Aalto University
School of Science
Degree programme in Engineering Physics and Mathematics

State-Space Inference in Gaussian Process Regression Models

Bachelor's thesis
September 19, 2013

Jukka Koskenranta

Aalto University

AALTO UNIVERSITY SCHOOL OF SCIENCE PO Box 11000, FI-00076 AALTO http://www.aalto.fi	ABSTRACT OF THE BACHELOR'S THESIS

Author: Jukka Koskenranta

Title: State-Space Inference in Gaussian Process Regression Models

Degree programme: Degree programme in Engineering Physics and Mathematics

Major subject: Systems Sciences	Major subject code: Mat-2

Supervisor: Prof. Harri Ehtamo

Instructor: Dr.Tech. Simo Särkkä, BECS, Aalto University

Abstract:

This thesis is concerned with two methods: Gaussian process regression models and state-space models. Gaussian process regression models have become increasingly popular tools for many type of applications. One driving force is flexibility of Gaussian processes. In this thesis we investigate methods for reducing the computational complexity and apply them to two Gaussian process regression problems. The first case is a Gaussian process with a Matérn covariance function as the prior and the second case is a Gaussian process with a squared exponential covariance function as the prior.

The computational complexity of Gaussian process regression grows cubically with the number of measurements. We reduce the computation complexity to linear by converting Gaussian process regression problems into state-space models and further into Kalman filtering and smoothing problems. After the conversion, we use these state-space models for two different experiments. The first experiment is computed with generated data to show the results of the conversion. The second experiment is real data regression, with prediction, to show some capabilities of these methods.

Date: September 19, 2013	Language: English	Number of pages: 4+29

Keywords: state-space model, stochastic differential equation, Gaussian process regression model, nonparametric regression, Kalman filtering and smoothing

Aalto-yliopisto

AALTO-YLIOPISTO PERUSTIETEIDEN KORKEAKOULU PL 11000, 00076 Aalto http://www.aalto.fi	KANDIDAATINTYÖN TIIVISTELMÄ

Tekijä: Jukka Koskenranta

Työn nimi: Tila-avaruuspohjainen laskenta gaussisiin prosesseihin perustuvissa regressiomalleissa

Tutkinto-ohjelma: Teknillisen fysiikan ja matematiikan tutkinto-ohjelma

Pääaine: Systeemitieteet	Pääaineen koodi: Mat-2

Vastuuopettaja(t): Prof. Harri Ehtamo

Ohjaaja(t): TkT. Simo Särkkä, BECS, Aalto-yliopisto

Tiivistelmä:

Tämä työ keskittyy kahteen menetelmään: regressiomalleihin gaussisten prosessien avulla (GPR-malleihin) ja tila-avaruusmalleihin. GPR-mallit ovat viime aikoina lisääntyneet niiden helppouden ja hyvän sovellettavuuden takia. GPR-mallit ovat kuitenkin laskennallisesti raskaita. Niissä laskenta-aika kasvaa kuutiollisesti, kun laskentapisteitä lisätään. Tila-avaruusmallit vastaavat tähän ongelmaan, sillä niissä laskenta-aika kasvaa ainoastaan lineaarisesti suhteessa laskentapisteiden määrään.

Tämän työn päätavoitteena on muuntaa kaksi yleistä GPR-mallia vastaavaan tila-avaruusmuotoon. Tila-avaruusmuodot muunnetaan vielä edelleen Kalmanin suodatus- ja silotusongelmiksi, jotka ovat yleisiä ja tehokkaita ratkaisumenetelmiä tila-avaruusmalleille. Ensimmäinen koetilanne tehdään simuloidulla aineistolla, siinä saadaan hyvin esiin muunnoksen ominaisuudet. Toinen koetilanne tehdään aidolla aineistolla ja menetelmiä käytetään ennustamiseen. Jälkimmäinen koetilanne on monimutkaisempi ja näyttää samalla, minkälaisiin ongelmiin menetelmiä voi soveltaa.

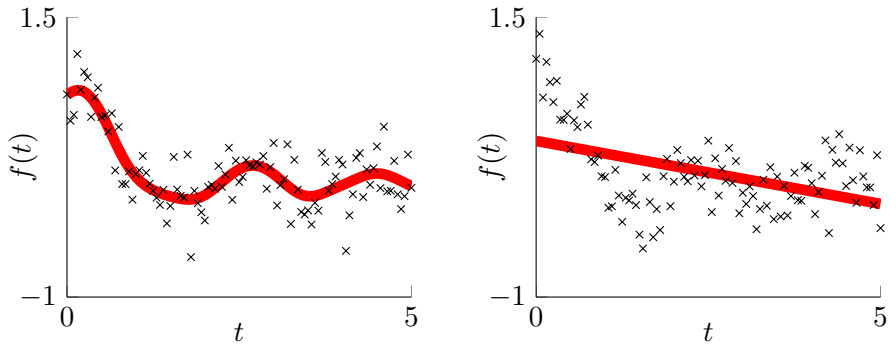
Päivämäärä: 19.9.2013	Kieli: Englanti	Sivumäärä: 4+29

Avainsanat: tila-avaruusmalli, stokastinen differentiaaliyhtälö, regressio gaussisten prosessien avulla, regressio ilman parametreja, Kalmanin silotus ja suodatus

Contents

1 Introduction

This thesis is concerned with Gaussian process regression models (Rasmussen and Williams, 2006) and state-space models. The idea is to combine the good sides of these models by converting the Gaussian process regression models into state-space models. During the recent decades Gaussian process regression models have become increasingly popular tools for many type of applications (see, e.g., Hartikainen, 2013). One driving force is flexibility of Gaussian processes. Figure 1 illustrates difference between Gaussian process regression and linear regression. It shows that Gaussian process regression models are flexible, for example, when data has nonlinear dependencies. This thesis is mainly based on the references: Särkkä et al. (2013), Hartikainen and Särkkä (2010), Särkkä and Hartikainen (2012) and Hartikainen (2013).



