Gaussian Process Kernels for Popular State-Space Time Series Models

Alexander Grigorievskiy
Department of Computer Science
Aalto University
Konemiehentie 2, Espoo, Finland
E-mail: alexander.grigorevskiy@aalto.fi

Juha Karhunen
Department of Computer Science
Aalto University
Konemiehentie 2, Espoo, Finland
E-mail: juha.karhunen@aalto.fi

***Abstract*—In this paper we investigate a link between state-space models and Gaussian Processes (GP) for time series modeling and forecasting. In particular, several widely used state-space models are transformed into continuous time form and corresponding Gaussian Process kernels are derived. Experimental results demonstrate that the derived GP kernels are correct and appropriate for Gaussian Process Regression. An experiment with a real world dataset shows that the modeling is identical with state-space models and with the proposed GP kernels. The considered connection allows the researchers to look at their models from a different angle and facilitate sharing ideas between these two different modeling approaches.**

I. Introduction and Motivation

Time series modeling and prediction is one of oldest topics in statistics. The very first statisticians already dealt with time dependent data. For example, Beveridge wheat price (years 1500 to 1869) or Wolfer's sunspot number (years 1610-1960) [1] are examples of very early time series. Nowadays time series analysis and forecasting is ubiquitous in many fields of science and engineering. Econometricians, physicists, statisticians, biologists, climatologists etc. encounter time dependent data in their daily work.

Since this problem is very old and very wide-spread, different fields of science developed their own sets of methods for analysis and forecasting of time series. For instance, in statistics and econometrics domains the most common models are state-space (SS) models [2], [3]. In the physics domain the dominating class of models constitute nonlinear dynamical models [4]. In the machine learning area time series are usually modeled by neural networks, fuzzy systems and Gaussian Processes. An overview of time series forecasting can be found in [5].

One historically important subclass of the state-space models is autoregressive integrated moving average (ARIMA). It is still widely used and considered one of the best [6] in time series analysis. A *structural time series model* (STM) is a version of ARIMA where some time series components like trends and periodicities are imposed explicitly. It has an advantage over the pure ARIMA methodology that model misspecification is much less probable [3]. Moreover, STM is a way to introduce prior information and desired behavior into a time series model. Often a practitioner finds it difficult to consider and comprehend different forecasting methods from different domains. This paper is intended to shorten the gap between widely used STM models and Gaussian Processes (GPs) used in machine learning. The term structural time series model and state-space time series model are used interchangeably in this paper [2].

Basic state-space models are usually presented in the books [2], [3] as discrete time models with Gaussian errors. A *structural time series* framework allows to combine several basic state-space models into more complex ones. There are generalizations of discrete-time SS models to continuous time [3, Chap. 9] which after a certain procedure may be converted back to the discrete time. Since the errors in the basic SS models are assumed to be Gaussian, those are also GP models, however a direct systematic connection to Gaussian Processes used in machine learning is unknown to authors. The goal of this paper is to provide explicit connections between GP models and structural time series models.

Gaussian Processes are an important class of models in machine learning [7]. Modeling of time series has been widely addressed by GP community [8]. The modeling principles differ significantly from the state-space models. Modeling is done in continuous time and the main object to model is covariance function (and optionally mean function). There exist a known connection between continuous-discrete state space model and Gaussian process [9]. The advantage of representing the GP in SS form is that the inference can be done in $O(N)$ time where N is the number of data points, while the classic GP regression requires $O(N^3)$ operations. However, if the amount of data points is relatively small $N < 10000$, or we use some modification of standard GP, the difference in computational time can become negligible [10] on modern computers.

In this paper we derive several GP covariance functions which correspond to the main structural time series models. This explicit connection is useful for the researches with different background. State-space modelers can see that their methods are equivalent to certain Gaussian Processes and they can try to use various extension developed in the GP literature. GP specialists on the other hand can analyze the covariance functions corresponding to state-space models and borrow some ideas from there.

II. Structural Time Series Models and Gaussian Processes

Random (Stochastic) process is a collection of random variables $X_t, t \in T$ parametrized by the set T. If T is a set of integers ($T = \mathcal{Z}$) then the random process is discrete. If is real-valued ($T = \mathcal{R}$) the process is continuous.

The random process can be completely described by the infinite number of distribution function of the form $F_N(v_1, v_2, \cdots, v_N) = \mathtt{Pr}[X(t_1) < v_1, X(t_2) < v_2, \cdots, X(t_n) < v_N]$ for any positive integer N and arbitrary selected time points $t_1, t_2, \cdots, t_N$. Although this description is complete it is cumbersome. Therefore, often in practice only the first two distribution functions are taken into account.

These first two distribution functions allow to define the first moments of the random process: mean and covariance. Using these first two moments we can define the important class of random processes - Wide-Sense Stationary (WSS) Random Process. For a random process to be WSS it is sufficient that the mean is constant, variance is finite, and covariance function depends only on difference between time points. More detailed information can be found in any book about stochastic processes e.g. [1].

A. Gaussian Process (GP)

A *Gaussian process* is a random process $f(t)$ where for arbitrary selected time points $t_1, t_2, \cdots, t_N$ the probability distribution $\mathtt{p}[f(t_1), f(t_2), \cdots f(t_N)]$ is multivariate Gaussian.

To define a Gaussian process it is necessary to define a mean function $m(t) = \mathtt{E}[f(t)]$ and covariance function $\mathtt{Cov}[\mathtt{t}_1, \mathtt{t}_2] = \mathtt{E}[(f(t_1) - m(t_1))(f(t_2) - m(t_2))]$.

B. State-Space Models

The state-space model is the model of the form:

$$\begin{aligned} &\mathbf{z}_n = A_{n-1}\mathbf{z}_{n-1} + \mathbf{q}_n \quad \text{(state / dynamic equation)} \\ &y_n = H_n\mathbf{z_n} + \epsilon_n \quad \text{(measurement equation)} \end{aligned} \tag{1}$$

It is assumed that y_n (scalar) are the observed values of this random process. The noise terms $\mathbf{q}_n$ and ϵ_n are, in basic case, assumed to be Gaussian. This is the assumption we do in this paper. When the noise terms are Gaussian the random process y_n is also Gaussian and we find the explicit form of covariance function for the most popular state-space models.

The Kalman filter algorithm allows to make inference about the model (1). It computes the different conditional distributions of the hidden state $\mathbf{z_n}$ as well as a likelihood of the model [11].

In the model (1) the state variable $\mathbf{z}_n$ is assumed to be discrete. There exist equivalent versions where the state variable is continuous and it is called continuous-discrete state-space model [3]. The relationships between continuous-discrete state-space models and Gaussian processes have been recently highlighted [9]. In this paper the connection is made more explicit and clear.

C. Combining Models / Structural Time-Series (STS) Models

The structural time series framework is a way to construct state-space models and incorporate the desired properties or prior information into them. These properties are fixed level, trend, periodicity and quasi-periodicity (cyclicity) [2], [3]. The ability to incorporate prior information is an advantage of the STS modeling framework over more general ARIMA approach. A certain state-space model corresponds to each aforementioned property. Let's show how to combine these models additively. Suppose that $y_n = z_n^{trend} + z_n^{periodic} + \epsilon_n$, so y_n is a sum of trend and periodic component. It is possible to write it in a single state-space model:

$$\begin{aligned} &\begin{bmatrix} \mathbf{z}_n^{(tr)} \\ \mathbf{z}_n^{(per)} \end{bmatrix} = \begin{bmatrix} A_{n-1}^{(tr)} & 0 \\ 0 & A_{n-1}^{(per)} \end{bmatrix} \begin{bmatrix} \mathbf{z}_{n-1}^{(tr)} \\ \mathbf{z}_{n-1}^{(per)} \end{bmatrix} + \begin{bmatrix} \mathbf{q}_n^{(tr)} \\ \mathbf{q}_n^{(per)} \end{bmatrix} \\ &y_n = [H_n^{(tr)} H_n^{(per)}] \begin{bmatrix} \mathbf{z}_{n-1}^{(tr)} \\ \mathbf{z}_{n-1}^{(per)} \end{bmatrix} + \epsilon_n \end{aligned} \tag{2}$$

It can be easily seen that $\mathbf{z}_n^{(tr)}$ and $\mathbf{z}_n^{(per)}$ are uncorrelated random processes if their noise terms are uncorrelated. In this case the covariance function of y_n is:

$$\begin{aligned} &\mathtt{Cov}[y_k, y_{k+n}] = H_n^{(tr)} \mathtt{Cov}[z_k^{(tr)}, z_{k+n}^{(tr)}](H_n^{(tr)})^T + \\ &+ H_n^{(per)} \mathtt{Cov}[z_k^{(per)}, z_{k+n}^{(per)}](H_n^{(per)})^T + \delta_{(n=0)}\sigma_\epsilon^2 \end{aligned} \tag{3}$$

Here $\delta_{(n=0)}$ is a Kronecker delta which equals 1 when $n = 0$. So, the covariance is a sum of two covariances (matrices H are often 1) and a white noise term from the measurement equation. This useful property will be utilized in the subsequent sections.

