  $\mathcal{M}_d$  (the  $\{a_j, b_j\}_{j=1}^7$  in the response functions  $h_j(\cdot)$ ), as well as the parameters in the compartment model  $\mathcal{M}_n$  (the parameters  $\{c_j\}_{j=1}^9$  in the difference equation (3)). The CM model explains 15.4 percent of the variance.

In the next experiment (RNN-FR) we estimate the blood glucose at time  $t$  as the output of the recurrent neural network as described in Section III-B. The neural network is used in the free running mode for training and prediction. We use RTRL to both adapt the weights in the neural network as well as all parameters in the response functions  $h_j(\cdot)$ .

The RNN-FR model explains 14.1 percent of the variance which is very comparable to the compartment model.

The RNN-TF model is identical to the previous experiment except that measurements are substituted whenever available (Section III-B). RNN-TF could explain more of the variance (18.8%) than the RNN-FR. This indicates that by substituting estimated blood glucose levels with measured ones, information about past measurements of the blood glucose is exploited in the prediction.

The model RNN-LEM2 (error model with order 2) corresponds to the combination of the recurrent neural network and the linear error model as introduced in Section III-D. As indicated in Fig. 9, the RNN-LEM2 model achieves the best prediction performance with an explained variance of 44.9%. The RNN-LEM1 model achieved almost equal performance with an explained variance of 43.7%. Note the considerable improvement in performance through the inclusion of the error models. The compartment model also showed a considerable improvement by the inclusion of the error model (CM-LEM1) but explains only approximately 10% less variance than the recurrent neural network with error models. As a comparison, we show the performance of just the linear error model LEM1 (this model ignores all inputs), a linear model (LM-FR) without an error model trained with RTRL and a linear model with an error model (LM-LEM1). Interestingly, the linear error model which does not see any of the inputs can explain more variance (12.9%) than the LM-FR model (8.9%). The LM-LEM1 model, which can be considered a combination of both, can explain more than the sum of the individual explained variances (31.5%) which indicates that the combined training gives better performance than training both submodels individually.

The first experiments based on the TSCNN used the TSCNN with hard limited time windows (TSCNN1-HL, Section III-C.1), the TSCNN with soft competitive time windows (TSCNN1-SC, Section III-C.2) and the TSCNN with soft competitive time windows where the parameters in the soft windows were adapted (TSCNN1-AD, Section III-C.3). Shown are just the results of the TSCNN1-AD approach which obtained the best results of the three. It explains 16.3 percent of the variance.