(a) Gaussian process regression (b) Linear regression

Figure 1: Example of Gaussian process regression and linear regression. Data is generated from sinc function with Gaussian noise. Red line represents mean of the regression results and black markings are the generated data.

One problem in Gaussian process regression is the computational complexity. Computation time grows cubically $\mathrm{O}(n^3)$ in the number of measurement n. This makes Gaussian process regression ineffective in large regression problems. As shown by Särkkä et al. (2013), one way to solve this problem is convert the Gaussian process into a state-space model. Some, but not all, Gaussian processes can be converted into state-space models and vice versa. When using a state-space model, computation time grows only linearly $\mathrm{O}(n)$ in the number of points, while solution is the same.

The main emphasis in this thesis is in converting two commonly used Gaussian process regression problems into state-space form. These converted Gaussian processes are used in two different regression experiments.

2 Models and methods

In the following sections, we describe more specific Gaussian process regression, Gaussian state-space models, conversion from Gaussian process regression problems into state-space models, and finally Kalman filtering and smoothing.

2.1 Gaussian processes

2.1.1 Gaussian process

A Gaussian process can be considered a generalization of multivariate Gaussian distribution to infinite dimensions (Rasmussen and Williams, 2006). A multivariate Gaussian distribution of finite dimension n is completely defined by its mean $\boldsymbol{\mu} \in \mathbb{R}^n$ and covariance $\boldsymbol{\Sigma} \in \mathbb{R}^{n \times n}$. If $\boldsymbol{z} = \{z_1, \cdots, z_n\}$ is multivariate Gaussian distributed it is denoted as

$$\boldsymbol{z} \sim N(\boldsymbol{\mu}, \boldsymbol{\Sigma}). \tag{1}$$

Difference between a Gaussian process and a multivariate Gaussian distribution (see, e.g., Rasmussen and Williams, 2006) is that Gaussian process is infinite dimensional. Because it is infinite dimensional, it has mean and covariance functions instead of a mean vector and covariance matrix. If $f(t)$ is Gaussian process, it is completely defined by its mean function $\mu(t)$ and covariance function $k(t, t')$, which is denoted as:

$$f(t) \sim GP(\mu(t), k(t, t')), \tag{2}$$

where the mean and covariance functions are defined as

$$\begin{aligned} \mu(t) &= \mathbb{E}[f(t)], \\ k(t, t') &= \mathbb{E}[(f(t) - \mu(t))(f(t') - \mu(t'))]. \end{aligned} \tag{3}$$

Usually, in Gaussian process regression, μ is assumed to be zero. It simplifies notation, but it is not necessary:

$$f(t) \sim GP(0, k(t, t')). \tag{4}$$

A Gaussian process can be formally defined as follows (Rasmussen and Williams, 2006).

Definition 2.1 (Gaussian process) *Gaussian process is a collection of random variables* $\{f(t) : t \in \mathbb{T}\}$ *such that any finite subset* $\{(x(t_1), \cdots, x(t_n)) : t_1, \cdots, t_n \in \mathbb{T}, n < \infty\}$ *is jointly Gaussian.*

2.1.2 Gaussian process regression

In regression, the idea is to model dependencies inside some given data. When we have the (estimated) model for these dependencies, we can predict values for any new point. These new points are usually called test points. For example, in linear regression we fit straight lines into (multidimensional) data. Then, we have a linear function for representing the dependencies. The parameters of the model are the linear regression coefficients.

Gaussian process regression (GPR) is non-parametric regression. Non-parametric means that we skip the estimation of parameters that we had in linear regression. Instead of estimating the parameters, we directly estimate values for the test points. This is closely related to so called kernel trick (Rasmussen and Williams, 2006). Because we do not form the parametric model, we can more freely choose the form of the dependencies. Only some GPR problems can be converted into equivalent parametric regression problems. In GPR, the dependencies of the data are encoded into the prior covariance function. Because of this prior covariance function, GPR still needs prior information. We have to choose shape and the hyperparameters of the prior covariance function. For the prior covariance function, an arbitrary function of the pair t and t' will not do, but the covariance function needs to be positive definite. Usually mean is assumed to be zero and the covariance function dependent only on $\tau \in \{t - t', |t - t'|, t \cdot t'\}$ (Rasmussen and Williams, 2006).

In addition to Equation (4), there might be noise in the measurements y_k. Thus we do not know the actual values even at the measurement points, but only noisy versions of them:

$$y_k = f(t_k) + \epsilon_k, \tag{5}$$

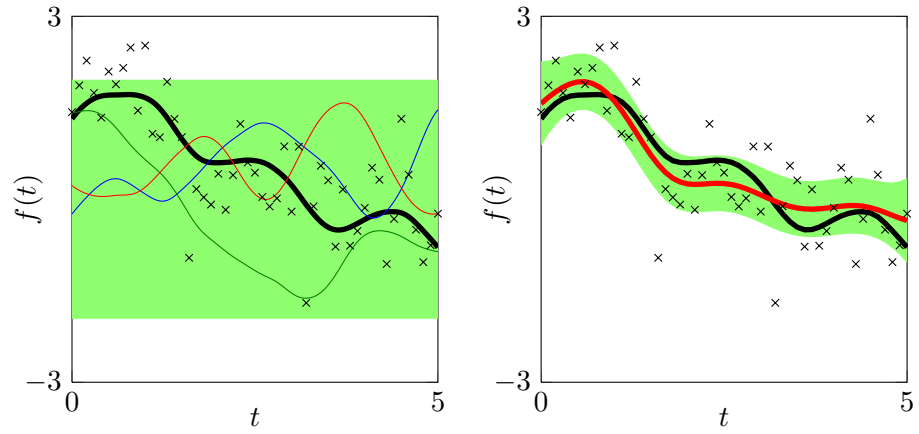
where ϵ_k is assumed to be independent and identically distributed Gaussian random variable with zero mean and variance σ_n^2.