III. Bayesian Linear Regression in State-Space Form

At first, recall the Bayesian Linear Regression (BLR) in the state-space form. Assume that we have N measurements $\mathbf{y} = [y_1, y_2, \ldots, y_N]^T$, which are observed at time points $\mathbf{t} = [t_1, t_2, \ldots, t_N]^T$. Further, assume that there is a linear dependency between measurements and time:

$$\begin{aligned} &y_k = \theta t_k + \epsilon_k \\ &\theta \sim \mathcal{N}(m_0, P_0) \quad \text{- prior of the parameter } \theta \\ &\epsilon_k \sim \mathcal{N}(0, \sigma_0^2) \quad \text{- Gaussian white noise} \end{aligned} \tag{4}$$

.

θ is a parameter of the model and the prior for it is $\theta \sim \mathcal{N}(m_0, P_0)$, ϵ is a Gaussian white noise: $\epsilon \sim \mathcal{N}(0, \sigma_0^2)$. In this formulation, the BLR provides us the posterior distribution of θ which we are not currently interested in. Besides, it provides the posterior predictive distribution which for any set of time points $t_1^\star, t_2^\star, \ldots, t_M^\star$ yields the distribution of corresponding measurements. It is well know [7] that the same posterior predictive distribution can be obtained by Gaussian Process Regression (GPR) with the kernel:

$$\mathbf{y} \sim \mathcal{GP}\left(m_0\mathbf{t}, P_0\mathbf{t}\mathbf{t}^T + \sigma_0^2 I\right) \tag{5}$$

We are interested in representing the BLR model in the state-space form because it allows us to look at the model in the sequential form when data arrives one by one. Moreover, the Kalman filter type inference which is the standard for the linear state-space models scales linearly with the number of samples, while Gaussian Process or batch BLR scales cubically [7]. There are several ways to express BLR in the state-space form, the one we are interested in is written below [11, p. 37]:

$$\begin{cases} \begin{bmatrix} x_k \\ \theta_k \end{bmatrix} = \begin{bmatrix} 1 & \Delta t_{k-1} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_{k-1} \\ \theta_{k-1} \end{bmatrix} \\ y_k = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_k \\ \theta_k \end{bmatrix} + \epsilon_k, \quad \text{where:} \end{cases}$$
$$x_0 = 0 \sim \mathcal{N}(0, 0), \quad \theta_0 \sim \mathcal{N}(m_0, P_0), \quad \epsilon_k \sim \mathcal{N}(0, \sigma_0^2)$$
$$\Delta t_{k-1} = t_k - t_{k-1}, \quad \text{and it is assumed that } t_0 = 0. \tag{6}$$

Now let's check that the state-space model listed above is indeed equivalent to Bayesian Linear Regression. Looking at the equation for θ we see that $\theta_k = \theta_{k-1}$ for all k, so it does not change with time. Since $t_0 = 0$ and $x_0 = 0$ we have that:

$$\begin{aligned} x_1 &= t_1\theta_0 = \theta t_1 \\ x_2 &= x_1 + (t_2 - t_1)\theta_1 = t_2\theta_1 + t_1(\theta_0 - \theta_1) = t_2\theta_1 = \theta t_2 \\ &\vdots \\ x_k &= x_{k-1} + (t_k - t_{k-1})\theta_{k-1} \\ &= t_k\theta_{k-1} + t_{k-1}(\theta_{k-2} - \theta_{k-1}) = t_k\theta_{k-1} = \theta t_k \end{aligned}$$

So, we see that $x_k = t_k\theta$ and if we insert the obtained result into the equation for y_k: $y_k = \theta t_k + \epsilon_k$ which exactly coincides with the original BLR formulation. Using the obtained state-space model we can find the covariance matrix of y_k. It would be the same as the one in Eq. (5). We are going to explicitly derive the covariance function for the more general state-space model in the next section.

In this section we have shown the equivalence of Gaussian Process Regression with covariance matrix in Eq. (5) and state-space formulation in Eq. (6). These two models are also equivalent to the Bayesian Linear Regression.

IV. General State-Space Model with Random Noise

In this section we derive the covariance function form for a more general state-space model than in the previous section. In the literature this model is called *Local Linear Trend Model* (LLLM). It is shown that this general state-space model under the special setting of parameters becomes equivalent to the well-known time series models: local level model, BLR, connection with the quasi-periodic (cyclic) model is very close as well. Derivation of covariance function provides us a useful connection to the Gaussian Process Regression for the aforementioned models. The general state-space model is:

$$\begin{cases} \begin{bmatrix} x_k \\ \theta_k \end{bmatrix} = \begin{bmatrix} 1 & \Delta t_{k-1} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_{k-1} \\ \theta_{k-1} \end{bmatrix} + \begin{bmatrix} q_k^{(1)} \\ q_k^{(2)} \end{bmatrix} \\ y_k = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_k \\ \theta_k \end{bmatrix} + \epsilon_k, \quad \text{where:} \quad \epsilon_k \sim \mathcal{N}(0, \sigma_0^2) \end{cases}$$
$$\Delta t_{k-1} = t_k - t_{k-1}, \quad \text{it is assumed that } t_0 = 0,$$
$$\begin{bmatrix} x_0 \\ \theta_0 \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} c_0 \\ m_0 \end{bmatrix}, \begin{bmatrix} K_0 & 0 \\ 0 & P_0 \end{bmatrix} \right)$$
$$\begin{bmatrix} q_k^{(1)} \\ q_k^{(2)} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} q_0^2\Delta t_{k-1} & 0 \\ 0 & g_0^2\Delta t_{k-1} \end{bmatrix} \right) \tag{7}$$

As we can see the difference with the state-space model from the previous section consist of extra noise terms in the dynamic (or state) equation. Another difference is non-zero prior distribution for the initial state variable x_0. Now it is distributed as a Gaussian random variable: $x_0 \sim \mathcal{N}(c_0, K_0)$.

A. Noise in Dynamic Equation

In this subsection the extra noise terms which appear in the dynamic equation are briefly discussed. In the two dimensional noise term $\mathbf{q} = \begin{bmatrix} q_k^{(1)} \\ q_k^{(2)} \end{bmatrix}$ the two components are independent and Gaussian distributed. Consider, for example, the first component $q_k^{(1)} \sim \mathcal{N}(0, q_0^2\Delta t_{k-1})$. It is a classical *Wiener process* [7] also called *standard Brownian motion* and is a generalization of a simple random walk to the continuous time when time measurements are not necessary equidistant. Its covariance function is $\texttt{Cov}[q_k^{(1)}(t_1), q_k^{(1)}(t_2)] = K_0 + q_0^2 \min(t_1, t_2)$ and it is a basic example of nonstationary Gaussian Process.

