Modeling Clinical Time Series Using Gaussian Process Sequences

Zitao Liu* Lei Wu† Milos Hauskrecht‡

Abstract

Development of accurate models of complex clinical time series data is critical for understanding the disease, its dynamics, and subsequently patient management and clinical decision making. Clinical time series differ from other time series applications mainly in that observations are often missing and made at irregular time intervals. In this work, we propose and test a new probabilistic approach for modeling clinical time series data that is optimized to handle irregularly sampled observations. Our model is defined by a sequence of Gaussian processes (GPs), each restricted to a window of a finite size, where dependencies among two consecutive Gaussian processes are represented using a linear dynamical system. We develop algorithms supporting both model learning and inference. Experiments on real-world clinical time series data show that our model is better for modeling clinical time series and that it outperforms or is close to alternative time series prediction models.

1 Introduction

Development of accurate models of clinical time series is extremely important for disease prediction and patient management. However, modeling of clinical time series comes with a number of challenges [21, 1]. First, the time series for an individual patient may vary in length and may span a number of days depending on the length of patient's hospitalization. Second, the time series observations are obtained at different times which means the time elapsed between the two consecutive observations may vary, see Figure 1(a). Our objective is to build dynamical models and algorithms that are flexible enough to work under these assumptions.

The key component of our approach is the Gaussian process (GP) model [20]. The GP model is a non-linear nonparametric model defining a multivariate Gaussian over collections of real-valued variables, and effectively defines distribution over functions $f(x)$ [20]. The GP model is robust to noise and can be used for predicting a function value f for any value x, given a set of observation-value pairs $\{(x_1, f_1), (x_2, f_2), \cdots, (x_k, f_k)\}$. We use this property of GP to model observations collected (sampled) at irregular times. Assuming x models time, we hope to use GPs to represent distributions of clinical time series values collected at different times.

Application of the GP model to the clinical time domain is not straightforward. First, one needs to define a mean function that is flexible enough for the clinical time series prediction. Second, the mean function in general depends on the time, which raises the question of how to align the different clinical time series data (corresponding to different patients).

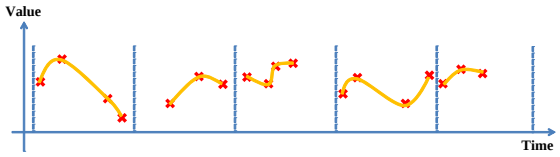
To address the above problems, instead of defining one Gaussian process model over the entire patient time series, we propose to split the process into a sequence of dependent Gaussian processes defined over time-windows of equal size(see Figure 1(b)). Time-points delimiting the windows then define the time origins of Gaussian processes active in respective windows which allows for suitable alignment of different time series for individual patients. The dependencies between the Gaussian processes in two consecutive time windows are modeled using a linear dynamical system (LDS)(see Figure 1(c)). The LDS is relatively simple, but unlike its typical application, we do not use it to model the dynamics of observed values directly, instead we use it to define and control the dynamics of Gaussian process sequences by controlling their parameters. We refer to our model as to the *State-Space Gaussian Process* (SSGP) model.

Our time series model is defined as a full probabilistic model. Hence all inference and learning tasks can be handled using the probabilistic framework. To support these tasks we propose and derive the algorithms: (1) for making future-value predictions, and (2) for learning the model from data. The learning algorithm is an extension of the well-known Expectation-Maximization algorithm [6] used to learn parameters of probabilistic models with hidden variables.

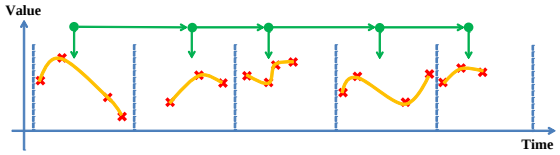
We test the model and the algorithm on the problem of time series prediction for six common blood tests from the complete blood count (CBC) panel. Our results demonstrate that our model leads to a more accurate predictive performance than alternative time series models. In addition, we show that our model is more robust than the alternatives when the number of patients and observations used to train the models is

*University of Pittsburgh. Email: ztliu@cs.pitt.edu
†University of Pittsburgh. Email: wuleibig@gmail.com
‡University of Pittsburgh. Email: milos@cs.pitt.edu

(a) Irregularly sampled time series data



(b) A sequence of dependent GPs over windows



(c) Dependences between GPs are modeled by LDS

Figure 1: Graphical illustration of our state-space Gaussian Process(SSGP)

small.

Our paper is structured as follows. First, in Section 2, we cover the basics of the linear state-space model, Gaussian processes and applications of Gaussian processes to time series modeling. In Section 3, we formulate the problem we want to solve and describe our new Gaussian process model. Section 4 explains the inference and learning details of the model. Section 5 describes the general procedure for applying the model to predict future time series values. Section 6 focuses on regression experiments on the real clinical data, and compares the results to alternative modeling approaches. Finally, Section 7 summarizes the work and outlines possible future extensions.

2 Background

In the following we first review the linear dynamical system (LDS) and two kinds of dynamical models based on Gaussian process.

2.1 Linear Dynamical System The time-invariant Linear Dynamical System (LDS), is a classic and widely used real-valued time series model [3, 25]. An LDS on variables $\mathbf{z}_{1:T}, \mathbf{y}_{1:T}$ is defined in terms of the following two equations:

$$\mathbf{z}_{t+1} = A\mathbf{z}_t + \mathbf{w}_t \tag{2.1}$$

$$\mathbf{y}_t = C\mathbf{z}_t + \mathbf{v}_t, \tag{2.2}$$

where $t \in \{1, \ldots, T\}$ is the discrete time index; $\mathbf{z}_1$ is the initial state distribution with mean π_1 and covariance V_1, $\mathbf{z}_1 \sim \mathcal{N}(\mathbf{z}_1|\pi_1, V_1)$, $\mathbf{z}_t$ are the hidden states generated by the transition matrix A with independent zero mean noise $\mathbf{w}_t, \mathbf{w}_t \sim \mathcal{N}(\mathbf{w}_t|0, Q)$; and $\mathbf{y}_t$ are the observations generated by the emission matrix C with independent variate noise $\mathbf{v}_t, \mathbf{v}_t \sim \mathcal{N}(\mathbf{v}_t|0, R)$. The LDS is characterized by a state transition probability $p(\mathbf{z}_{t+1}|\mathbf{z}_t)$ where $p(\mathbf{z}_{t+1}|\mathbf{z}_t) = \mathcal{N}(\mathbf{z}_{t+1}|A\mathbf{z}_t, Q)$, and a state to observation probability $p(\mathbf{y}_t|\mathbf{z}_t)$ where $p(\mathbf{y}_t|\mathbf{z}_t) = \mathcal{N}(\mathbf{y}_t|C\mathbf{z}_t, R)$. The graphical illustration of LDS is shown in Figure 2.