The result of Gaussian process regression is the posterior distribution of the function f. Posterior can be evaluated for any test point t_* or a vector of test points $\boldsymbol{t}_*$. Finite dimensional posterior characterizes multivariate normal distribution with mean $\bar{\boldsymbol{f}}_*$ and covariance $\mathbb{V}(\boldsymbol{f}_*)$. The following equations can be used to compute posterior distribution, when prior mean is zero (see, e.g., Rasmussen and Williams, 2006):

$$\bar{\boldsymbol{f}}_* = k(\boldsymbol{t}, \boldsymbol{t}_*)^{\mathsf{T}}(k(\boldsymbol{t}, \boldsymbol{t}) + \sigma_n^2 \boldsymbol{I})^{-1}\boldsymbol{y}, \tag{6}$$

$$\mathbb{V}(\boldsymbol{f}_*) = k(\boldsymbol{t}_*, \boldsymbol{t}_*) - k(\boldsymbol{t}, \boldsymbol{t}_*)^{\mathsf{T}}(k(\boldsymbol{t}, \boldsymbol{t}) + \sigma_n^2 \boldsymbol{I})^{-1}k(\boldsymbol{t}, \boldsymbol{t}_*), \tag{7}$$

where $k(\boldsymbol{x}, \boldsymbol{x}')$ is a prior covariance function giving the covariance of points $\boldsymbol{x}$ and $\boldsymbol{x}'$, $\boldsymbol{t}_*$ is a vector of test points, $\boldsymbol{t}$ is a vector of measurement points, and σ_n^2 is measurement noise variance.



(a) Random draws from prior (b) Gaussian regression example

Figure 2: Example of Gaussian process regression. In both figures, bold black line represents the function realization used in regression and crosses are noisy measurements from that function. Figure 2a: Random realizations from prior distribution of regression, green area represents 95% confidence interval of prior. Figure 2b: A bold red line represents mean of regression result and green represents 95% confidence region of regression result.

2.1.3 Linear Gaussian state-space models

State-space models (see, e.g. Hartikainen, 2013) are a general framework for modeling dynamic systems. In dynamic system aim is to estimate states of the system, $\boldsymbol{f}(t) \in \mathbb{R}^n$, from given measurements $\boldsymbol{y}_k \in \mathbb{R}^m$. Measurements $y_1, ..., y_T$ are given at time steps $t_1, ..., t_T$. Measurements might be noisy and indirect functions of the states, and are typically modeled using a *measurement model* of the form

$$\boldsymbol{y}_k \sim p(\boldsymbol{y}_k | \boldsymbol{f}(t_k)), \tag{8}$$

where p is the probability density of measurements given the state.

State's dependence from previous states is expressed as an *ordinary differential equation* (ODE), which forms the *dynamic model*. This ODE might also involve noise, so ODE becomes *stochastic differential equation* (SDE) of the form (Särkkä, 2006)

$$\frac{\mathrm{d}\boldsymbol{f}(t)}{\mathrm{d}t} = \mathbf{a}(\boldsymbol{f}(t), t) + \mathbf{L}(t)\boldsymbol{w}(t). \tag{9}$$

where $\mathbf{a}(\boldsymbol{f}(t), t)$ and $\mathbf{L}(t)$ are some functions and $\boldsymbol{w}(t)$ is a vector of white noise processes. A *white noise process* refers to a zero mean Gaussian random

process, where each pair of values $\boldsymbol{w}(t)$ and $\boldsymbol{w}(t')$ are uncorrelated when $t \neq t'$.

The state-space model used in this thesis is as follows:

$$\begin{aligned} \frac{\mathrm{d}\boldsymbol{f}(t)}{\mathrm{d}t} &= \mathbf{F}\boldsymbol{f}(t) + \mathbf{L}w(t), \\ y_k &= \mathbf{H}\boldsymbol{f}(t_k) + \epsilon_k, \end{aligned} \tag{10}$$

where $k = 1, ..., T$ and $\mathbf{F}$, $\mathbf{L}$ and $\mathbf{H}$ are given matrices, $\boldsymbol{\epsilon}_k$ is a zero mean Gaussian measurement noise process, with constant covariance σ_n^2, and $w(t)$ is a scalar white noise process. The used state-space model in Equation (10), is *linear time-invariant* (LTI) and noises are Gaussian, scalar, and *independent and identically distributed* (IID).

2.1.4 Kalman filtering and smoothing

Kalman filter and *Rauch–Tung–Striebel smoother* (RTSS) (Kalman, 1960) (Rauch et al., 1965) (Särkkä, 2006) (Särkkä, 2013) produce effectively exact solutions for state inference in linear-Gaussian state-space models, described as,

$$\begin{aligned} \frac{\mathrm{d}\boldsymbol{f}(t)}{\mathrm{d}t} &= \mathbf{F}\boldsymbol{f}(t) + \mathbf{L}\boldsymbol{w}(t), \\ \boldsymbol{y}_k &= \mathbf{H}\boldsymbol{f}(t_k) + \boldsymbol{\epsilon}_k, \quad \boldsymbol{\epsilon}_k \sim N(\mathbf{0}, \boldsymbol{\sigma}_{n,k}^2), \end{aligned} \tag{11}$$

where $k = 1, ..., T$, $\mathbf{A}$, $\mathbf{L}$, and $\mathbf{H}$ are given matrices, $\boldsymbol{\epsilon}_k$ is a vector of zero mean Gaussian measurement noise processes with covariances $\boldsymbol{\sigma}_{n,k}^2$. Note that the used state-space model in Equation (10), is a special case of the model in Equation (11). Kalman filtering and smoothing solutions are optimal in sense of Bayesian filtering and smoothing (see, e.g. Särkkä, 2013) and they are recursive algorithms which can be used when state-space model is discretized.