Recall that the TSCNN2 approaches are identical to the TSCNN1 approaches except

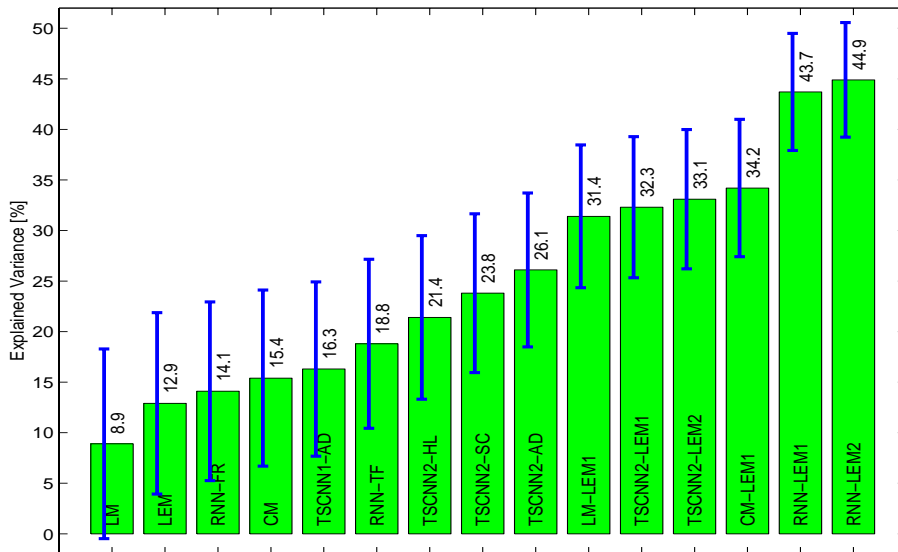


Fig. 9. The explained variance on the test set for the various approaches. The error bars are calculated based on the  $\chi^2$ -distribution with 151 degrees of freedom. They indicate the region with 68% of the probability mass corresponding to the region within two standard deviations in the normal distribution.

that two additional inputs are used: the first is the last blood glucose measurement  $y_{m-1}^{bg}$  and the second is the time delay between the actual and the last blood glucose measurement (Section III-C.4). The experiments show that this modification improves the performance. Shown are the results for the TSCNN2-HL, TSCNN2-SC and the TSCNN2-AD approaches which explained 21.4, 23.8 and 26.1 percent of the variance, respectively. The combination of the TSCNN2 models with the linear error model on the other hand did not lead to a considerable improvement. The reason is of course that the TSCNN2-models are already informed about the latest measurement which is otherwise supplied by the error model.

We would like to remark that the nonlinear models (CM, RNN-FR, RNN-TF, RNN-LEM, TSCNN1, TSCNN2) give considerably better results than their linear counterparts, confirming that the system is highly nonlinear.

It is worth noting that the training of the combined approaches (nonlinear neural model combined with linear error model) leads to a faster convergence than the adaptation of the neural network without the linear error model.

## V. CONCLUSIONS

In our paper we considered a number of models for predicting the blood glucose of a diabetic patient. Our experiments showed that the inclusion of a proper error model improves performance considerably. In combination with the linear error model the recurrent neural network is a powerful model for blood glucose prediction and gave best results and outperformed both a compartment model and the time series convolution neural network approach. The better performance of the recurrent neural networks 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 time series convolution neural network approach might be explained by the fact that recurrent neural networks better represent dynamical systems than models whose predictions solely depend on past inputs.

The time constant of the linear error model is slightly less than one hour. Consequently, one hour is also the time constant with which our system “forgets” the last blood glucose measurement. This is in agreement with the observation of other researchers that the blood glucose system is self stabilizing with approximately a time constant of one hour. Since in their work, Andreassen, Hejlesen and co-workers are only interested in predictions over a time horizon of more than one hour [13], [14], these authors never use past blood glucose measurements in the prediction of their models. Our results on the other hand indicate that for neural networks, the inclusion of an error model which responds to past measurements improves prediction accuracy considerably. This might be explained by the fact that neural networks, in contrast to causal probabilistic networks, are inherently deterministic and benefit from incorporating a proper stochastic error model. The explained variance of the best model is 45% which corresponds to 51 mg/dl and is slightly better than the inherent patient noise of 54 mg/dl mentioned in Section III-D. 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 nonstationarity. 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 physiology changes with a time constant on the order of days.

This might be due to 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.

The overall future aim is the development of a system for making recommendations to optimize the patient’s therapy or a system which is able to warn the patient of dangerous metabolic states, or finally, a system which can be used in the design of a stabilizing control system for blood glucose regulation, a so-called “artificial beta cell” [17].

Based on the prediction and the estimated variance, given by the linear Kalman filter in the error model, we can derive error bars for the prediction and therefore it will be possible to do a risk analysis for the diabetic which enables us to realize a warning system for dangerous metabolic states. Further work will also focus on developing error models for the input measurements (for example, the number of food calories are typically estimated with great uncertainty).

#### ACKNOWLEDGEMENTS

We thank Michael Ritter, Department of Medicine II, Ludwig Maximilians Universität München, Germany, for valuable discussions and Wolf-Rüdiger Delong, Siemens AG, Erlangen, Germany, for providing us with diabetic data.