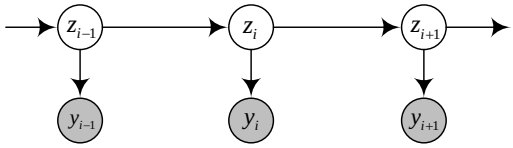


Figure 2: Graphical representation of a linear dynamical system. Shaded nodes $\mathbf{y}_i$ denote observations and unshaded nodes $\mathbf{z}_i$ correspond to hidden states.

The complete set of LDS parameters is $\Theta_{LDS} = \{A, C, Q, R, \pi_1, V_1\}$. The parameters can be estimated (learned) from data, for example, using the Expectation-Maximization (EM) algorithm [7].

The advantage of the linear dynamical model is its simplicity. A disadvantage is its linearity which may prevent one from modeling more complex time series data, and the fact that the model is a discrete time model with observations and predictions restricted to fixed time intervals. For example, discretization of irregularly sampled time series may introduce unnecessary inaccuracy and hence lower the model's performance.

2.2 Gaussian Processes and Dynamical System **Gaussian process (GP)** is a nonparametric nonlinear Bayesian model popular in statistical machine learning. The GP is an extension of multivariate Gaussians to infinite-sized collections of real-valued variables. We can think of this extension as a distribution over random functions [20]. A GP is specified by its mean function $m(\mathbf{x}) = \mathbb{E}[f(\mathbf{x})]$ and its covariance function $K(\mathbf{x}, \mathbf{x}') = \mathbb{E}[(f(\mathbf{x}) - m(\mathbf{x}))(f(\mathbf{x}') - m(\mathbf{x}'))]$, where $f(\mathbf{x})$ is the real process. Since GP can be viewed as a Gaussian distribution over functions, it can be used to estimate the values of function f at an arbitrary position x_*. This application is referred to as *Gaussian Process Regression*

[20]. The basic GP regression equations are

$$\bar{f}_* = K(x_*, \mathbf{x}) \left[K(\mathbf{x}, \mathbf{x}) + \sigma^2 I\right]^{-1} \mathbf{y} \quad (2.3)$$

$$Cov(f_*) = K(x_*, x_*) - K(x_*, \mathbf{x}) \cdot \left[K(\mathbf{x}, \mathbf{x}) + \sigma^2 I\right]^{-1} K(x_*, \mathbf{x}), \quad (2.4)$$

where I is the identity matrix, $\mathbf{x}$ is the input vector and $\mathbf{y}$ is the output or target, $\bar{f}_*$ is the posterior function mean and $Cov(f_*)$ is the posterior covariance. With the right choice of the covariance function, the associated prediction uncertainty increases in regions away from past observations, while it shrinks when it is close to observed data.

The Gaussian process methodology can be applied to modeling of dynamical systems by either: (1) modeling non-linearities in state transitions and observations for discrete-time systems, or, (2) modeling the time series of observations as a function of time.

Discrete-time Gaussian process dynamical system (DTGPDS). Let $\mathbf{z}_t$, and $\mathbf{y}_t$ respectively define a hidden state and an observation at time t, similarly to the LDS system (see Figure 2). Then the Gaussian process discrete-time dynamical system is defined as:

$$\mathbf{z}_{t+1} = r(\mathbf{z}_t) + \mathbf{w}_t \quad (2.5)$$

$$\mathbf{y}_t = u(\mathbf{z}_t) + \mathbf{v}_t, \quad (2.6)$$

where the equations mirror the LDS equations in 2.2 and 2.2. The transition function r and the observation function u represent stochastic transitions and observations, and are represented with the help of Gaussian processes. $\mathbf{w}_t$ and $\mathbf{v}_t$ are the same as in 2.2 and 2.2. Briefly, the LDS assumes linear dependencies among latent states and observations, while the the Gaussian-process-based model replaces the linear dependencies with more general nonlinear functions r and u. Please note that if $\mathbf{z}_t$ states are observed then the model collapses to an autoregressive model which is represented by a single GP.

A number of approaches for inference and optimization of DTGPDS model and its clones have been proposed in the literature. Briefly, [5] proposed GP-Assumed Density Filter(GP-ADF), an inference algorithm that approximates the predictive distribution using the moment matching. The GP Dynamical Model(GPDM) [26] and GP-Bayes Filter [14] develop and rely on the the MAP approximation of the distribution of the latent state. Finally, [23] introduced the GPIL algorithm for inference and learning in the nonlinear dynamical system based on the Expectation-Maximization (EM) framework.

The advantage of the DTGPDS is that it lets us represent more general transition and observation models than LDS. However, the model is still the discrete-time model and the discretization may introduce inaccuracies especially when it is applied to data that are irregularly sampled in time.

Continuous-time Gaussian process dynamical system (CTGPDS). An alternative approach is to model observation dynamics as a function of time $q(t)$ and use the Gaussian process to model the distribution of these functions [2, 19]. In such a case, the Gaussian process defines a continuous-time process, as opposed to a discrete time model, which appears to be promising especially when the problem is hard to discretize in time. This is particularly useful for our problem in which observations are spaced irregularly in time. Unfortunately, this approach also comes with limitations; the most serious one is that the mean function of the GP is a function of time. This makes the time series modeling approach dependent on the time origin and the length of the different time series, which is hard to assure for real world clinical data. More specifically, patients may be encountered at different times (it is not clear where the time origin should be) and the lengths of their hospital stays may vary. Hence the only way to align them properly is to make the mean function of the GP time invariant (equal to a constant value), which would significantly limit our ability to capture various time series variations.

In this paper, we address the shortcomings of the CTGPDS approach by splitting the process into a sequence of local Gaussian processes and by using the discrete-time LDS model to capture the dependences between these local GPs. This is unlike [17] where local GPs are independent to each other. The local GPs' dependences naturally account for the transitions and changes of mean functions and the irregular samples are handled by local GPs.