In the LTI case, as in Equation (11), the discretized model matrices can be efficiently solved as a function of the time step size $\Delta t_k = t_{k+1} - t_k$ as (Särkkä, 2006)

$$\begin{aligned} \mathbf{A}(\Delta t_k) &= \mathbf{A}_k = \Phi(\Delta t_k), \\ \mathbf{Q}(\Delta t_k) &= \mathbf{Q}_k = \int_0^{\Delta t_k} \Phi(\Delta t_k - \tau)\mathbf{L}\mathbf{Q}_c\mathbf{L}^\mathsf{T}\Phi(\Delta t_k - \tau)^\mathsf{T}\mathrm{d}\tau, \end{aligned} \tag{12}$$

where $\Phi(\tau)$ denotes the matrix exponential, $\Phi(\tau) = \exp(\mathbf{F}\tau)$. The used dynamic model in Equation (10) is a special case of LTI dynamic model in Equation (11) when white noise process is scalar, $\boldsymbol{w}(t) = w(t)$. With

notation from Equation (12), discretized solution to model in Equation (10) is as follows:

$$\begin{aligned} \boldsymbol{f}(t_{k+1}) &= \mathbf{A}_k \boldsymbol{f}(t_k) + \boldsymbol{q}_k, \quad \boldsymbol{q}_k \sim N(\mathbf{0}, \mathbf{Q}_k), \\ y_k &= \mathbf{H}\boldsymbol{f}(t_k) + \epsilon_k, \quad \epsilon_k \sim N(0, \sigma^2_{n,k}). \end{aligned} \tag{13}$$

Then this discretized state space model in Equation (13) can be solved with the Kalman filtering and smoothing equations. Kalman filtering equations are as follows (originally derived by Kalman, 1960, notation from Solin, 2012):

- The prediction step is

$$\begin{aligned} \mathbf{m}_{k|k-1} &= \mathbf{A}_{k-1}\mathbf{m}_{k-1|k-1} \\ \mathbf{P}_{k|k-1} &= \mathbf{A}_{k-1}\mathbf{P}_{k-1|k-1}\mathbf{A}_{k-1}^{\mathsf{T}} + \mathbf{Q}_{k-1} \end{aligned} \tag{14}$$

- The update step is

$$\begin{aligned} \mathbf{v}_k &= \boldsymbol{y}_k - \mathbf{H}_k\mathbf{m}_{k|k-1} \\ \mathbf{S}_k &= \mathbf{H}_k\mathbf{P}_{k|k-1}\mathbf{H}_k^{\mathsf{T}} + \mathbf{R}_k \\ \mathbf{K}_k &= \mathbf{P}_{k|k-1}\mathbf{H}_k^{\mathsf{T}}\mathbf{S}_k^{-1} \\ \mathbf{m}_{k|k} &= \mathbf{m}_{k|k-1} + \mathbf{K}_k\mathbf{v}_k \\ \mathbf{P}_{k|k} &= \mathbf{P}_{k|k-1} - \mathbf{K}_k\mathbf{S}_k\mathbf{K}_k^{\mathsf{T}}, \end{aligned} \tag{15}$$

where $\mathbf{m}_{k|k-1}$ and $\mathbf{P}_{k|k-1}$ are the predicted mean and covariance of $\boldsymbol{f}(t_k)$, $\mathbf{m}_{k|k}$ and $\mathbf{P}_{k|k}$ are the updated mean and covariance of $\boldsymbol{f}(t_k)$ after update by measurement $\boldsymbol{y}_k$. $\mathbf{v}_k$, $\mathbf{S}_k$ and $\mathbf{K}_k$ are simplifying variables. $\mathbf{A}_k$, $\mathbf{Q}_k$ and $\mathbf{H}_k$ are from Equations (12) and (13). $\mathbf{R}_k$ is measurement noise covariance, in our case, Equation (10), denoted as $\sigma^2_{n,k}$ and it is positive constant. Initial value for mean, $\mathbf{m}_{0|0}$ and covariance $\mathbf{P}_{0|0}$ depends on a case.

In our case initial mean is zero and initial covariance is a solution to matrix Riccati equation (see, e.g. Hartikainen, 2013),

$$\frac{\mathrm{d}\mathbf{P}}{\mathrm{d}t} = \mathbf{F}\mathbf{P} + \mathbf{F}\mathbf{P}^{\mathsf{T}} + \mathbf{L}\mathbf{Q}_c\mathbf{L}^{\mathsf{T}} = \mathbf{0}. \tag{16}$$

Initial covariance is from Equation (16), because we use only stationary prior covariance functions. A *stationary covariance* refers to a covariance function $k(\tau')$ which is only dependent on $\tau' = t - t'$, instead of both t and t'.