#### APPENDIX A: RTRL ADAPTATION RULES

Let  $u$  be a generic weight in the weight vector  $w$  of the neural network  $f_w$ ,  $c$  be a generic parameter in the compartment module  $\mathcal{M}_n$ ,  $v$  be a generic parameter in the compartment module  $\mathcal{M}_d$ ,  $z^{\text{RNN}} = (u, v)^\top$  and  $z^{\text{CM}} = (c, v)^\top$ . The adaptation rule for  $z^{\text{RNN}}$  and  $z^{\text{CM}}$  is gradient descent in the error  $E^{\text{RNN}} = \sum_m (y_m^{bg} - y_{t_m}^{\text{RNN}})^2$  and  $E^{\text{CM}} = \sum_m (y_m^{bg} - y_{t_m}^{\text{CM}})^2$ , respectively

$$\Delta z^{\text{RNN}} \propto -\frac{\partial E^{\text{RNN}}}{\partial z^{\text{RNN}}} = \sum_m (y_m^{bg} - y_t^{\text{RNN}}) \frac{\partial y_t^{\text{RNN}}}{\partial z^{\text{RNN}}} \quad (16)$$

$$\Delta z^{\text{CM}} \propto -\frac{\partial \mathbf{E}^{\text{CM}}}{\partial z^{\text{CM}}} = \sum_m (y_m^{\text{bg}} - y_{t_m}^{\text{CM}}) \frac{\partial y_t^{\text{CM}}}{\partial z^{\text{CM}}} \quad (17)$$

where  $y_m^{\text{bg}}$  is the measured blood glucose level at time  $t_m$ ,  $y_{t_m}^{\text{RNN}}$  is the neuro-compartment model glucose value at time  $t_m$  and  $y_{t_m}^{\text{CM}}$  is the compartment model glucose value at time  $t_m$ . The sensitivities  $\frac{\partial y_t^{\text{RNN}}}{\partial z^{\text{RNN}}}$  and  $\frac{\partial y_t^{\text{CM}}}{\partial z^{\text{CM}}}$  obey difference equations obtained by differentiating equation (3) and (4) respectively with respect to  $z$ . For network weights  $u$  we obtain

$$\frac{\partial y_t^{\text{RNN}}}{\partial u} = \frac{\partial y_{t-1}^{\text{RNN}}}{\partial u} + \frac{\partial f_w}{\partial u} + \frac{\partial f_w}{\partial y_{t-1}^{\text{RNN}}} \frac{\partial y_{t-1}^{\text{RNN}}}{\partial u} + \frac{\partial f_w}{\partial y_{t-2}^{\text{RNN}}} \frac{\partial y_{t-2}^{\text{RNN}}}{\partial u} \quad (18)$$

and for the compartment module parameters  $v$  in the neuro-compartment approach we get

$$\frac{\partial y_t^{\text{RNN}}}{\partial v} = \frac{\partial y_{t-1}^{\text{RNN}}}{\partial v} + \frac{\partial f_w}{\partial y_{t-1}^{\text{RNN}}} \frac{\partial y_{t-1}^{\text{RNN}}}{\partial v} + \frac{\partial f_w}{\partial y_{t-2}^{\text{RNN}}} \frac{\partial y_{t-2}^{\text{RNN}}}{\partial v} + \sum_{j=1}^3 \frac{\partial f_w}{\partial \text{in}_{t,j}^{\text{RNN}}} \frac{\partial \text{in}_{t,j}^{\text{RNN}}}{\partial v}. \quad (19)$$