3 State-Space Gaussian Process model

In this section we develop a new probabilistic model, the state-space Gaussian process (SSGP) model, for representing clinical time series. Our model is able to support time series prediction with irregularly sampled observations that are characteristic of clinical time series, and lets one align more flexibly multiple clinical time series.

3.1 Time series prediction We define the time series prediction/regression function for clinical time series as: $g : \mathbf{Y}_{\text{obs}} \times t \to \hat{\mathbf{y}}$, where $\mathbf{Y}_{\text{obs}}$ is a sequence of past observation-time pairs $\mathbf{Y}_{\text{obs}} = (\mathbf{y}_i, t_i)_{i=1}^n$, such that, $0 < t_i < t_{i+1}$, $\mathbf{y}_i$ is a p-dimensional observation vector made at time (t_i), and n is the number of past observations; and $t > t_n$ is the time at which we would

like to predict the observation $\hat{\mathbf{y}}$.

Typically, in a discrete-time dynamical system the prediction function assumes values (observations) are regularly sampled, that is, the time difference in between two consecutive time points is a constant $t_{i+1} - t_i = L$, and that predictions are made at some future times consistent with the sampling constant L. In this work we assume observations can be spaced irregularly in time. For example, in the clinical domain, observations that correspond to lab test values for a patient during his or her hospital period are often recorded irregularly due to different patients' health conditions.

3.2 The Model Our model consists of two hierarchically related dynamical processes: the Gaussian process and the linear dynamical process. The Gaussian process is a continuous-time model restricted to a time window of a finite duration and is used to represent time series and its changes for shorter time spans. Longer-term process changes are modeled and controlled by the linear dynamical system. We start the description of our model, by first describing its Gaussian process component aimed to model observations that are made at irregular times.

We consider the Gaussian process $q(\mathbf{t})$ with the mean function formed by a combination of a fixed set of basis functions with coefficients, $\boldsymbol{\beta}$:

$$q(\mathbf{t}) = f(\mathbf{t}) + \mathbf{h}(\mathbf{t})^T\boldsymbol{\beta},\ f(\mathbf{t}) \sim \mathcal{GP}_f(0, K(\mathbf{t}, \mathbf{t}')) \tag{3.7}$$

In this definition, $f(\mathbf{t})$ is a zero mean GP, $\mathbf{h}(\mathbf{t})$ denotes a set of fixed basis functions, for example, $\mathbf{h}(\mathbf{t}) = (1, t, t^2, \ldots)$, and $\boldsymbol{\beta}$ is a Gaussian prior, $\boldsymbol{\beta} \sim \mathcal{N}(\mathbf{b}, I)$. Following [18], $q(\mathbf{t})$ is another GP process, defined by:

$$q(\mathbf{t}) \sim \mathcal{GP}_q(\mathbf{h}(\mathbf{t})^{\mathbf{T}}\mathbf{b}, K(\mathbf{t}, \mathbf{t}') + \mathbf{h}(\mathbf{t})^T\mathbf{h}(\mathbf{t}')) \tag{3.8}$$

The above Gaussian process (with parameters $\boldsymbol{\beta}$) may not be flexible enough for the entire time series. In addition, the mean function of the process depends on time (and the time origin $t = 0$) which begs the question of how to align time series obtained for multiple patients. To achieve more flexibility, we assume the above Gaussian process represents the time series only in the time window of a limited span, and that the dynamics of the entire time series is captured by a linear state-space model representing the transitions of $\boldsymbol{\beta}$ for two consecutive time windows. This allows us to represent the entire time series variations in a more flexible manner, with the Gaussian process being reset to $t = 0$ at the beginning of each window.

More specifically, we divide entire irregular time series data into m windows $\mathbf{W} = \{w_1, w_2, \ldots, w_m\}$. For each window w_i, we define s_i as the size of w_i, which is the number of observations in window w_i. We use $\mathbf{y}_{i,:}$ to represent all the observations that fall into window w_i and $\mathbf{y}_{i,j}$ is the jth observation in w_i. Then, instead of using a single GP to capture the variation in the entire time series, we divide the responsibility into different windows, and each window w_i is associated with a new GP. We use $\boldsymbol{\beta}_i$ to denote the Gaussian prior coefficients in GP_i's mean function, $\boldsymbol{\beta}_i \sim \mathcal{N}(\mathbf{b}_i, I)$, which means every component in $\boldsymbol{\beta}_i$ follows the multivariate Gaussian distribution with mean vector $\mathbf{b}_i$ and covariance matrix I. Each $\boldsymbol{\beta}_i$ describes the combination weights for the mean function of GP_i. To relate Gaussian processes associated with pairs of consecutive windows we define a hidden state linear process $\mathbf{z} \equiv \{\mathbf{z}_i\}$ that captures the dependencies among Gaussian processes in terms of the transition of mean functions' combination weights $\boldsymbol{\beta} \equiv \{\boldsymbol{\beta}_i\}$. This leads to the linear process: $\mathbf{z}_{t+1} = A\mathbf{z}_t + \mathbf{w}_t$ and $\boldsymbol{\beta}_t = C\mathbf{z}_t + \mathbf{v}_t$, where $\mathbf{w}_t$ and $\mathbf{v}_t$ are zero-mean normally distributed random variables with covariance matrices Q and R respectively. This captures the relations between different $\boldsymbol{\beta}_i$s. Since the entire time series data is generated from the same stochastic process, we assume different windows' GPs share the same covariance function, which is parametrized by Θ.

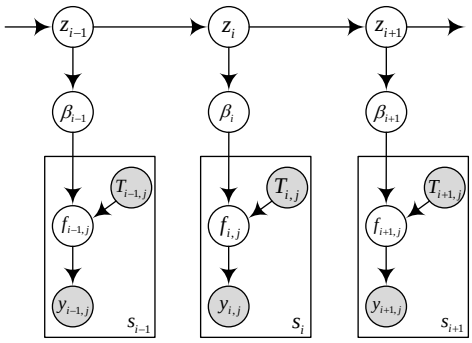


Figure 3: Graphical representation of the state-space Gaussian process model. Shaded nodes $\mathbf{y}_{i,j}$ denote (irregular) observations and shaded nodes $T_{i,j}$ denote times associated with each observation. Each rectangle (plate) corresponds to a window, which is associated with its own local GP. s_i is the number of observations in each window. $f_{i,j}$ is Gaussian field.