Kalman filter equations are computed forwards, prediction step for all points and update step only when there is data for that point. After Kalman filtering equations we compute RTS smoother equations. These are computed

backwards for all points. RTS smoother equations are as follows (originally derived by Rauch et al., 1965, notation from Solin, 2012),

$$\begin{aligned}
\mathbf{m}_{k+1|k} &= \mathbf{A}_k \mathbf{m}_{k|k} \\
\mathbf{P}_{k+1|k} &= \mathbf{A}_k \mathbf{P}_{k|k} \mathbf{A}_k^\mathsf{T} + \mathbf{Q}_k \\
\mathbf{G}_k &= \mathbf{P}_{k|k} \mathbf{A}_k^\mathsf{T} [\mathbf{P}_{k+1|k}]^{-1} \\
\mathbf{m}_{k|T} &= \mathbf{m}_{k|k} + \mathbf{G}_k [\mathbf{m}_{k+1|T} - \mathbf{m}_{k+1|k}] \\
\mathbf{P}_{k|T} &= \mathbf{P}_{k|k} + \mathbf{G}_k [\mathbf{P}_{k+1|T} - \mathbf{P}_{k+1|k}] \mathbf{G}_k^\mathsf{T},
\end{aligned} \tag{17}$$

where the first two equations are the Kalman filtering prediction step, $\mathbf{m}_{k|T}$ and $\mathbf{P}_{k|T}$ are RTSS solutions to mean and covariance of $\boldsymbol{f}(t_k)$ and $\mathbf{G}_k$ is a simplifying variable. The initial step of RTSS, $\mathbf{m}_{T|T}$ and $\mathbf{P}_{T|T}$, is the same as solution to last step of Kalman filter, $\mathbf{m}_{T|T}$ and $\mathbf{P}_{T|T}$.

2.2 Converting Gaussian process regression into Kalman filtering and smoothing problem

In the following subsections, we first explain how a linear time-invariant Gaussian process can be converted to a state-space form and then we work out the conversion of two specific Gaussian processes. In Results, Section 3, we use these converted Gaussian processes in experiments. Some Gaussian processes can be represented equivalently in state-space form, some can be only approximated, and some cannot be converted to state-space form at all. In the experiments, the first Gaussian process represents one which can be converted to state-space form without approximation and the second one represents a class which has to be approximated.

The first Gaussian process has a Matérn covariance function with smoothness parameter $\nu = 7/2$. In the experiments, other parameters are $\ell = 1$ and $\sigma^2 = 1$. Matérn covariance functions, with finite and half-integer value for smoothness parameter ν, are class of Gaussian processes, which can be converted without approximation to state-space form. In the Experiment 1, we show how the regression result does not differ but computation time differ.

The second Gaussian process has a *squared exponential* (SE) covariance function. It is actually the same as Matérn covariance function with infinite smoothness parameter $\nu \to \infty$. When the smoothness parameter has higher value than $5/2$ it is hard to see the difference between SE covariance function and Matérn covariance function. In the Experiment 2, Gaussian process with SE covariance function is used for real data regression.

2.2.1 Linear time-invariant SDEs and Gaussian processes

One way to convert Gaussian process to state-space model is to approximate Gaussian process as a solution to nth order linear SDE (Särkkä et al., 2013),

$$\frac{\mathrm{d}^n f(t)}{\mathrm{d}t^n} + a_{n-1}\frac{\mathrm{d}^{n-1} f(t)}{\mathrm{d}t^{n-1}} + ... + a_1\frac{\mathrm{d}f(t)}{\mathrm{d}t} + a_0 f(t) = w(t), \tag{18}$$

where $w(t)$ is a zero-mean continuous time Gaussian white noise process. The solution process $f(t)$, random function, is a Gaussian process because $w(t)$ is Gaussian process and solution of a linear differential equation is a linear operation on the input. SDE in Equation (18) can equivalently be represented in the following state space form. If we define $\boldsymbol{f} = (f, \mathrm{d}f/\mathrm{d}t, ..., \mathrm{d}^{n-1}f/\mathrm{d}t^{n-1})$, then we have

$$\frac{\mathrm{d}\boldsymbol{f}(t)}{\mathrm{d}t} = \underbrace{\begin{pmatrix} 0 & 1 & & \\ & \ddots & \ddots & \\ & & 0 & 1 \\ -a_0 & -a_1 & \cdots & -a_{n-1} \end{pmatrix}}_{\mathbf{F}} \boldsymbol{f}(t) + \underbrace{\begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix}}_{\mathbf{L}} w(t). \tag{19}$$

Notice that the scalar function $f(t)$ is just the first component of the vector $\boldsymbol{f}(t)$. Thus if we assume that we measure noise corrupted values y_k of $f(t_k)$ at points $t_1, ..., t_N$, we can write this as

$$y_k = \underbrace{\begin{pmatrix} 1 & 0 & \cdots & 0 \end{pmatrix}}_{\mathbf{H}} \boldsymbol{f}(t) + \epsilon_k, \tag{20}$$

which indeed is a model of the form Equation (10).

We still need to get the values $a_0, ..., a_{n-1}$. These can be found by finding the bond between Gaussian process representation in Equation (18) and the covariance function of the same Gaussian process. If we take the formal Fourier transform of the Equation (18) and solve for the Fourier transform of the process, $F(i\omega)$, we get

$$F(i\omega) = \underbrace{\left(\frac{1}{(i\omega)^n + a_{n-1}(i\omega)^{n-1} + ... + a_1(i\omega) + a_0}\right)}_{G(i\omega)} W(i\omega), \tag{21}$$

where $W(i\omega)$ is the (formal) Fourier transform of the white noise process $w(t)$. The above equation can be interpreted such that the process $F(i\omega)$ is obtained by feeding white noise through a system with the transfer function $G(i\omega)$.