For the parameters  $c$  in the compartment module  $\mathcal{M}_n$ , we obtain

$$\frac{\partial y_t^{\text{CM}}}{\partial c} = \frac{\partial \mathcal{M}_n}{\partial c} + \frac{\partial \mathcal{M}_n}{\partial y_{t-1}^{\text{CM}}} \frac{\partial y_{t-1}^{\text{CM}}}{\partial c} \quad (20)$$

and for parameters  $v$  in  $\mathcal{M}_d$  we derive

$$\frac{\partial y_t^{\text{CM}}}{\partial v} = \frac{\partial \mathcal{M}_n}{\partial v} + \frac{\partial \mathcal{M}_n}{\partial y_{t-1}^{\text{CM}}} \frac{\partial y_{t-1}^{\text{CM}}}{\partial v} + \sum_{j=1}^7 \frac{\partial \mathcal{M}_n}{\partial v_{t,j}^{\text{CM}}} \frac{\partial v_{t,j}^{\text{CM}}}{\partial v}. \quad (21)$$

During training, the sensitivities  $\frac{\partial y_t^{\text{RNN}}}{\partial z^{\text{RNN}}}$  ( $\frac{\partial y_t^{\text{CM}}}{\partial z^{\text{CM}}}$ ) can be unstable and become large. Therefore, we introduce additional decay terms on the right sides of equations (18)-(19) and (20)-(21) of the form  $-\alpha \frac{\partial y_{t-1}^{\text{RNN}}}{\partial z^{\text{RNN}}}$  ( $-\alpha \frac{\partial y_{t-1}^{\text{CM}}}{\partial z^{\text{CM}}}$ ). This guarantees that only events with a certain temporal proximity influence the training, and earlier events are forgotten exponentially in time.

## APPENDIX B: EM ADAPTATION RULES FOR TRAINING THE LINEAR ERROR MODEL

Model and observation equations of a general model are<sup>7</sup>

$$x_t = \Theta x_{t-1} + \epsilon_t \quad z_t = M_t x_t + \delta_t. \quad (22)$$

where  $\Theta$  is the  $K \times K$  transition matrix of the  $K$ -order linear error model. The  $K \times 1$  noise terms  $\epsilon_t$  are zero-mean uncorrelated normal vectors with common covariance matrix  $Q$ .

<sup>7</sup>Note that any linear system of order  $K$  can be transformed into a first order linear system of dimension  $K$ .

$\delta_t$  is  $m$ -dimensional <sup>8</sup> zero-mean uncorrelated normal noise vector with covariance matrix  $R_t$ . Recall that we consider certain measurements and missing values as special cases of noisy measurements. The initial state of the system is assumed to be a normal vector with mean  $\mu$  and covariance  $\Sigma$ . We describe the EM equations for maximizing the likelihood of the model. Define the estimated parameters at the  $(r + 1)$ st iterate of EM as the values  $\mu, \Sigma, \Theta, Q$  which maximize

$$G(\mu, \Sigma, \Theta, Q) = E_r(\log L | z_1, \dots, z_n) \quad (23)$$

where  $\log L$  is log-likelihood of the complete data  $x_0, x_1, \dots, x_n, z_1, \dots, z_n$  and  $E_r$  denotes the conditional expectation relative to a density containing the  $r$ th iterate values  $\mu(r), \Sigma(r), \Theta(r)$  and  $Q(r)$ . Recall that missing targets are modeled implicitly by the definition of  $M_t$  and  $R_t$ . For calculating the conditional expectation defined in (23) the following set of recursions are used (using standard Kalman filtering results, see [27]). First, we use the forward recursion <sup>9</sup>

$$\begin{aligned} x_t^{t-1} &= \Theta x_{t-1}^{t-1} \\ P_t^{t-1} &= \Theta P_{t-1}^{t-1} \Theta^\top + Q \\ K_t &= P_t^{t-1} M_t^\top (M_t P_t^{t-1} M_t^\top + R_t)^{-1} \\ x_t^t &= x_t^{t-1} + K_t (y_t^{res} - M_t x_t^{t-1}) \\ P_t^t &= P_t^{t-1} - K_t M_t P_t^{t-1} \end{aligned} \quad (24)$$