The graphical representation of our state-space Gaussian process model is shown in Figure 3. The prior of the initial state $\mathbf{z}_1$ is a Gaussian distribution with mean π_1 and covariance V_1. The mean function for the GP_i is parametrized by $\boldsymbol{\beta}_i$. The entire parameter space can be summarized as $\Omega := \{\Theta, \{\boldsymbol{\beta}_i\}, A, C, R, Q, \pi_1, V_1\}$. When the horizontal ar-

rows in Figure 3 are removed, breaking the time dynamics, the graphical model reduces to a set of independent Gaussian process regression models. With time dynamics, the coefficients of the mean function at slice i has smoothly evolved from those at slice i-1.

3.3 Choice of the Covariance Function

Mean Reverting Property. To define the covariance function of the Gaussian process we resort to the mean reverting process. The mean-reverting, or Ornstein-Uhlenbeck process, is a stationary Gaussian process that obeys the Markov property [9]. It assumes that over time, the process tends to drift towards its long-term mean. Mean reversion is an important property in clinical time series prediction; it forces the process to approach the long term mean, but at the same time permits temporary deviations from the mean corresponding to episodic events or complications. To incorporate the mean reverting phenomenon into the Gaussian process we rely on the 'mean-reverting' covariance function $K_1 = \sigma_1 \exp(\theta_1 |\mathbf{t} - \mathbf{t}'|)$.

Periodicity. The time series often reflect periodic information. The periodic form can capture the fluctuation within the short period of time. In addition, a periodic function can keep the variation of different values within a reasonable range. In context of the Gaussian process the periodicity can be captured by a special covariance function: $K_2 = \sigma_2 \exp(\theta_2 \sin^2 \left[\frac{\omega}{2\pi}(\mathbf{t} - \mathbf{t}')\right])$.

To incorporate the two properties, we choose $K = K_1 + K_2$ as our $\mathcal{GP}$'s covariance function. However, we would like to note that at this point this choice of a covariance function is a heuristic, and more advanced covariance functions may be designed.

4 Model Learning

In the following section, we describe an algorithm developed for learning the model from the data. The data consists of time series recorded for multiple patients. Our algorithm is based on the well-known Expectation-Maximization (EM) algorithm that iterates two steps: the expectation step that infers values of all hidden variables and/or missing values, and the maximization step that uses the inferred values to calculate the new parameters of the model.

4.1 Inference (E-step) Since both the Markov chain defined by the linear dynamical model $\{\mathbf{z}_i\}$ and the mean coefficient $\{\boldsymbol{\beta}_i\}$ are unobserved, we cannot learn $\{GP_i\}$ directly; instead, we apply the EM algorithm to learn linear hidden transition of GPs' mean coefficients and its covariance hyper parameter together.

The E-step infers a posterior distribution of latent states $\mathbf{z}, \boldsymbol{\beta}$ given the observation sequences $\mathbf{Y}_{\text{obs}}$, $p(\mathbf{z}, \boldsymbol{\beta} | \mathbf{Y}_{\text{obs}}, \Omega)$. In the following, we omit the explicit conditioning on Ω and use $\mathbf{Y}$ to replace $\mathbf{Y}_{\text{obs}}$ for notational brevity. Due to the conditional independence encoded in SSGP, the joint distribution of the data is given by:

$$\begin{aligned} p(D) = p(\mathbf{z}, \boldsymbol{\beta}, \mathbf{Y}) &= p(\mathbf{z}_1) \prod_{i=2}^{m} p(\mathbf{z}_i | \mathbf{z}_{i-1}) \\ &\cdot \prod_{i=1}^{m} (\boldsymbol{\beta}_i | \mathbf{z}_i) \prod_{i=1}^{m} \prod_{j=1}^{s_i} p(\mathbf{y}_{i,j} | \boldsymbol{\beta}_i) \end{aligned} \tag{4.9}$$

This E-step requires computing the expected log likelihood $\mathcal{Q} = \mathbb{E}_{\boldsymbol{\beta},\mathbf{z}}[\log p(\boldsymbol{\beta}, \mathbf{z}, \mathbf{Y} | \Omega)]$, which is depends on $\mathbb{E}[\mathbf{z}_i|\mathbf{Y}]$, $\mathbb{E}[\mathbf{z}_i \mathbf{z}_i^{'}|\mathbf{Y}]$ and $\mathbb{E}[\mathbf{z}_i \mathbf{z}_{i-1}^{'}|\mathbf{Y}]$. Let $\hat{\mathbf{z}}_{i|T} \equiv \mathbb{E}[\mathbf{z}_i|\mathbf{Y}]$, $M_{i|T} \equiv \mathbb{E}[\mathbf{z}_i \mathbf{z}_i^{'}|\mathbf{Y}]$, $M_{i,i-1|T} \equiv \mathbb{E}[\mathbf{z}_i \mathbf{z}_{i-1}^{'}|\mathbf{Y}]$, $P_{i|T} = \mathbb{VAR}[\mathbf{z}_i|\mathbf{Y}]$ and $P_{i,i-1|T} = \mathbb{VAR}[\mathbf{z}_i \mathbf{z}_{i-1}^{'}|\mathbf{Y}]$. T is the length of time series. Note that the hidden state estimate $\hat{\mathbf{z}}_{i|T}$ depends on both past and future observations. To compute $\hat{\mathbf{z}}_{i|T}$ and $M_{i|T}$, we follow [22] performing a backward algorithm to compute these hidden state estimations given on all(previous, current, and future) observations. See details in Algorithm 1.

Algorithm 1 EM: E-step

Backward algorithm for SSGP:
// Compute $\hat{\mathbf{z}}_{i|T}$, $M_{i|T}$ and $M_{i,i-1|T}$
// By definition, $M_{i|T} = P_{i|T} + \hat{\mathbf{z}}_{i|T} \hat{\mathbf{z}}_{i|T}^{'}$
// By definition, $M_{i,i-1|T} = P_{i,i-1|T} + \hat{\mathbf{z}}_{i|T} \hat{\mathbf{z}}_{i-1|T}^{'}$
// Initialization: $P_{T,T-1|T} = (I - K_T C) A P_{T-1|T-1}$
$J_{i-1} = P_{i-1|i-1} A^{'} (P_{i|i-1})^{-1}$
$\hat{\mathbf{z}}_{i-1|T} = \hat{\mathbf{z}}_{i-1|i-1} + J_{i-1}(\hat{\mathbf{z}}_{i|T} - A \hat{\mathbf{z}}_{i-1|i-1})$
$P_{i-1|T} = P_{i-1|i-1} + J_{i-1}(P_{i|T} - P_{i|i-1}) J_{i-1}^{'}$
$P_{i-1,i-2|T} = P_{i-1|i-1} J_{i-2}^{'} + J_{i-1}(P_{i,i-1|T} - A P_{i-1|i-1}) J_{i-2}^{'}$
// where $P_{i-1|i-1}, P_{i|i-1}, \hat{\mathbf{z}}_{i|i-1}, \hat{\mathbf{z}}_{i|i}$ and K_i are computed by $Kalman\_Filter$. See Appendix A.1.