B. Covariance Function Derivation

Before commencing the derivation of the covariance function we consider an important property of the state-space model in Eq. (7). Denote:

$$A[\Delta t_{k-1}] = \begin{bmatrix} 1 & \Delta t_{k-1} \\ 0 & 1 \end{bmatrix} \tag{8}$$

We can easily verify that:

$$\begin{aligned} A[\Delta t_k]A[\Delta t_{k-1}] &= \begin{bmatrix} 1 & \Delta t_k \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & \Delta t_{k-1} \\ 0 & 1 \end{bmatrix} = \\ &= \begin{bmatrix} 1 & \Delta t_k + \Delta t_{k-1} \\ 0 & 1 \end{bmatrix} = A[\Delta t_k + \Delta t_{k-1}] \end{aligned} \tag{9}$$

This is a convenient property which will be utilized during the derivation and later in the Sec. V. Consider the covariance function:

$$\begin{aligned} &\texttt{Cov}[y_k, y_{k+n}] = \texttt{E}[(y_k - \texttt{E}[y_k])(y_{k+n} - \texttt{E}[y_{k+n}])] = \\ &= \texttt{E}[(x_k + \epsilon_k - \texttt{E}[x_k])(x_{k+n} + \epsilon_{k+n} - \texttt{E}[x_{k+n}])] = \\ &= \texttt{Cov}[x_k, x_{k+n}] + \delta_{(n=0)}\sigma_0^2 \end{aligned} \tag{10}$$

Therefore, we see that in order to find the covariance function of y_k it is enough to find the covariance function

of x_k and add the Kronecker symbol mentioned above. So, we can ignore the measurement equations right now and write our state-space model in the vector form:

$$\mathbf{z}_k = A[\Delta t_{k-1}]\mathbf{z}_{k-1} + \mathbf{q}_k; \;\; \mathbf{z}_k = \begin{bmatrix} x_k \\ \theta_k \end{bmatrix}; \;\; \mathbf{q}_k = \begin{bmatrix} q_k^{(1)} \\ q_k^{(2)} \end{bmatrix} \quad (11)$$

.

Lets express z_k through the initial conditions and noise terms:

$$\begin{aligned} &\mathbf{z}_k = A[\Delta t_{k-1}]\mathbf{z}_{k-1} + \mathbf{q}_k = \\ &= A[\Delta t_{k-1}](A[\Delta t_{k-2}]\mathbf{z}_{k-2} + \mathbf{q}_{k-1}) + \mathbf{q}_k = \\ &= A[\Delta t_{k-1} + \Delta t_{k-2}]\mathbf{z}_{k-2} + A[\Delta t_{k-1}]\mathbf{q}_{k-1} + \mathbf{q}_k = \ldots = \\ &= A[\Delta t_{k-1} + \Delta t_{k-2} + \ldots + \Delta t_0]\mathbf{z}_0 + A[\Delta t_{k-1} + \ldots \\ &\ldots + \Delta t_{k-2} + \Delta t_1]\mathbf{q}_1 + \ldots + A[\Delta t_{k-1}]\mathbf{q}_{k-1} + \mathbf{q}_k \end{aligned} \quad (12)$$

Here we use the property from Eq. (9) of the transition matrix. We see that $\mathbf{z}_k$ is a sum of terms each of which is a vector times matrix A with different arguments. Vectors are $\mathbf{z}_0, \mathbf{q}_0 \cdots \mathbf{q}_k$. Arguments of matrix A are also sum of terms Δt_i and the number of terms decreases one by one from k in $A[\Delta t_{k-1} + \Delta t_{k-2} + \ldots + \Delta t_0]$ to zero in front of $\mathbf{q}_k$. We can easily compute the mean of $\mathbf{z}_k$, taking into account the fact that the mean $\mathtt{E}[\mathbf{q}_i] = 0$ and expanding the expressions for Δt_i

$$\mathtt{E}[\mathbf{z}_k] = A[\Delta t_{k-1} + \Delta t_{k-2} + \ldots + \Delta t_0]\mathtt{E}[\mathbf{z}_0] = A[t_k]\begin{bmatrix} c_0 \\ m_0 \end{bmatrix} \quad (13)$$

Having the expression for $\mathbf{z}_k$ and its mean we can compute the covariance $\mathtt{Cov}[\mathbf{z}_k, \mathbf{z}_{k+n}] = \mathtt{E}[(\mathbf{z}_k - \mathtt{E}[\mathbf{z}_k])(\mathbf{z}_{k+n}^T - \mathtt{E}[\mathbf{z}_{k+n}^T])]$. The computation is quite straightforward using Eq. (12) and the fact that $\mathbf{z}_0$ and all $\mathbf{q}_i$ are mutually independent. The final answer is presented below:

$$\begin{aligned} &\mathtt{Cov}[\mathbf{z}_k, \mathbf{z}_{k+n}] = A[\Delta t_{k-1} + \Delta t_{k-2} + \ldots + \Delta t_0]\,\mathtt{Cov}[\mathbf{z}_0, \mathbf{z}_0] \\ &A[\Delta t_{k+n-1} + \Delta t_{k+n-2} + \ldots + \Delta t_0]^T + A[\Delta t_{k-1} + \ldots \\ &+ \Delta t_{k-2} + \Delta t_1]\,\mathtt{Cov}[\mathbf{q}_1, \mathbf{q}_1]\,A[\Delta t_{k+n-1} + \Delta t_{k+n-2} + \ldots \\ &+ \Delta t_1]^T + \cdots + \mathtt{I}\,\mathtt{Cov}[\mathbf{q}_k, \mathbf{q}_k]\,A[\Delta t_{k+n-1} + \ldots + \Delta t_k]^T \end{aligned} \quad (14)$$

As we see the expression is the sum of terms $A[\cdot]\,\mathtt{Cov}[\cdot,\cdot]\,A[\cdot]^T$ where the arguments of $A[\cdot]$ and $A[\cdot]^T$ are different while arguments in $\mathtt{Cov}[\cdot,\cdot]$ are the same.

Now suppose we want to compute all possible covariances up to some maximal time index N, i. e. $\mathtt{Cov}[\mathbf{z}_k, \mathbf{z}_n]$, where $1 \le k \le N$, $1 \le n \le N$. These covariances can be written in a matrix consisting of 2×2 blocks (because $\mathtt{Cov}[\mathbf{z}_k, \mathbf{z}_n]$ - one block is 2×2), and so in total it is a $2N \times 2N$ matrix. In the next formula we present the form of this matrix, and later the expression for the single components are provided. To simplify the notations and make them more vivid, suppose $N = 3$:

$$\begin{bmatrix} \mathtt{Cov}[\mathbf{z}_0, \mathbf{z}_0] & \mathtt{Cov}[\mathbf{z}_0, \mathbf{z}_1] & \mathtt{Cov}[\mathbf{z}_0, \mathbf{z}_2] & \mathtt{Cov}[\mathbf{z}_0, \mathbf{z}_3] \\ \mathtt{Cov}[\mathbf{z}_1, \mathbf{z}_0] & \mathtt{Cov}[\mathbf{z}_1, \mathbf{z}_1] & \mathtt{Cov}[\mathbf{z}_1, \mathbf{z}_2] & \mathtt{Cov}[\mathbf{z}_1, \mathbf{z}_3] \\ \mathtt{Cov}[\mathbf{z}_2, \mathbf{z}_0] & \mathtt{Cov}[\mathbf{z}_2, \mathbf{z}_1] & \mathtt{Cov}[\mathbf{z}_2, \mathbf{z}_2] & \mathtt{Cov}[\mathbf{z}_2, \mathbf{z}_3] \\ \mathtt{Cov}[\mathbf{z}_3, \mathbf{z}_0] & \mathtt{Cov}[\mathbf{z}_3, \mathbf{z}_1] & \mathtt{Cov}[\mathbf{z}_3, \mathbf{z}_2] & \mathtt{Cov}[\mathbf{z}_3, \mathbf{z}_3] \end{bmatrix} = \\ = \mathcal{P}\{T\}\; D_0\; (\mathcal{P}\{T\})^T \quad (15)$$

We are not interested in computing the first row and column of this covariance matrix since variable $\mathbf{z}_0$ does not correspond to any real observation, it is just an initial random variable. Also, $\mathcal{P}\{T\}$ in the above formula equals:

$$\mathcal{P}\{T\} = \begin{bmatrix} A[0] & 0 & 0 & 0 \\ A[\Delta t_0] & A[0] & 0 & 0 \\ A[\Delta t_1 + \Delta t_0] & A[\Delta t_1] & A[0] & 0 \\ A[\Delta t_2 + \Delta t_1 + \Delta t_0] & A[\Delta t_2 + \Delta t_1] & A[\Delta t_2] & A[0] \end{bmatrix} \quad (16)$$

Here each element of the $\mathcal{P}\{T\}$ matrix is a (2×2) block. The notation $\mathcal{P}\{T\}$ means that some matrix operator $\mathcal{P}\{\cdot\}$ is applied to the matrix T. Currently we are not specifying what are $\mathcal{P}\{\cdot\}$ and T, it is done later in this section when we obtain covariances of x_k.