where we take  $x_0^0 = \mu$  and  $P_0^0 = \Sigma$ . Next, we use the backward recursion

$$\begin{aligned} J_{t-1} &= P_{t-1}^{t-1} \Theta^\top (P_t^{t-1})^{-1} \\ x_{t-1}^n &= x_{t-1}^{t-1} + J_{t-1} (x_t^n - \Theta x_{t-1}^{t-1}) \\ P_{t-1}^n &= P_{t-1}^{t-1} + J_{t-1} (P_t^n - P_t^{t-1}) J_{t-1}^\top \\ P_{t-1, t-2}^n &= P_{t-1}^{t-1} J_{t-2}^\top + J_{t-1} (P_{t, t-1}^n - \Theta P_{t-1}^{t-1}) J_{t-2}^\top \end{aligned} \quad (25)$$

with initialization  $P_{n, n-1}^n = (I - K_n M_n) \Theta P_{n-1}^{n-1}$ . One forward and one backward recursion completes the E-step of the EM algorithm. To derive the M-step first realize that the

<sup>8</sup> $m$  indicates the dimension of the measurements of the time series.

<sup>9</sup> $y_t^{res}$  is defined as the residual error which cannot be explained by the neural network model if a blood glucose measurement occurs (i.e.  $y_t^{res} = y_m^{bg} - y_t^{RNN}$ ). This equation is always well defined since the Kalman corrector step ( $x_t^t, P_t^t$ ) is not used in the case of missing observations if one defines  $y_t^{res} = 0$  and  $M_t = 0$  or  $R_t = \infty$ , respectively.

conditional expectations in (23) yield the following equation:

$$\begin{aligned}
G &= -\frac{1}{2} \log |\Sigma| - \frac{1}{2} \text{tr} \{ \Sigma^{-1} (P_0^n + (x_0^n - \mu)(x_0^n - \mu)^\top) \} \\
&\quad - \frac{n}{2} \log |Q| - \frac{1}{2} \text{tr} \{ Q^{-1} (C - B\Theta^\top - \Theta B^\top - \Theta A\Theta^\top) \} \\
&\quad - \frac{n}{2} \log |R_t| - \frac{1}{2} \text{tr} \{ R_t^{-1} \sum_{t=1}^n [(y_t^{res} - M_t x_t)(y_t^{res} - M_t x_t)^\top + M_t P_t^n M_t^\top] \}
\end{aligned} \tag{26}$$

where  $\text{tr}\{\cdot\}$  denotes the trace,  $A = \sum_{t=1}^n (P_{t-1}^n + x_{t-1}^n x_{t-1}^{n\top})$ ,  $B = \sum_{t=1}^n (P_{t,t-1}^n + x_t^n x_{t-1}^{n\top})$  and  $C = \sum_{t=1}^n (P_t^n + x_t^n x_t^{n\top})$ .

$\Theta(r+1) = BA^{-1}$  and  $Q(r+1) = n^{-1}(C - BA^{-1}B^\top)$  maximize the log-likelihood equation (26).  $\mu(r+1)$  is set to  $x_0^n$  and  $\Sigma$  may be fixed at some reasonable baseline level. The derivation of these equations can be found in [28].

The E- (forward and backward Kalman filter equations) and M-steps are alternated repeatedly until convergence.