4.2 Learning(M-step) In the following, we derive the M-step for gradient based optimization of the parameters Ω. In the M-step, we try to find Ω that maximizes the likelihood lower bound $\mathcal{Q} = \mathbb{E}_{\boldsymbol{\beta},\mathbf{z}}[\log p(\boldsymbol{\beta}, \mathbf{z}, \mathbf{Y} | \Omega)]$. In the following, we omit the explicit conditioning on Ω for notational brevity. The factorization properties of SSGP yield the decomposition $\mathcal{Q}$ into

$$
\begin{aligned}
\mathcal{Q} =& \mathbb{E}_{\boldsymbol{\beta},\mathbf{z}}[\log p(\boldsymbol{\beta},\mathbf{z},\mathbf{Y})] = \mathbb{E}_{\boldsymbol{\beta},\mathbf{z}}[\log p(\mathbf{z}_1)] \\
&+\mathbb{E}_{\boldsymbol{\beta},\mathbf{z}}\Big[\sum_{i=2}^{m}\log p(\mathbf{z}_i|\mathbf{z}_{i-1})\Big] \\
&+\mathbb{E}_{\boldsymbol{\beta},\mathbf{z}}\Big[\sum_{i=1}^{m}\log p(\boldsymbol{\beta}_i|\mathbf{z}_i)\Big] \\
&+\mathbb{E}_{\boldsymbol{\beta},\mathbf{z}}\Big[\sum_{i=1}^{m}\sum_{j=1}^{s_i}\log p(\mathbf{y}_{i,j}|\boldsymbol{\beta}_i)\Big] \qquad (4.10)
\end{aligned}
$$

As we can see from eq.(4.10), the shares parameters Θ of the covariance function for all $\{GP_i\}$ only appear in the last term of $\mathcal{Q}$, which is $\mathbb{E}_{\boldsymbol{\beta},\mathbf{z}}\Big[\sum_{i=1}^{m}\sum_{j=1}^{s_i}\log p(\mathbf{y}_{i,j}|\boldsymbol{\beta}_i)\Big]$. We can easily get the derivative(see eq. 4.11) and use any gradient based optimizer to estimate them.

$$
\begin{aligned}
\frac{\partial \log p(\mathbf{Y}|\Theta)}{\partial\Theta} =& -\frac{1}{2}\mathrm{Tr}\left[K^{-1}\frac{\partial K}{\partial\Theta}\right] \\
&+\frac{1}{2}\mathbf{Y}^T K^{-1}\frac{\partial K}{\partial\Theta}K^{-1}\mathbf{Y} \qquad (4.11)
\end{aligned}
$$

For each of the rest of parameters $\{\{\boldsymbol{\beta}_i\}, A, C, R, Q, \pi_1, V_1\}$, we re-estimate them by taking the corresponding partial derivative of the expected log likelihood, setting to zero, and solving. These result in Algorithm 2.

4.3 Summary of the Learning Algorithm The parameter estimation method for the SSGP is summarized by Algorithm 3. Let us define $\hat{\mathbf{z}}_i \equiv \{\hat{\mathbf{z}}_{i|T}\}_1^T$, $\mathbf{M}_i \equiv \{M_{i|T}\}_1^T$ and $\mathbf{M}_{i,i-1} \equiv \{M_{i,i-1|T}\}_1^T$. The function *SSGPsmoother* implements the E-step and the *maximize* routine implements the M-step.

Algorithm 2 EM: M-step

//Define H_i matrix collects the $\mathbf{h}(\mathbf{x})$ vectors for all the observations $\mathbf{y}_{i,:}$ in window w_i.
for $i = 1$ **to** m **do**
$E_i = (K_{\mathbf{y}_{i,:}} + H_i^{'} H_i)^{-1}$
$\boldsymbol{\beta}_i = (R^{-1} + H_i E_i H_i^{'})^{-1}(R^{-1} C\hat{\mathbf{z}}_{i|T} + H_i E_i \mathbf{y}_{i,:})$
end for
$\pi_1 = \hat{\mathbf{z}}_{1|T}$
$V_1 = M_{1|T} - \hat{\mathbf{z}}_{1|T}\hat{\mathbf{z}}_{1|T}^{'}$
$A = (\sum_{i=2}^{m} M_{i,i-1|T})(\sum_{i=2}^{m} M_{i-1|T})^{-1}$
$Q = \frac{1}{m-1}(\sum_{i=2}^{m} M_{i|T} - A\sum_{i=2}^{m} M_{i-1,i|T})$
$R = \frac{1}{m}\sum_{i=1}^{m}(\boldsymbol{\beta}_i\boldsymbol{\beta}_i^{'} - C\hat{\mathbf{z}}_{i|T}\boldsymbol{\beta}_i^{'})$
$C = (\sum_{i=2}^{m}\boldsymbol{\beta}_i\hat{\mathbf{z}}_{i|T}^{'})(\sum_{i=1}^{m} M_{i|T})^{-1}$

Algorithm 3 Parameter Estimation in SSGP

Get Θ by any gradient optimizer based on eq.(4.11).
init $\Omega\backslash\Theta$
repeat
E-step: Section 4.1, Algorithm 1
$\hat{\mathbf{z}}_i$, $\mathbf{M}_i$ and $\mathbf{M}_{i,i-1} \leftarrow$ *SSGPsmoother* $(\mathbf{Y}, \Omega\backslash\Theta)$
M-step: Section 4.2, Algorithm 2
$\Omega\backslash\Theta \leftarrow$ *maximize* $\mathcal{Q}(\Omega, \hat{\mathbf{z}}_i, \mathbf{M}_i, \mathbf{M}_{i,i-1})$ wrt $\Omega\backslash\Theta$
until Convergence
return $\Omega = \{\Theta, \{\boldsymbol{\beta}_i\}, A, C, R, Q, \pi_1, V_1\}$

5 Prediction

Once the state-space Gaussian process model is learned from the training data we would like to use it to support time series prediction on future time series. Given initial observations $\mathbf{Y}_{\text{obs}}$ and an arbitrary time index t, our objective is to predict the future value $\hat{\mathbf{y}}$ at time t.