From the above description it is now easy to compute the corresponding spectral density of the process, which is just the square of the absolute value

of the Fourier transform of the process. If we denote the spectral density of the white noise $|W(i\omega)|^2 = Q_c$, the spectral density of the process is

$$S(\omega) = G(i\omega)Q_c G(-i\omega). \tag{22}$$

Then the classical Wiener-Khinchin theorem states that the stationary covariance function $k(t)$ of the process is given by the inverse Fourier transform of the spectral density (Särkkä et al., 2013):

$$k(t) = \mathcal{F}^{-1}[S(\omega)] = \frac{1}{2\pi}\int S(\omega)\exp(i\omega t)\mathrm{d}t. \tag{23}$$

Now we know how to form state space model as in Equation (19) from a Gaussian process:

1. Solve $S(\omega)$ from Equation (23).
2. Solve $G(i\omega)$ and Q_c from Equation (22). There are many solutions for $G(i\omega)$. It has to be chosen so that all of its poles are in left side of the imaginary plane. Pole means a root of the denominator in variable iw.
3. Find right values to the matrix $\mathbf{F}$ from $G(i\omega)$. The matrices $\mathbf{L}$ and $\mathbf{H}$ are always have the same form, but dimensions might differ. They can be found from Equations (19) and (20).

If neither step 1 or 2 can be solved exactly, they can be approximated with finite-dimensional polynomials. There are still problems, and as said earlier, all Gaussian processes cannot be converted into state-space models. For example, Fourier transform, as an integral transform, might not converge. The second example shows that SE covariance function cannot be converted exactly without additional approximations.

2.2.2 Gaussian processes with Matérn covariance functions

Gaussian processes with Matérn covariance functions are widely used Gaussian processes. There are general algorithms for converting this class of Gaussian processes to state-space form (Särkkä et al., 2013). In this thesis we convert one special case of this class to state-space form. In our case smoothness parameter $\nu = 7/2$. There are two other parameters and in this case, the solution has a closed form.

The conversion begins with covariance function. The general form of Matérn covariance function is (Rasmussen and Williams, 2006)

$$k_{\mathrm{Matern}}(\tau) = \sigma^2\frac{2^{1-\nu}}{\Gamma(\nu)}\left(\frac{\sqrt{2\nu}\tau}{\ell}\right)^{\nu} K_\nu\frac{\sqrt{2\nu}\tau}{\ell}, \tag{24}$$

with positive parameters σ^2, ν and ℓ, where K_ν is a modified Bessel function (Abramowitz and Stegun, 1965). Covariance function in Equation (24) simplifies a lot when ν is half-integer: $\nu = p+1/2$, where p is a non-negative integer. We can derive the general expression (see, e.g., Abramowitz and Stegun, 1965), giving

$$k_{\text{Matern}}^{\nu=p+1/2}(\tau) = \exp\left(-\frac{\sqrt{2\nu}\tau}{\ell}\right)\frac{\Gamma(p+1)}{\Gamma(2p+1)}\sum_{i=0}^{p}\frac{(p+i)!}{i!(p-i)!}\left(\frac{\sqrt{8\nu}\tau}{\ell}\right)^{p-i}. \quad (25)$$

From Equation (25) we can derive simple covariance functions for given ν. In this case we have $\nu = 7/5$, then we get

$$k_{\text{Matern72}}(\tau) = \sigma^2\left(1+\frac{\sqrt{7}\tau}{\ell}+\frac{14\tau^2}{5\ell^2}+\frac{7\sqrt{7}\tau^3}{15\ell^3}\right)\exp\left(-\frac{\sqrt{7}\tau}{\ell}\right). \quad (26)$$

By using methods introduced earlier, we can derive the state space model, Equation (13), for this Gaussian process. Spectral density can be derived with Fourier transform, inverse to Equation (23), giving

$$S_{\text{Matern72}}(\omega) = \sigma^2\frac{10976\ell\sqrt{7}}{5\left(7+\ell^2\omega^2\right)^4}. \quad (27)$$

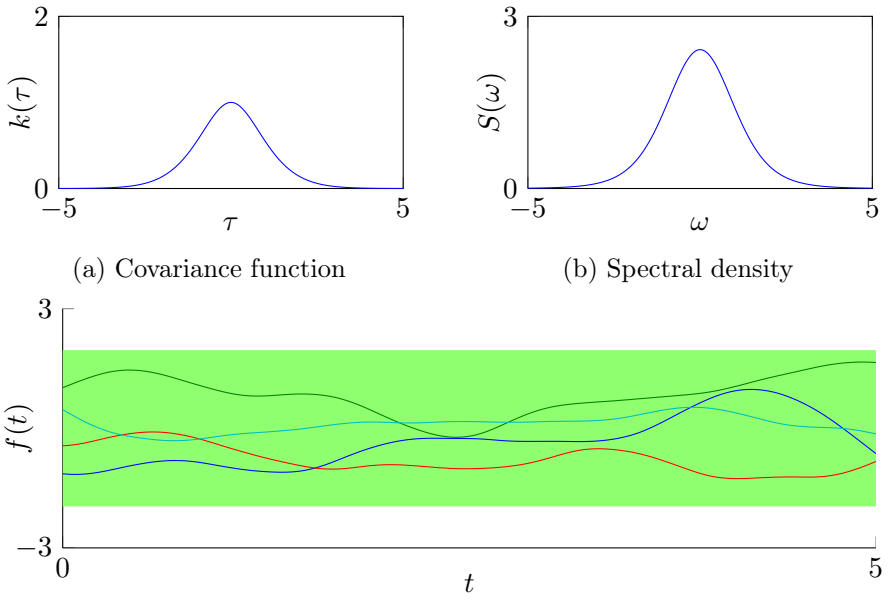
Reordering spectral density in Equation (27) we can derive the spectral density of the white noise Q_c and the transfer function $G(i\omega)$,