Matrix D_0 in Eq. (15) is a block diagonal matrix written below:

$$D_0 = \begin{bmatrix} \mathtt{Cov}[\mathbf{z}_0, \mathbf{z}_0] & 0 & 0 & 0 \\ 0 & \mathtt{Cov}[\mathbf{q}_1, \mathbf{q}_1] & 0 & 0 \\ 0 & 0 & \mathtt{Cov}[\mathbf{q}_2, \mathbf{q}_2] & 0 \\ 0 & 0 & 0 & \mathtt{Cov}[\mathbf{q}_3, \mathbf{q}_3] \end{bmatrix} \quad (17)$$

It can be verified that expressions in Eq. (14) and Eq. (15) are equal. Covariances $\mathtt{Cov}[\mathbf{q}_i, \mathbf{q}_i]$ are diagonal matrices shown in Eq. (7). Thus, we have derived the expression for $\mathtt{Cov}[\mathbf{z}_k, \mathbf{z}_n]$. However we are not interested in it as is. We would like to know the covariances $\mathtt{Cov}[x_k, x_n]$ because they are directly related with covariances of the observed variable y_k which is shown in Eq. (10). It means that we are interested in the covariance matrix consisting of odd columns and rows of the matrix $\mathcal{P}\{T\}\; D_0\; (\mathcal{P}\{T\})^T$. To derive it consider the structure of the expression which is the main building block in covariance functions in Eq. (14) and Eq. (15):

$$\begin{aligned} &A\left[\sum \Delta t_m\right]\begin{bmatrix} q_0^2\Delta t_{i-1} & 0 \\ 0 & g_0^2\Delta t_{i-1} \end{bmatrix} A^T\left[\sum \Delta t_n\right] = \\ &= \begin{bmatrix} 1 & \sum \Delta t_m \\ 0 & 1 \end{bmatrix}\begin{bmatrix} q_0^2\Delta t_{i-1} & 0 \\ 0 & g_0^2\Delta t_{i-1} \end{bmatrix}\begin{bmatrix} 1 & 0 \\ \sum \Delta t_n & 1 \end{bmatrix} = \\ &= \begin{bmatrix} \boxed{q_0^2\Delta t_{i-1} + g_0^2\Delta t_{i-1}\left(\sum \Delta t_m\right)\left(\sum \Delta t_n\right)} & \cdot \\ \cdot & \cdot \end{bmatrix} \end{aligned} \quad (18)$$

In the above formula we are interested in the top left element which is emphasized by the rectangle, because it gives exact covariances of $\mathtt{Cov}[x_k, x_n]$ from the covariances $\mathtt{Cov}[\mathbf{z}_k, \mathbf{z}_n]$. Now we see that the required covariance consist of two parts

which correspond to the two terms in the sum above. The first term is affected only by the top diagonal entry of the middle matrix in the initial product. The second term is affected by the bottom diagonal entry and the arguments of the matrices A. Now we are ready to write the required correlations by looking at Eq. (14), analyzing contributions of each term there and taking into account Eq. (18). Representation (15) is also useful in deriving the second part of the following result:

$$\begin{bmatrix} \texttt{Cov}[x_1,x_1] & \texttt{Cov}[x_1,x_2] & \texttt{Cov}[x_1,x_2] \\ \texttt{Cov}[x_2,x_1] & \texttt{Cov}[x_2,x_2] & \texttt{Cov}[x_2,x_3] \\ \texttt{Cov}[x_3,x_1] & \texttt{Cov}[x_3,x_2] & \texttt{Cov}[x_3,x_3] \end{bmatrix} = \texttt{Cov}_1[\cdot] + \texttt{Cov}_2[\cdot] \tag{19}$$

where:

$$\begin{aligned} &\texttt{Cov}_1[x_k, x_{k+n}] = K_0 + q_0^2\Delta t_0 + q_0^2\Delta t_1 + \dots \\ &+ q_0^2\Delta t_{k-1} = K_0 + q_0^2 t_k \end{aligned} \tag{20}$$

Another way to write $\texttt{Cov}_1[x_k, x_{k+n}]$ is:

$$\texttt{Cov}_1[x_k, x_{k+n}] = K_0 + q_0^2 \min(x_k, x_{k+n}) \tag{21}$$

The expression for $\texttt{Cov}_2[\cdot]$ is written below:

$$\texttt{Cov}_2[\cdot] = TDT^T$$

where:

$$T = \begin{bmatrix} 0 & 0 & 0 & 0 \\ \Delta t_0 & 0 & 0 & 0 \\ \Delta t_1 + \Delta t_0 & \Delta t_1 & 0 & 0 \\ \Delta t_2 + \Delta t_1 + \Delta t_0 & \Delta t_2 + \Delta t_1 & \Delta t_2 & 0 \end{bmatrix} \tag{22}$$

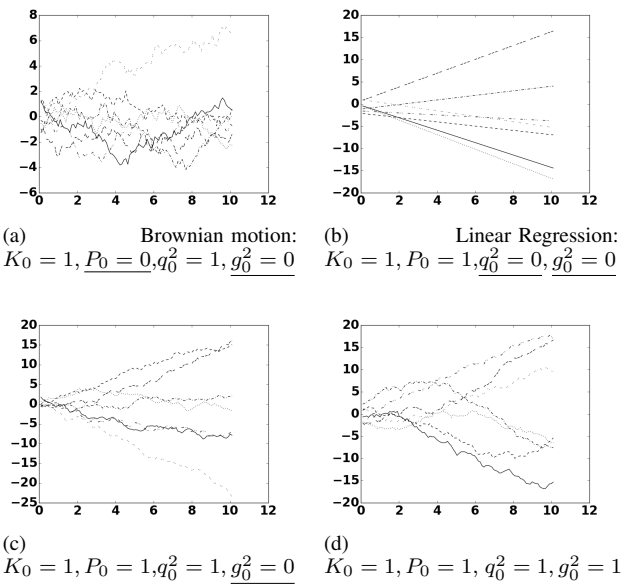
$$D = \begin{bmatrix} P_0 & 0 & 0 & 0 \\ 0 & g_0^2\Delta t_0 & 0 & 0 \\ 0 & 0 & g_0^2\Delta t_1 & 0 \\ 0 & 0 & 0 & g_0^2\Delta t_2 \end{bmatrix}$$

The matrix T can also be represented as:

$$T = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 & 0 \\ \Delta t_0 & 0 & 0 & 0 \\ \Delta t_1 & \Delta t_1 & 0 & 0 \\ \Delta t_2 & \Delta t_2 & \Delta t_2 & 0 \end{bmatrix} \tag{23}$$

In the Eq. (22) we must ignore the first row and the first column so that the resulting $\texttt{Cov}_2[\cdot]$ matrix is 3×3. It is possible to write this formula directly by 3×3 matrices but this form is useful for the derivation of quasi-periodic (cyclic) covariance in the next section. The Eq. (19) is the final answer for the covariance function of the model stated in Eq. (7). Now given time points $\mathbf{t} = [t_1, t_2, \dots, t_N]^T$ we can compute the covariance function, the mean function which is given in Eq. (13) and use Gaussian Process Regression in a regular way. The sample paths from GP with this covariance function are presented on Fig. 1.