To support the prediction inference, we need the following steps:

Step 1. Split $\mathbf{Y}_{\text{obs}}$ and t into windows.

Step 2. For windows that do not contain t, extract the last values in those windows as $\boldsymbol{\beta}$s and feed them into Kalman Filter algorithms(See $Kalman\_Filter$ in Appendix A.1) to infer the most recent hidden state $\mathbf{z}_k$ where k is the index of the last window that does not contain t.

Step 3. Get $\boldsymbol{\beta}_{k+1} = CA\mathbf{z}_k$ from $\mathbf{z}_{k+1} = A\mathbf{z}_k$ and $\boldsymbol{\beta}_{k+1} = C\mathbf{z}_{k+1}$.

Step 4. If t is in window $k+1$ use observations $(\mathbf{y}_{k+1}, t_{k+1})$ in window $k+1$ and $\boldsymbol{\beta}_{k+1}$ to make the prediction, where $\hat{\mathbf{y}} = \boldsymbol{\beta}_{k+1} + K(t, t_{k+1})K^{-1}(t_{k+1}, t_{k+1})(\mathbf{y}_{k+1} - \boldsymbol{\beta}_{k+1})$; otherwise find out the window index i where t belongs to. The prediction at t is $\hat{\mathbf{y}} = CA^{i-k}\mathbf{z}_k$.

The prediction algorithm can be summarized in Algorithm 4.

6 Experiments and Results

We have tested our approach on time series data obtained from electronic health records of approximately 4,500 post-surgical cardiac patients stored in PCP database [12, 10, 24]. To test the performance of our prediction model, we randomly selected 1000 patients with the *Complete Blood Count*(CBC) panel test [1] whose hospitalization is longer than 10 days. We selected six tests from the CBC panel to learn the time series models, and applied them to time series prediction tasks. The six tests used in the experiment are listed in Table 1.

[1] CBC test is used as a broad screening test to check for such disorders as anemia, infection, and many other diseases.

Algorithm 4 Prediction in SSGP

// Split $\mathbf{Y}_{\text{obs}}$ and t into windows.
// Find all k windows that do not contain t and the last observations $\mathbf{Y}_{\text{last}}$ in these k windows.
// Compute z_k by $Kalman\_Filter$ algorithm.(See Appendix A.1)
$\mathbf{z}_k = Kalman\_Filter(\mathbf{Y}_{\text{last}}, A, C, R, Q, \pi_1, V_1)$
if t is in window $k+1$ **then**
  $\boldsymbol{\beta}_{k+1} = C\mathbf{z}_{k+1} = CA\mathbf{z}_k$
  // Observations in window $k+1$ are $(\mathbf{y}_{k+1}, t_{k+1})$
  $\hat{\mathbf{y}} = \boldsymbol{\beta}_{k+1} + K(t, t_{k+1})K^{-1}(t_{k+1}, t_{k+1})(\mathbf{y}_{k+1} - \boldsymbol{\beta}_{k+1})$
else
  $\hat{\mathbf{y}} = CA^{i-k}\mathbf{z}_k$
end if
return $\hat{\mathbf{y}}$

Table 1: Six lab test from the CBC panel.

Name	Explanation
WBC	White blood cell
MCH	Mean corpuscular hemoglobin
MCHC	Mean corpuscular Hgb concentration
MCV	Mean corpuscular volume
PLT	Platelet count
RDW	Red cell distribution width

These time series data are noisy, their signals fluctuate in time, and observations are obtained with varied time-interval period. Figure 4 illustrates such time series for one of the patients. The X-axis is the time index aligned by hour and the Y-axis are normalized values/observations for each test.

To evaluate the performance of our SSGP approach we applied the five-fold cross validation approach to split the examples into the training and testing sets, such that 200 examples formed the test data, and 800 training examples were used to vary the size of the training set from 100 to 800 in increments of 100 examples. We report average results over different folds. Since the CBC panel is ordered once or just a few times a day, we used the default Gaussian process window size of seven days. We compared the SSGP predictions to four other methods: (1) Linear dynamical model(LDS) trained on the entire time series with a fixed time period of three hours. The values at these times were obtained by interpolating the closest observed values [8, 16]. (2) Discrete-time Gaussian process dynamical system (DTGPDS) implemented using GPIL algorithm [23]. We used the same time period of three hours and the same interpolation approach to estimate observations as in method (1). The Gaussian kernel was used to model the covariance functions for both the transition and observation models. (3) Continuous-time Gaussian process dynamical system (CTGPDS). The covariance function $K(\mathbf{t}, \mathbf{t}') = \sigma_1 \exp(\theta_1 |\mathbf{t} - \mathbf{t}'|) + \sigma_2 \exp(\theta_2 \sin^2\left[\frac{\omega}{2\pi}(\mathbf{t} - \mathbf{t}')\right])$; (4) Window-based linear dynamical system (WLDS). This model is different from the LDS model (model 1). It splits the time series first into windows the same way as SSGP and, after that, it trains an LDS using last observations in each window.

We evaluated and compared the performances of the different methods by calculating the Root Mean Square Error(RMSE) on the test set predictions. More specifically, the RMSE is defined as follows:

$$RMSE = \left[n^{-1} \sum_{i=1}^{n} |y_i - \hat{y}_i|^2 \right]^{1/2}$$

where y_i is the true value, $\hat{y}_i$ is the predicted value and n is the number of data points. The results of RMSE on the six lab tests from the CBC panel (for the training sets of increasing size) are summarized in Figure 5.

6.1 Discussion The results of our experiments show that our state-space Gaussian process(SSGP) model outperforms all other methods in terms of prediction errors on all six CBC lab tests. One of the advantages of our method is that its prediction error is small even when it is trained on a small number of patients and observations. Specifically, from Figure 5, we find the following results:

First, when comparing *CTGPDS, SSGP* to *LDS, GPIL*, we can see that the continuous methods (CTGPDS, SSGP) outperform the discretized methods (LDS, GPIL). We believe this is because 1) the values from patients' tests are always around a normal range plus some variation. The combination of the mean reverting function and the periodic function captures this phenomenon: the mean reverting function forces the predicted values within a normal range and the periodic function allows the fluctuation and variation flexibility. Clearly, LDS cannot capture these variations by their linear equations. 2) LDS solve the multi-step prediction problem by constructing a single model from past observations and by predicting the future values iteratively. Since they use predictions from the past, they are susceptible to the *error accumulation*: errors generated in the history are propagated into future predictions [4]. *CTGPDS, SSGP* make the multi-step prediction directly and hence suffer less from this problem.