## REFERENCES

- [1] Diabetes Care, "Diabetes statistics," *NIH Publication No. 96-3926*, 1997.
- [2] Schiffrin A., Mihic M., Leibel B. S., and Albisser M., "Computer assisted insulin dosage adjustment," *Diabetes Care*, vol. 8, 1985.
- [3] Harvey F. E. and 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.
- [4] Chao S. C. and Albisser A. M., "The diabetes simulator," in *Decision support for patient management*. 1989, British medical informatics society.
- [5] Deutsch T., Carson E. R., and Harvey F. E. et al., "Computer-assisted diabetic management: a complex approach," *Comput. methods programs biomed.*, vol. 32, 1990.
- [6] Berger M. P. and Rodbard D., "Computer simulation of plasma insulin and glucose dynamics after subcutaneous insulin injection," *Diabetes Care*, vol. 12, 1989.
- [7] Salzsieder E., Albrecht G., Fischer U., Rutscher A., and Thierbach U., "Computer-aided systems in the management of type I diabetes: the application of a model-based strategy," *Comput. methods programs biomed.*, vol. 32, 1990.
- [8] Leaning M. S. and Boroujerdi M. A., "A system for compartmental modelling and simulation," *Comput. methods programs biomed.*, vol. 35, 1991.
- [9] Schneider J., Piwernetz K., Engelbrecht R., and Renner R., "Diacons-a consultation system to assist in the management of diabetes," in *Expert systems and decision support in medicine*, Reinhoff O. et al., Ed. 1988, Springer.
- [10] Berger M. P., Gelfand R. A., and 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] Lehmann E. D., Deutsch T., Roudsari A. V., Carson J. J., Benn J. J., and Sonksen P. H., "A metabolic prototype to aid in the management of insulin treated diabetic patients," *Diabetes nutr. metab.*, vol. 4:1, 1991.

- [12] Prank K., Jürgens C., von der Mühlen A. and 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.
- [13] Andreassen S., Benn J., Hovorka R., Olesen K., and 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, 1994, Elsevier Sc. Publ. Ireland.
- [14] Hejlesen O. K., Andreassen S., Hovorka R., and 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.
- [15] Detschew W., “Theoretische und experimentelle Beiträge zur Messwertverarbeitung und Reglerentwicklung für die künstliche Betazelle”, TH Ilmenau, 1990.
- [16] Oranien S., “Das Beratungssystem Diabetex – Ein Beitrag zur Verbesserung der ambulanten Betreuung von Diabetikern mit kontinuierlicher subkutaner Insulininfusion”, TH Ilmenau, 1989.
- [17] Tresp V., Moody J., and DeLong W.-R., “Neural modeling of physiological processes,” *Computational Learning Theory and Natural Learning Systems*, vol. 2, 1994, S. Hanson *et al.*, eds., MIT Press.
- [18] Cobelli C. and 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.
- [19] Zander E., “Einsatz der Muskelarbeit in der Diabetesbehandlung in Diabetes Mellitus”, Springer, 1989.
- [20] Williams R. J. and Zipser D., “A learning algorithm for continually running fully recurrent neural networks,” *Neural Computation*, vol. 1, pp. 270–280, 1989.
- [21] Berger M. and Jörgens V., “Praxis der Insulintherapie”, Springer, 1990.
- [22] Ljung L., “System Identification, Theory for the User”, Prentice-Hall, 1987.
- [23] LeCun Y., Boser B., Denker J. S., Henderson D., Howard R. E., Hubbard W., and Jackel L. D., “Handwritten digit recognition with a back propagation network,” *Advances in Neural Information Processing Systems*, vol. 2, pp. 396–404, 1990, Morgan Kaufmann, San Mateo.
- [24] Gill P. E., Murray W., Saunders M. A., and Wright M., “Practical Optimization”, Academic Press, 1981.
- [25] Briegel T. and Tresp V., “A neuro-fuzzy approach to feedforward modeling of nonlinear time series,” *Proceedings of the 5.th International Conference Fuzzy-Neuro Systems '98*, 1998, Munich.
- [26] Tresp V. and 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, Eds. M. Jordan, M. Kearns, and S. Solla, MIT Press, 1998.
- [27] Jazwinski A. H., “Stochastic Processes and Filtering Theory”, Academic Press, 1970.
- [28] Shumway R. H. and Stoffer D. S., “Time series smoothing and forecasting using the em algorithm,” *Technical Report No. 27*, 1981, Division of Statistics, UC Davis.