Several standard structural time series models are actually versions of the general model described above, they are listed later in this section. The Bayesian Linear Regression model considered in Eq. (6) in Section III is a lucid representative as well. If we set $c_0 = 0, K_0 = 0, q_0^2 = g_0^2 = 0$ as in the expression for BLR, then $\texttt{Cov}_1 = 0$ and in $\texttt{Cov}_2$ only the first element in the diagonal matrix is non-zero. Then, expanding all Δt we easily get that the covariance becomes equal to the one of BLR (Eq. 5).



(a) Brownian motion: $K_0 = 1, \underline{P_0 = 0}, q_0^2 = 1, \underline{g_0^2 = 0}$

(b) Linear Regression: $K_0 = 1, P_0 = 1, \underline{q_0^2 = 0}, \underline{g_0^2 = 0}$

(c) $K_0 = 1, P_0 = 1, q_0^2 = 1, \underline{g_0^2 = 0}$

(d) $K_0 = 1, P_0 = 1, q_0^2 = 1, g_0^2 = 1$

Fig. 1. GP sample paths of general covariance (LLLM) for various parameter values. Underline emphasize parameters which equal zero.

C. Local Level Model

Local Level Model (LLM) is the simplest model among the structural time series models [2]. Its standard representation in the literature is:

$$\begin{cases} x_k = x_{k-1} + q_k; & q_k \sim \mathcal{N}(0, q_0^2) \\ y_k = x_k + \epsilon_k; & \epsilon_k \sim \mathcal{N}(0, \sigma_0^2) \end{cases} \tag{24}$$
$$x_0 \sim \mathcal{N}(c_0, K_0)$$

As we can see this is a random walk expressed by dynamic variable x_k additionally submerged into white noise ϵ_k. The covariance of this model as was mentioned in IV-A is: $\texttt{Cov}[q_k^{(1)}(t_1), q_k^{(1)}(t_2)] = K_0 + q_0^2 \min(t_1, t_2)$. Now if we generalize this model to arbitrary time intervals it can be written as:

$$\begin{cases} \begin{bmatrix} x_k \\ \theta_k \end{bmatrix} = \begin{bmatrix} 1 & \Delta t_{k-1} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_{k-1} \\ \theta_{k-1} \end{bmatrix} + \begin{bmatrix} q_k^{(1)} \\ q_k^{(2)} \end{bmatrix} \\ y_k = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_k \\ \theta_k \end{bmatrix} + \epsilon_k, \quad \text{where:} \quad \epsilon_k \sim \mathcal{N}(0, \sigma_0^2) \end{cases}$$
$$\Delta t_{k-1} = t_k - t_{k-1}, \quad \text{it is assumed that } t_0 = 0,$$
$$\begin{bmatrix} x_0 \\ \theta_0 \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} c_0 \\ \boxed{0} \end{bmatrix}, \begin{bmatrix} K_0 & 0 \\ 0 & \boxed{0} \end{bmatrix} \right)$$
$$\begin{bmatrix} q_k^{(1)} \\ q_k^{(2)} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} q_0^2\Delta t_{k-1} & 0 \\ 0 & \boxed{0} \end{bmatrix} \right) \tag{25}$$

The parameters which are nullified with respect to the general model (LLLM) are denoted by boxes. We can see that the equation for θ_k is a redundant equation because θ_0 is initialized as zero and corresponding noise term is also zero. The covariance function could also be obtained by using the formula for the general covariance function and putting to zeros the corresponding coefficients.

In the end of this section it is worth to mention that although the LLM is the simplest structural time series model, it can be successfully applied to the real world data [2, p. 16].

D. Local Linear Trend Model (LLLM)

The next model we consider is called Local Linear Trend Model (LLLM). As was previously said it is the same as general model discussed in this section. The slope variable θ_k is changing by random walk, and the coordinate variable x_k has also random walk components similarly to LLM. One can consider a simplification of LLLM: only θ_k changes by random walk but x_k does not. As stated in [2, p. 44] this simplified model produces smoother sample paths than general LLLM.

V. Periodic and Quasi-Periodic (Cyclic) Modeling

In the structural time series framework there are several models for periodicities and cycles (quasi-periodicities). We consider here the most popular model which is frequently used for cyclic modeling [2, p. 44]:

$$\begin{cases} \begin{bmatrix} x_k \\ x_k^\star \end{bmatrix} = \begin{bmatrix} \cos(\omega_c \Delta t_{k-1}) & \sin(\omega_c \Delta t_{k-1}) \\ -\sin(\omega_c \Delta t_{k-1}) & \cos(\omega_c \Delta t_{k-1}) \end{bmatrix} \begin{bmatrix} x_{k-1} \\ x_{k-1}^\star \end{bmatrix} + \\ \qquad\qquad + \begin{bmatrix} q_k^{(1)} \\ q_k^{(2)} \end{bmatrix} \\ y_k = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_k \\ x_k^\star \end{bmatrix} + \epsilon_k, \quad \text{where:} \quad \epsilon_k \sim \mathcal{N}(0, \sigma_0^2) \end{cases}$$
$$\Delta t_{k-1} = t_k - t_{k-1}, \quad \text{it is assumed that } t_0 = 0,$$
$$\begin{bmatrix} x_0 \\ x_0^\star \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} m_0 \\ m_0 \end{bmatrix}, \begin{bmatrix} P_0 & 0 \\ 0 & P_0 \end{bmatrix} \right)$$
$$\begin{bmatrix} q_k^{(1)} \\ q_k^{(2)} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} g_0^2 \Delta t_{k-1} & 0 \\ 0 & g_0^2 \Delta t_{k-1} \end{bmatrix} \right) \tag{26}$$

The presented equations are already a generalization of discrete time model which is usually encountered in the books [2] [3], to the continuous time model. The Δt_i are used to express the uneven time sampling. If the sampling is even all the Δt_i equal to one.

Notice that the model is completely symmetric with respect to the vector $\mathbf{x} = [x_k, x_k^\star]$. The initial conditions are symmetric and the noise is symmetric. If we suppose no noise in the model then it is straightforward to show that the covariance function of x_k is:

$$Cov[x_k, x_{k+n}] = P_0^2 \cos[\omega_c (t_{k+n} - t_k)] \tag{27}$$

So, it is a periodic covariance function. The process x_k can be considered as a random process where randomness originates only from the initial conditions. This process is also wide sense stationary since the covariance function depend on the difference of the time points. Again if we suppose that the the noise vector is absent from the dynamic model (i.e. $q_0^2 = 0$) then the x_n variable is just a cosine wave. This can be deduced by considering x_1 which is a sum of cosine and sine with coefficients which are initial values: $x_0, x_0^\star$. This sum can be represented as a cosine wave where the phase depend on those coefficients. Also, we need to consider the property (28) which is discussed soon. Hence, without extra white noise the x_n is a cosine wave, however with the presence of white noise the deviations from the strict periodicity are possible.