Second, comparing *SSGP* and *CTGPDS*, we can see, *SSGP* performs much better than *CTGPDS*. It shows that a single constant mean is not enough for

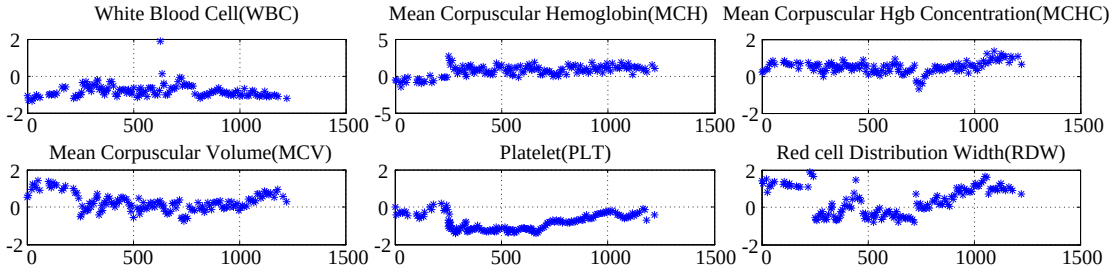


Figure 4: Time series for six tests from the Complete Blood Count(CBC)panle for one of the patients.

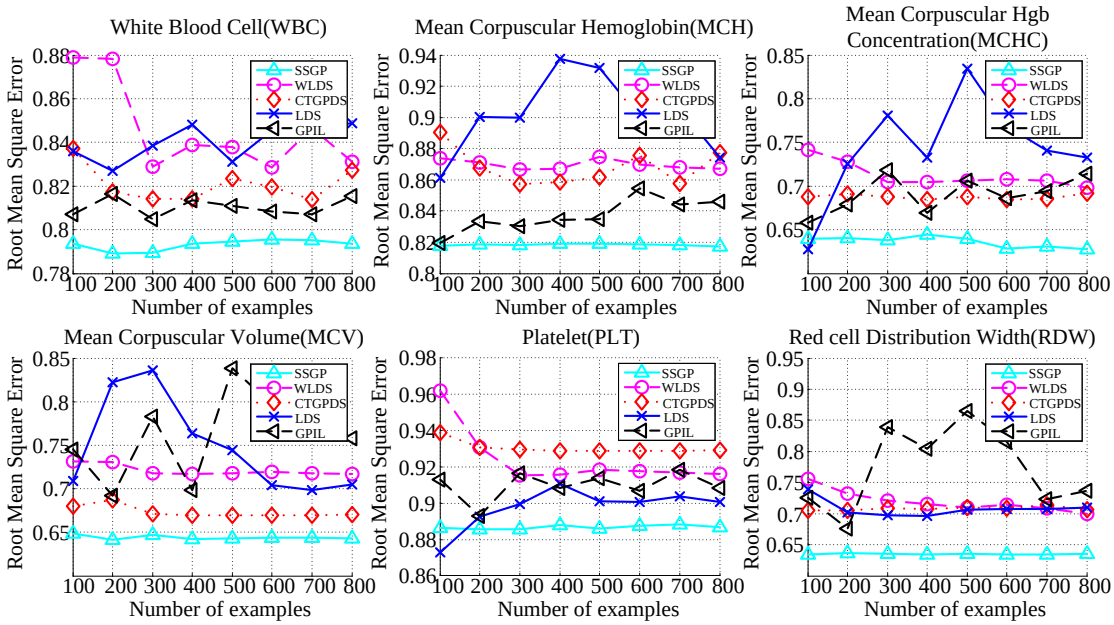


Figure 5: Root Mean Square Error(RMSE) on CBC test samples.

complex time series. The evolution of mean variables in the consecutive windows is modelled by a linear dynamical system, which expresses a stronger descriptive ability. During the prediction phase, its predicted mean is used by the subsequent GP to make more accurate predictions.

Third, compared to other methods, *SSGP* does not require a large number of training examples and it can perform well even with small training data. However, the error rates of other methods decrease by a large amount due to the decrease in the number of examples. In the clinical domain, dataset availability is a big issue. The data is very expensive to obtain. Stable performance on small-size training data is very important in practice.

Fourth, comparing *GPIL* and *LDS*, we can see, *GPIL*'s nonlinear transition function and measurement function boost its performance a lot. It overcomes the linearity problem in *LDS* and has the flexibility to capture the measure noise and model uncertainty to some extent.

7 Conclusion

In this paper, we have presented a state-space Gaussian process system for multi-step prediction. Comparing with the traditional linear state-space systems and modern Gaussian process regression, special features of this novel system are (1) its robustness to irregular sampling; (2) small training sequence data requirement, which is very important in clinical monitoring and alerting systems; (3) its ability to make accurate long-term multi-step predictions. Experimental results on real world clinical data from electronic health records systems demonstrated that the novel prediction model achieves errors that is statistically significantly lower than errors of other state of the art approaches used in

time sequence data prediction. In the future, we plan to study and model dependences among multiple time series, as well as, extensions to switching-state [25] and controlled [11, 15] dynamical systems.

8 Acknowledgement

This research work was supported by grants R01LM010019 and R01GM088224 from the National Institutes of Health. Its content is solely the responsibility of the authors and does not necessarily represent the ofcial views of the NIH.

References

[1] I. Batal, H. Valizadegan, G.F. Cooper, and M. Hauskrecht, *A pattern mining approach for classifying multivariate temporal data*, in IEEE International Conference on Bioinformatics and Biomedicine, 2011, pp. 358 – 365.

[2] J.Q. Candela, A. Girard, J. Larsen, and C.E. Rasmussen, *Propagation of uncertainty in bayesian kernel models-application to multiple-step ahead forecasting*, in ICASSP, vol. 2, IEEE, 2003, pp. II–701.

[3] C.M. Carvalho and H.F. Lopes, *Simulation-based sequential analysis of markov switching stochastic volatility models*, Computational Statistics & Data Analysis, 51 (2007), pp. 4526–4542.