$$\begin{aligned} G(i\omega) &= \frac{1}{\left(\frac{\sqrt{7}}{\ell}\right)^4+4\left(\frac{\sqrt{7}}{\ell}\right)^3 i\omega+6\left(\frac{\sqrt{7}}{\ell}\right)^2(i\omega)^2+4\left(\frac{\sqrt{7}}{\ell}\right)(i\omega)^3+(i\omega)^4}, \\ Q_c &= \frac{\sigma^2 10976\sqrt{7}}{5\ell^7}, \end{aligned} \quad (28)$$

and from the transfer function we can find values for the matrix $\mathbf{F}$, $a_0 = \left(\sqrt{7}/\ell\right)^4$, $a_1 = 4\left(\sqrt{7}/\ell\right)^3$, $a_2 = 6\left(\sqrt{7}/\ell\right)^2$ and $a_3 = 4\sqrt{7}/\ell$. Matrices $\mathbf{L}$ and $\mathbf{H}$ are the same as in Equations (19) and (20). With this notation we can derive state-space model matrices,

$$\begin{aligned} \mathbf{F} &= \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -\left(\frac{\sqrt{7}}{\ell}\right)^4 & -4\left(\frac{\sqrt{7}}{\ell}\right)^3 & -6\left(\frac{\sqrt{7}}{\ell}\right)^2 & -4\left(\frac{\sqrt{7}}{\ell}\right) \end{pmatrix} \\ \mathbf{L} &= \begin{pmatrix} 0 & 0 & 0 & 1 \end{pmatrix}^{\mathsf{T}} \\ \mathbf{H} &= \begin{pmatrix} 1 & 0 & 0 & 0 \end{pmatrix}. \end{aligned} \quad (29)$$

Figure 3 illustrates a Gaussian process with Matérn covariance function, with parameters $\nu = 7/2$, $\ell = 1$, and $\sigma^2 = 1$. We want to represent the Gaussian



(a) Covariance function

(b) Spectral density

(c) Random draws from prior

Figure 3: Figures of Gaussian process with Matérn covariance function and parameters $\nu = 7/2$, $\ell = 1$ and $\sigma^2 = 1$. Green in Figure 3c represents 95% confidence interval of prior.

process as a state space model. To convert the Gaussian process, we need to know the covariance function, Figure 3a, and the spectral density, Figure 3b. Figure 3c shows random draws from this Gaussian process and green area represents 95% confidence region.

2.2.3 Gaussian processes with squared exponential covariance functions

Gaussian processes with squared exponential covariance functions are another widely used Gaussian processes in GP regression (Rasmussen and Williams, 2006). State-space form of the squared exponential covariance function is not so simple; the exact solution is infinite dimensional, so it has to be approximated (Hartikainen, 2013, Hartikainen and Särkkä, 2010). There are also general state-space solution to squared exponential covariance function, Hartikainen (2013), but there is needed numerical computations. In this chapter, we derive a state-space solution to squared exponential covariance function with approximation degree $N = 2$. There are two parameters and in this case, the solution has a closed form. Same methods can be used for higher approximation degrees.

Squared exponential covariance function is as follows (Rasmussen and Williams, 2006):

$$k_{\mathrm{SE}}(\tau) = \sigma^2 \exp(-\tau^2/\ell^2). \tag{30}$$

Transforming of the covariance function to state-space form begins with computing the spectral density. It can be derived with Fourier transform, inverse to Equation (23):

$$S_{\mathrm{SE}}(\omega) = \sigma^2\sqrt{2\pi}\ell \exp\left(-\ell^2\omega^2\right), \tag{31}$$

and this can be approximated by the Taylor series,

$$\begin{aligned} S_{\mathrm{SE}}(\omega) &\approx \frac{\sigma^2\sqrt{2\pi}\ell}{1 + \frac{\ell^2}{2}\omega^2 + \cdots + \frac{1}{N!2^N}\ell^{2N}\omega^{2N}} \quad |N = 2 \\ &= \underbrace{\frac{8\sigma^2\sqrt{2\pi}}{\ell^3}}_{Q_c} \underbrace{\left(\frac{1}{\frac{8}{\ell^4} + \frac{4}{\ell^2}\omega^2 + \omega^4}\right)}_{|G(i\omega)|^2}, \end{aligned} \tag{32}$$

where Q_c is given above. $G(i\omega)$ can be solved as follows:

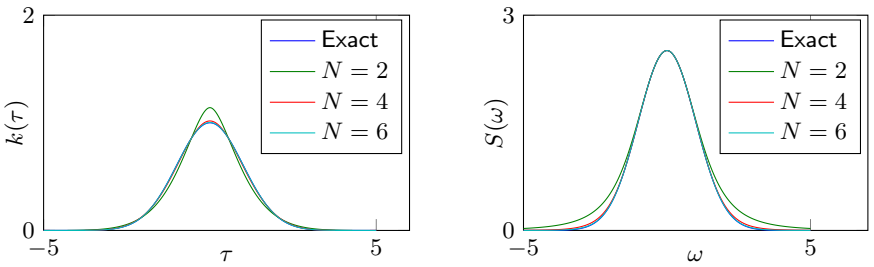
1. Find the poles of $|G(i\omega)|^2$. In this case the poles are:

$$\begin{aligned} \omega &= \frac{\pm\sqrt{-2 \pm 2i}}{\ell} \\ i\omega &= \frac{\pm\sqrt{2 \pm 2i}}{\ell} \\ i\omega &\approx \frac{\pm 1.55377 \pm 0.643594i}{\ell} \end{aligned}$$

   In general, when N is unknown, poles can be found numerically for every ℓ.