A. Quasi-Periodic (Cyclic) Covariance function

Let's consider the dynamic matrix. Its spectral decomposition is written below:

$$\begin{aligned} &A[\Delta t_{k-1}] = \\ &\begin{bmatrix} \cos(\omega_c \Delta t_{k-1}) & \sin(\omega_c \Delta t_{k-1}) \\ -\sin(\omega_c \Delta t_{k-1}) & \cos(\omega_c \Delta t_{k-1}) \end{bmatrix} = \\ &= \frac{1}{2} \begin{bmatrix} 1 & 1 \\ i & -i \end{bmatrix} \begin{bmatrix} e^{i\omega_c \Delta t_{k-1}} & 0 \\ 0 & e^{-i\omega_c \Delta t_{k-1}} \end{bmatrix} \begin{bmatrix} 1 & -i \\ 1 & i \end{bmatrix} \end{aligned} \tag{28}$$

Using this it is easy to show that the property (9) is valid again. Therefore, we conclude that all the results which are derived in the section IV and which are based on the property (9) are also valid. In particular expressions (14) and (15) are valid which already give us the results for the covariance matrices of $\mathbf{z}_k = [x_k, x_k^\star]^T$. Repeating the same steps as are done to dive the covariance formula (19) we can derive the similar formula for the cyclic model (26). The derived covariance function consist of two parts as in Eq. (19), however we must exclude the first row and the first column form the covariance matrix provided below similarly to formula (22). The two parts $\texttt{Cov}_1$ and $\texttt{Cov}_2$ are written below:

$$\texttt{Cov}_1[\cdot] = \mathcal{L}\{Cos\{T\}\}\ D\ (\ \mathcal{L}\{Cos\{T\}\}\ )^T \tag{29}$$

In this expression matrices T and D are exactly the same as in Eq. (22). There are two new matrix operations which are nested: $\mathcal{L}\{\cdot\}$ and $Cos\{\cdot\}$. The first one leaves the lower triangular part (including the main diagonal) of the argument matrix intact, and put zeros to the upper-triangular part. The second one applies cos function element-wise to the matrix.

Similarly,

$$\texttt{Cov}_2[\cdot] = \mathcal{L}\{Sin\{T\}\}\ D\ (\ \mathcal{L}\{Sin\{T\}\}\ )^T \tag{30}$$

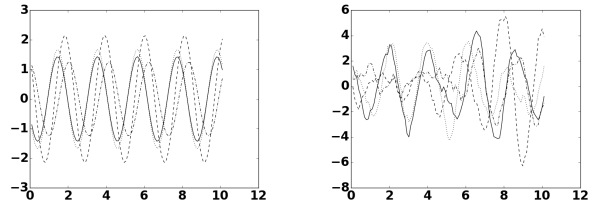
Here, $Sin\{\cdot\}$ is used instead of $Cos\{\cdot\}$ with the similar meaning - element-wise application of sin function to the argument matrix.

Thus, we obtained the expression for the covariance matrix of the quasi-periodic model Eq. (26). Hence, it is now possible to model this cyclic state-space model as a Gaussian Process

with the obtained covariance function. The GP sample paths with cyclic covariance function are shown on the Fig. 2a 2b.

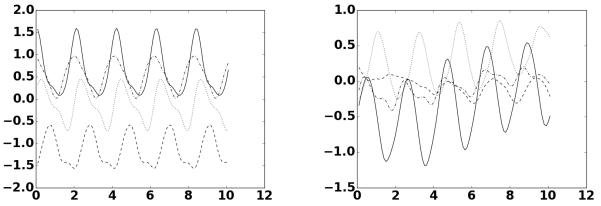
If the data contains several frequencies or periodicities then the corresponding state-space models can be combined in the measurement equations for y_k, see subsection II-C. In GP regression this is equivalent to the summation of covariance functions.

Also, if the periodic pattern in the data is not close to cosine wave then we need to take more harmonics to model this pattern. Then we need to combine several frequencies: $\omega_c, 2\omega_c, \cdots, k\omega_c$ (k harmonics) as described in the previous paragraph.



(a) Zero Noise: $\omega_c = 3, P_0 = 1, g_0^2 = 0$

(b) Non-Zero Noise: $\omega_c = 3, P_0 = 1, g_0^2 = 1$



(c) Standard Periodic Covariance [7]:

(d) Quasi-Periodic Covariance [7]:

Fig. 2. Quasi-periodic (cyclic) sample paths.

B. Gaussian Periodic covariance function

It is interesting to compare the periodicity modeling approach proposed above with the approach used in Gaussian Process Regression (GPR). In GPR there exist a periodic covariance function [7, p. 92] which is expressed as:

$$Cov[t_1, t_2] = \exp\left(-\sin^2\left(\frac{\omega_c(t_1 - t_2)}{2}\right)\right) \qquad (31)$$

This is also a periodic covariance function with a frequency ω_c. Sample paths from GP with periodic covariance are presented on Fig. 2c. Since the covariance function is periodic it is possible to represent it as a Fourier series with a harmonics $\omega_c, 2\omega_c, 3\omega_c, \cdots$. This is exactly the case which can be represented by combining state-space models and which is described in the previous subsection. Thus, the periodic covariance function used in GPR can be represented by equivalent random process in the state-space form. It is done in the paper [12].

In the same paper the question of representing the quasi-periodic covariance function is also discussed. The quasi-periodic covariance function is a multiplication of some stationary covariance functions (e.g. Matern covariance) [12] and the periodic one in Eq. (31). The random process which is modeled by quasi-periodic covariance function has no fixed period, the period length is fluctuating. Sample paths of quasi-periodic covariance are shown on Fig. 2d. By using noise $\mathbf{q_k}$ we also deviate from strict periodicity, however there is no direct correspondence between model in Eq. (26) and quasi-periodic covariance function in the paper [12]. This question requires further investigation and is not touched here anymore.

VI. Damped Trend Model

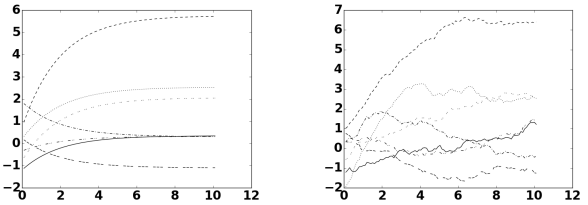
In this section we consider damping trend model. It is similar to the general model Eq. (7), except that a slope gradually decreases. Here we present only the dynamic equation for this model because the rest is the same as in Eq. (7).

$$\begin{bmatrix} x_k \\ \theta_k \end{bmatrix} = \begin{bmatrix} 1 & \Delta t_{k-1} \\ 0 & \boxed{\phi} \end{bmatrix} \begin{bmatrix} x_{k-1} \\ \theta_{k-1} \end{bmatrix} + \begin{bmatrix} q_k^{(1)} \\ q_k^{(2)} \end{bmatrix} \qquad (32)$$

The damping factor is denoted by the box around it. It must satisfy $0 < \phi < 1$ so that the trend to be damping.

Next we present the covariance function of this damping trend. The derivation is omitted because it is very similar to the derivation LLLM model Eq. (19). The first part of the covariance $\mathtt{Cov}_1[\cdot]$ is the same as in Eq. (21). The second is also similar to Eq. (22) except that matrix T must be substituted to:

$$T = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & \phi & 0 & 0 \\ 1 & \phi & \phi^2 & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 & 0 \\ \Delta t_0 & 0 & 0 & 0 \\ \Delta t_1 & \Delta t_1 & 0 & 0 \\ \Delta t_2 & \Delta t_2 & \Delta t_2 & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & \frac{1}{\phi} & 0 & 0 \\ 0 & 0 & \frac{1}{\phi^2} & 0 \end{bmatrix} \qquad (33)$$



(a) No noise: $\phi = 0.95$ $K_0 = P_0 = 1,\ q_0^2 = g_0^2 = 0$

(b) Non-Zero Noise: $\phi = 0.95$ $K_0 = P_0 = 1,\ q_0^2 = g_0^2 = 0.05$

Fig. 3. Damped sample paths.

As before, to obtain the final covariance we must discard the first row and the first column from the resulting covariance. It is worth noting that the model is not completely adapted to the continuous time. The reason is that damping factor ϕ does not depend on the time interval Δt_{k-1} between two consecutive measurements y_k. So, strictly speaking the covariance Eq. (33) is valid only when all Δt_i are the same. It is possible to extend the derived covariance to cover the general case as well,

however for simplicity of presentation and space constrains it is not done here. Sample paths from GP with a damped trend covariance are given on Fig. 3.