[4] H. Cheng, P.N. Tan, J. Gao, and J. Scripps, *Multistep-ahead time series prediction*, Proceedings of the 10th Pacific-Asia conference on Advances in Knowledge Discovery and Data Mining, (2006), pp. 765–774.

[5] M.P. Deisenroth, M.F. Huber, and U.D. Hanebeck, *Analytic moment-based gaussian process filtering*, in ICML, ACM, 2009, pp. 225–232.

[6] A.P. Dempster, N.M. Laird, and D.B. Rubin, *Maximum likelihood from incomplete data via the em algorithm*, in Journal of the Royal Statistical Society. Series B (Methodological), vol. 39 of 1, 1977, pp. 1–38.

[7] Zoubin Ghahramani and Geoffrey E. Hinton, *Parameter esimation for linear dynamical systems*, tech. report, University of Toronto, 1996.

[8] Mark Gibbs and David J.C. MacKay, *Efficient implementation of gaussian processes*, tech. report, 1997.

[9] Daniel T. Gillespie, *Exact numerical simulation of the ornstein-uhlenbeck process and its integral*, Phys. Rev. E, 54 (1996), pp. 2084–2091.

[10] Milos Hauskrecht, Iyad Batal, Michal Valko, Shyam Visweswaran, Gregory F Cooper, and Gilles Clermont, *Outlier detection for patient monitoring and alerting*, Journal of Biomedical Informatics, 46(1) (2013), pp. 47–55.

[11] M. Hauskrecht and H. Fraser, *Modeling treatment of ischemic heart disease with partially observable markov decision processes.*, in Proceedings of the AMIA Symposium, 1998, pp. 538–542.

[12] M. Hauskrecht, M. Valko, I. Batal, G. Clermont, S. Visweswaran, and G.F. Cooper, *Conditional outlier detection for clinical alerting*, in AMIA Annual Symposium Proceedings, 2010, pp. 286 – 890.

[13] R. E. Kalman, *A new approach to linear filtering and prediction problem*, in Transactions of the ASMEJournal of Basic Engineering, no. 82 in D, 1960, pp. 35–45.

[14] J. Ko and D. Fox, *Learning gp-bayes filters via gaussian process latent variable models*, Autonomous Robots, 30 (2011), pp. 3–23.

[15] Branislav Kveton and Milos Hauskrecht, *Solving factored mdps with exponential-family transition models*, in Proceedings of the 16th International Conference on Automated Planning and Scheduling, 2006, pp. 114–120.

[16] Xin Li and Michael T. Orchard, *New edge-directed interpolation*, in IEEE Transactions on Image Processing, vol. 10, October 2001, pp. 1521–1527.

[17] D. Nguyen-tuong and J. Peters, *Local gaussian process regression for real time online model learning and control*, in NIPS, Citeseer, 2008.

[18] A. O'Hagan and J. F. C. Kingman, *Curve fitting and optimal design for prediction*, Journal of the Royal Statistical Society. Series B (Methodological), 40 (1978), pp. 1–42.

[19] C.E. Rasmussen, M. Kuss, et al., *Gaussian processes in reinforcement learning*, NIPS, 16 (2004).

[20] Carl Edward Rasmussen and Christopher K. I. Williams, *Gaussian Processes for Machine Learning*, MIT Press, 2006.

[21] Ben Y Reis and Kenneth D Mandl, *Time series modeling for syndromic surveillance*, BMC Medical Informatics and Decision Making, 3 (2003).

[22] R. H. Shumway and D. S. Stoffer, *An approach to time series smoothing and forecasting using the em algorithm*, Journal of Time Series Analysis, 3 (1982), pp. 253–264.

[23] R. Turner, M.P. Deisenroth, and C.E. Rasmussen, *State-space inference and learning with gaussian processes*, in AISTATS, vol. 9, 2010, pp. 868–875.

[24] Michal Valko and Milos Hauskrecht, *Feature importance analysis for patient management decisions*, in Proceedings of the 13th International Congress on Medical Informatics, 2010, pp. 861–865.

[25] James M. Rehg Vladimir Pavlovic and John MacCormick, *Learning switching linear models of human motion*, in NIPS, 2000.

[26] J.M. Wang, D.J. Fleet, and A. Hertzmann, *Gaussian process dynamical models for human motion*, Pattern Analysis and Machine Intelligence, IEEE Transactions on, 30 (2008), pp. 283–298.

[27] Greg Welch and Gary Bishop, *An introduction to the kalman filter*, Tech. Report TR 95-041, University of North Carolina at Chapel Hill, 2006.

9 Appendix

9.1 A.1 Kalman Filter Inference [13, 27].

Input for Kalman Filter is $A, C, R, Q, \pi_1, V_1, \{\mathbf{y}_t\}$, which is defined in Section 2.1. $\{\mathbf{z}_t\}$ denotes the hidden state. $\hat{\mathbf{z}}_{t|t-1} = \mathbb{E}[\mathbf{z}_t|\{\mathbf{y}_i\}_1^{t-1}]$ is the *priori* estimation and $P_{t|t-1} = \mathbb{E}[(\mathbf{z}_t - \hat{\mathbf{z}}_{t|t-1})(\mathbf{z}_t - \hat{\mathbf{z}}_{t|t-1})^T]$ is the *priori* estimate error covariance. $\hat{\mathbf{z}}_{t-1|t-1} = \mathbb{E}[\mathbf{z}_{t-1}|\{\mathbf{y}_i\}_1^{t-1}]$ is the *posteriori* estimation and $P_{t-1|t-1} = \mathbb{E}[(\mathbf{z}_{t-1} - \hat{\mathbf{z}}_{t-1|t-1})(\mathbf{z}_{t-1} - \hat{\mathbf{z}}_{t-1|t-1})^T]$ is the *posteriori* estimate error covariance.

Algorithm 5 Kalman_Filter

// Time Update:
$\hat{\mathbf{z}}_{t|t-1} = A\hat{\mathbf{z}}_{t-1|t-1}$
$P_{t|t-1} = AP_{t-1|t-1}A^T + Q$
// Measure Update:
$K_t = P_{t|t-1}C^T(CP_{t|t-1}C^T + R)^{-1}$
$\hat{\mathbf{z}}_{t|t} = \hat{\mathbf{z}}_{t|t-1} + K_t(\mathbf{y}_t - C\hat{\mathbf{z}}_{t|t-1})$