VII. External Variables

So far we have considered the modeling of y_k with respect to time. These might include local level model, deterministic or stochastic trend, one or more periodicities etc.. These time patterns are modeled by state-space model for variable x_k. Quite often, there might be other explanatory variables e.g. day of the week. We can also include them into the model. Suppose that y_k depends linearly on a set of explanatory variables $\mathbf{z} = [z_1, z_2, \cdots z_m]^T$:

$$y_k = x_k + \mathbf{b}^T \mathbf{z}_k + \epsilon_k \tag{34}$$

In the formula above $\mathbf{b}$ is some vector of parameters. Denote also that $f_k = \mathbf{b}^T \mathbf{z}_k$. If we assume that the vector $\mathbf{b}$ is a vector of constant but unknown parameters we again can express this model both in state-space and in GP forms. To express in the state-space form it is enough to assign $\mathbf{b}$ as a state variable with unit dynamic (transition) matrix and no noise. Then we need to combine this state-space model with the one for x_k. It is shown in the Sec. II-C how to do that.

It is easy to check that if x_k and f_k are independent random processes:

$$Cov[y_k, y_{k+n}] = Cov[x_k, x_{k+n}] + Cov[f_k, f_{k+n}] \tag{35}$$

So, the covariance function is the sum of two covariance functions for x_k and z_k. In our case, x_k and $\mathbf{b}^T \mathbf{z}_k$ are independent. The randomness to the second process is introduced only through the prior distribution of parameters $\mathbf{b}$, which is independent of a randomness in x_k. We can also see that the dependency of $\mathbf{z}_k$ is exactly Bayesian Linear Regression introduced in the Sec. III. The only difference is that now $\mathbf{z}_k$ is possibly multidimensional vector. Anyway, the covariance function of BLR part is:

$$\mathbf{f} \sim \mathcal{GP}\left(0, z_0^2 Z Z^T\right) \tag{36}$$

where it is assumed that:

$$\mathbf{b} \sim \mathcal{N}(0, z_0^2 I) \quad - \text{prior} \tag{37}$$

And Z is a matrix composed of vectors $\mathbf{z}_k$ row-wise. This is analogous to the Eq. (5) except that the noise term is missing in this covariance.s

VIII. ARMA Models

The discrete WSS random processes are frequently modeled as an Auto-Regressive Moving-Average (ARMA) process [6]:

$$x_n + a_1 x_{n-1} + a_2 x_{n-2} + \cdots + a_p x_{n-p} = b_0 \xi_0 + b_1 \xi_1 + \cdots + b_q \xi_q \tag{38}$$

Where a_i and b_i are some real valued coefficients, ξ_i are independent Gaussian white noise with unit variance. It is straightforward to write this ARMA(p,q) model in the state-space form [2]. We do not present it here due to the space constraints.

The process in Eq. (38) is stationary under some conditions on the coefficients and its power spectrum is:

$$P_x(\omega) = \frac{|B_q(e^{i\omega})|^2}{|A_p(e^{i\omega})|^2} \tag{39}$$

where $A_p(e^{i\omega})$ and $B_q(e^{i\omega})$ are polynomials with corresponding coefficients from Eq. (38). For instance:

$$A_p(e^{i\omega}) = 1 + a_1 e^{i\omega} + a_2 e^{i2\omega} + \cdots + a_p e^{ip\omega} \tag{40}$$

Also we can see that the this power spectrum is periodic with the period 2π because exponent is the periodic function with this period.

The ARMA processes can be generalized to the continuous time: ARMA process in continuous time has rational spectral density of the form Eq. (39) except that instead of the argument $e^{i\omega}$ the argument $i\omega$ must be used. There must be extra requirements in order that the rational function represents a power spectrum of a random process. Namely, numerator and denominator do not have common roots, there can not be real roots, and that $p \geq q + 1$ [1, p. 133].

It is possible to write a covariance function of this random process by computing the Fourier transform of the power spectrum. The covariance function is a sum of the terms which depend on the roots of the denominator of the $|A_p(i\omega)|^2$. Since roots appear in conjugate pairs and every pair must be taken into account only once we consider only the roots with positive imaginary parts. We write here formulas for the case when all the roots are of unit multiplicity, for the general case see e.g. [1]. Each fully imaginary root (with positive imaginary part) $i\alpha_k$ brings the following term to the sum:

$$r_1(\tau) = C e^{-\alpha_k |\tau|} \tag{41}$$

This is an exponential covariance function. Each complex root (with positive imaginary part) $\alpha_k^{(1)} + i\alpha_k^{(2)}$ introduces a term:

$$\begin{aligned} &r_2(\tau) = C e^{-\alpha_k^{(2)} |\tau|} cos(\alpha_k^{(1)} |\tau| - \psi) \qquad \text{where:} \\ &|\psi| \leq \tan^{-1}(\alpha_k^{(2)} / \alpha_k^{(1)}) \quad - \text{some phase shift.} \end{aligned} \tag{42}$$

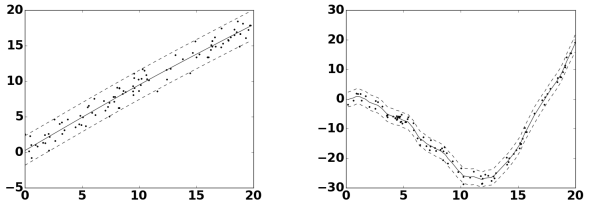
For instance the last covariance function is a correlation function of continuous ARMA(2,1) process. More information on this topic can be found in [1], [13], [7].

IX. Experiments

In this section we perform a number of basic experiments in order to demonstrate that the derived in Sec. IV, V, VI covariance functions are applicable in the GP regression framework and to show that the results are equivalent to state-space modeling. The proposed kernels are applied to several artificially generated datasets and it is shown that GP regression results are meaningful. Furthermore, we compare the GP regression approach and the state-space approach for the Nile Water Level [14] dataset which is frequently used in

the time series literature. It is shown on the simple example of LLM model from IV that the modeling results are equivalent.

All new kernels proposed in this paper have been implemented as an add-ons to the *GPy toolbox*. This a powerful toolbox for Gaussian Process modeling and inference [15]. Crucial part of GP inference is finding hyper-parameters of a kernel. A standard way to do this is to find maximum (MAP estimate) of marginal log-likelihood [7, p. 112]. In the subsequent experiments maximum is searched by BFGS algorithm. Since marginal log-likelihood is non-convex function, each optimization procedure is run 10 times with different random initial conditions. The hyper-parameters which produce the highest marginal log-likelihood are considered as final answer.



(a) Dataset 1: $q_0^2 = g_0^2 = 0$ (b) Dataset 2: $q_0^2 = g_0^2 = 1$

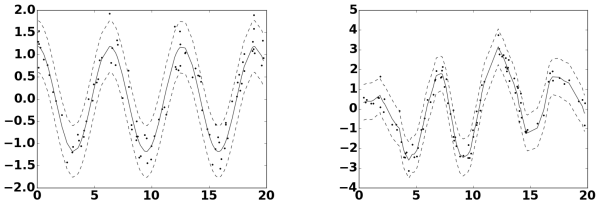
Fig. 4. GP regression with general state-space kernel Eq. (19)

TABLE I
GP REGRESSION WITH GENERAL STATE-SPACE KERNEL EQ. (12)

	Dataset 1		Dataset 2	
Param. name	True value	MAP estimation	True value	MAP estimation
K_0	1.0	0.14	1.0	$5.79 * 10^{-7}$
P_0	1.0	0.09	1.0	$2.02 * 10^{-7}$
q_0^2	0.0	$6.05 * 10^{-8}$	1.0	1.53
g_0^2	0.0	$8.08 * 10^{-4}$	1.0	3.17
σ_0^2	1.0	1.08	1.0	1.39

The first experiment is designed to test the general state-space covariance Eq. (19). Two datasets from the model Eq. (7) are generated each containing 100 points. In the first dataset the parameters $q_0^2 = 0, g_0^2 = 0$ which means the absence of noise of the dynamic model and equivalence to BLR Eq. (4). In the second dataset noise parameters are $q_0^2 = 1, g_0^2 = 1$, so they are non-zero. All the remaining parameters K_0, P_0, σ_0^2 equal to 1, and c_0, m_0 equal to zero. The results of GP regression modeling with general state-space covariance Eq. (19) are presented in Table I and Figure 4.

As we can see the Table I and Figure 4. The modeling provides quite feasible results. All parameters except K_0 and P_0 are estimated with reasonable accuracy for this kind of modeling. The large error in estimation of K_0 and P_0 probably stems from the fact that the values of corresponding random variables are observed only once during the generation of initial state variables. This situation is quite typical for subsequent experiments as well.



(a) Dataset 1: $g_0^2 = 0$ (b) Dataset 2: $g_0^2 = 1$

Fig. 5. GP regression with quasi-periodic kernel Eq. (22), (24)

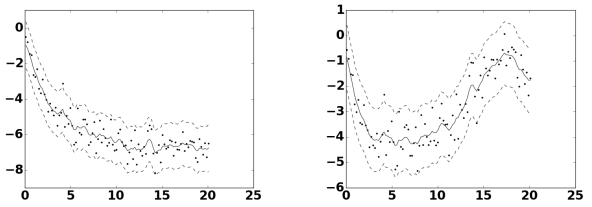
TABLE II
GP REGRESSION WITH PERIODIC KERNEL EQ. (22), (24)

	Dataset 1		Dataset 2	
Param. name	True value	MAP estimation	True value	MAP estimation
ω_c	1.0	0.99	1.0	1.04
P_0	1.0	0.70	1.0	$1.64 * 10^{-7}$
g_0^2	0.0	$1.23 * 10^{-15}$	0.1	0.44
σ_0^2	0.1	0.09	0.1	0.16

Similar experiment is performed for the periodic (or cyclic) covariance function which is a sum of Eq. (29) and Eq. (30). Dataset 1 which is generated with no noise in dynamic model correspond to purely periodic random process. The dataset 2 which has this noise correspond to quasi-periodic or cyclic behavior. Results of GP regression with periodic kernel is presented in Table II and Figure 5. They are also very reasonable. It is more important that noise levels and angular frequency of oscillations are estimated well.

The last kernel we experimented with is the damped trend model in Eq. (33). If there is no noise in the dynamic equation then the data generated by the model in Eq. (32) is the damped trend, if noise is present then the generated data is more complex. Experimental results for this two cases are present in Table III and Figure 6. We can see that the estimated parameters are much less accurate. Perhaps, this happens because this is the most complex model we have considered so far (in terms of number of parameters) and the same data can be generated by several different sets of parameters. Therefore, the true set is harder to identify. Anyway the plots on Figure 6 show quite feasible results.

Finally, we want to demonstrate that the GP regression approach complemented with the kernels proposed in this paper is completely equivalent with state-space modeling approach. This is demonstrated on the classical Nile Water Level dataset [14] which contains 100 years (100 data points) of measurements. The state-space inference is performed by Kalman Filtering (KF) and Rauch-Tung-Striebel (RTS) smoother. We have taken the simple Local Level Model (LLM) from Eq. (25). Often this model is used as a starting point for time series analysis. The results of modeling and forecasting of Nile dataset are presented on Figure 7. From the figure it is impossible to see any difference between approaches.



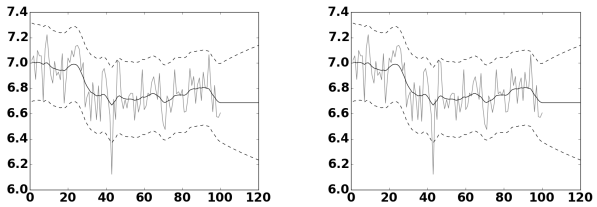
(a) Dataset 1: $q_0^2 = g_0^2 = 0$ (b) Dataset 2: $q_0^2 = g_0^2 = 0.01$

Fig. 6. GP regression with damped trend kernel Eq. (33)

TABLE III
GP REGRESSION WITH DAMPED TREND KERNEL EQ. (33)

	Dataset 1		Dataset 2	
Param. name	True value	MAP estimation	True value	MAP estimation
ϕ	0.94	0.02	0.94	0.68
K_0	3.0	1.50	3.0	0.4
P_0	1.0	0.5	1.0	0.85
q_0^2	0.0	0.46	0.01	0.3
g_0^2	0.0	0.5	0.01	0.19
σ_0^2	0.4	0.33	0.4	0.33

Analysis of numerical data, which is not presented here also shows that the difference is negligible. Hence, we have shown experimentally for one model that state-space approach and GP regression can be used interchangeably depending on the modeler's preferences and other relevant considerations.



(a) State-space model (b) GP regression

Fig. 7. Comparison of time series forecasting of GP regression and state-space model.

X. CONCLUSION

In this paper we have considered the question of transforming popular state-space models (or structural time series models) into corresponding Gaussian Processes. The reverse transformation is studied in e.g. [12] and references there in. We have considered general Local Linear Trend Model (LLLM) and its simplifications, quasi-periodic (cyclic) state-space model, damped trend model. At first, these models are written in the continuous time forms and then corresponding GP kernels are derived. Other widely used models like ARMA, external variables and model combinations have been mentioned and the way to construct GP kernels for them have been shown.

We have demonstrated the correctness and feasibility of the GP regression with novel kernels on the several synthetic datasets and equivalence with state-space modeling is shown on a real world dataset.

Thus, this paper makes a bridge between state-space and GP modeling and forecasting of time series data. It allows experts in either of the fields to look at their models from the other point of view and share the ideas between those approaches of modeling.

REFERENCES

[1] A. Yaglom, *Correlation Theory of Stationary and Related Random Functions*. Springer, 1987.

[2] T. Durbin and S. Koopman, *Time Series Analysis by State Space Methods: Second Edition*, ser. Oxford Statistical Science Series. OUP Oxford, 2012.

[3] A. Harvey, *Forecasting, Structural Time Series Models and the Kalman Filter*. Cambridge University Press, 1990.

[4] H. Kantz and T. Schreiber, *Nonlinear Time Series Analysis*, ser. Cambridge nonlinear science series. Cambridge University Press, 2004.

[5] J. G. D. Gooijer and R. J. Hyndman, "25 years of time series forecasting," *International Journal of Forecasting*, pp. 443–473, 2006.

[6] G. Box, G. Jenkins, and G. Reinsel, *Time Series Analysis: Forecasting and Control*, ser. Wiley Series in Probability and Statistics. Wiley, 2008.

[7] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2005.

[8] S. Roberts, M. Osborne, M. Ebden, S. Reece, N. Gibson, and S. Aigrain, "Gaussian processes for time-series modelling," *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, vol. 371, no. 1984, 2012.

[9] J. Hartikainen and S. Särkkä, "Kalman filtering and smoothing solutions to temporal gaussian process regression models," in *Machine Learning for Signal Processing (MLSP), 2010 IEEE International Workshop on*, Aug 2010, pp. 379–384.

[10] S. Ambikasaran, D. Foreman-Mackey, L. Greengard, D. W. Hogg, and M. O'Neil, "Fast direct methods for gaussian processes," 2014.

[11] S. Särkkä, *Bayesian Filtering and Smoothing*. Cambrige University Press, 2013.
[12] A. Solin and S. Särkkä, "Explicit link between periodic covariance functions and state space models," in *Proceedings of the 17-th Int. Conf. on Artificial Intelligence and Statistics (AISTATS 2014)*, ser. JMLR Workshop and Conference Proceedings, vol. 33, 2014, pp. 904–912.
[13] S. Ihara, *Information Theory for Continuous Systems*. World Scientific, 1993.
[14] G. W. Cobb, *Biometrika*, vol. 65, no. 2, pp. 243–251, 1978.
[15] The GPy authors, "GPy: A gaussian process framework in python," http://github.com/SheffieldML/GPy, 2012–2015.