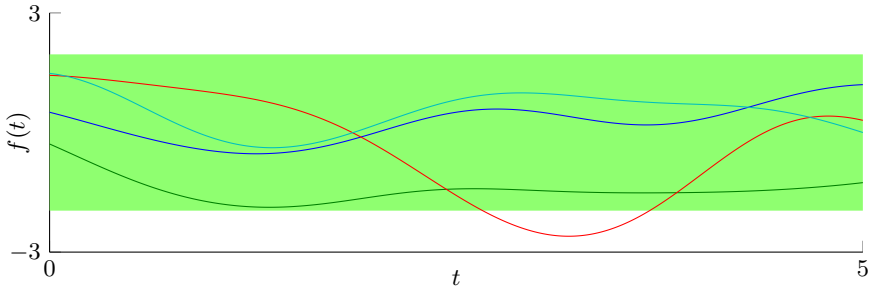
2. There should be poles in pairs, which are complex conjugates. Take the poles from left half of imaginary plane and construct a polynomial from these. In this case:

$$\begin{aligned} i\omega &\approx \frac{-1.55377 - 0.643594i}{\ell}, \quad i\omega \approx \frac{-1.55377 + 0.643594i}{\ell} \\ G(i\omega) &\approx \frac{1}{\left(i\omega - \frac{-1.55377-0.643594i}{\ell}\right)\left(i\omega - \frac{-1.55377+0.643594i}{\ell}\right)} \\ &= \frac{1}{\frac{2.82841}{\ell^2} + \frac{3.10754}{\ell}(i\omega) + (i\omega)^2}. \end{aligned}$$



(a) Covariance function

(b) Spectral density



(c) Random draws from prior

Figure 4: Figures of Gaussian process with squared exponential covariance function, with parameters $\ell = 1$ and $\sigma^2 = 1$. In this case, state-space model is only approximation. Exact values are from Gaussian form and approximations are from state space model, with different approximation degrees N. Green in Figure 4c represents 95% confidence interval of prior.

We now have $G(i\omega)$ and from it we can find values for the matrix $\mathbf{F}$: $a_0 = \frac{2.82841}{\ell^2}$ and $a_1 = \frac{3.10754}{\ell}$. Then we need to form matrices $\mathbf{L}$ and $\mathbf{H}$. These matrices are the same as in Equations (19) and (20). Thus we get the state-space model matrices:

$$\begin{aligned} \mathbf{F} &= \begin{pmatrix} 0 & 1 \\ -\frac{2.82841}{\ell^2} & -\frac{3.10754}{\ell} \end{pmatrix}, \\ \mathbf{L} &= \begin{pmatrix} 0 & 1 \end{pmatrix}^{\mathsf{T}}, \\ \mathbf{H} &= \begin{pmatrix} 1 & 0 \end{pmatrix}. \end{aligned} \tag{33}$$

Figure 4 illustrates a Gaussian process with a squared exponential covariance function, with parameters $\ell = 1$ and $\sigma^2 = 1$. We want to represent the Gaussian process as a state space model. To convert the Gaussian process, we need to know the covariance function, as in Figure 4a, and the spectral density, as in Figure 4b. Figure 4c shows random draws from this Gaussian process and green area represents 95% confidence region.

2.2.4 Hyperparameter optimization

Usually the prior covariance function has some unknown hyperparameters. One way to choose the parameters is to minimize energy function $\varphi(\boldsymbol{\theta})$ – *the negative logarithm of the posterior distribution* – with respect to hyperparameters. For example, in Matérn covariance function, there are three hyperparameters, ν, σ^2 and ℓ. This kind of optimization refers to a *maximum a posteriori* (MAP) estimate. In our case, when we have a uniform prior, this optimization gives the *maximum likelihood* (ML) estimate. When using a linear state-space model, as in Equation (11), the recursive algorithm for computing energy function $\varphi_T(\boldsymbol{\theta})$ for given parameters $\boldsymbol{\theta}$, is as follows (Särkkä, 2013):

$$\varphi_k(\boldsymbol{\theta}) = \varphi_{k-1}(\boldsymbol{\theta}) + \frac{1}{2}\log|2\pi \boldsymbol{S}_k(\boldsymbol{\theta})| + \frac{1}{2}\boldsymbol{v}_k^{\mathsf{T}}(\boldsymbol{\theta})\boldsymbol{S}_k^{-1}(\boldsymbol{\theta})\boldsymbol{v}_k(\boldsymbol{\theta}), \qquad (34)$$

where $\boldsymbol{S}_k(\boldsymbol{\theta})$ and $\boldsymbol{v}_k(\boldsymbol{\theta})$ can be found from Kalman filtering Equations (14) and (15), for given hyperparameters $\boldsymbol{\theta}$. Initial value $\varphi_0(\boldsymbol{\theta}) = 0$. Once we have the algorithm for computing energy function, we can optimize it. For example, MATLAB® has many optimization algorithms which can be used (MathWorks, 2013a). We could also use Markov chain Monte Carlo methods for generating Monte Carlo approximations for the posterior distribution (Särkkä, 2013).

If we have discrete state-space model, as in Equation (13), it is straight forward to use energy function (Särkkä, 2013). When we have a prior covariance function for a Gaussian process regression, it needs many steps before we can compute the energy function. The steps are listed here:

1. Form continuous LTI model, as in Section 2.2.3 or 2.2.3. If there exists multiple models, form a superposition model of the models, as described in Equations (35) and (36) in Section 2.3.1.
2. Evaluate a discretized solution to a continuous LTI model, with a discretization step $\Delta t = t_{k+1} - t_k$ in Equation (12).
3. Compute the Kalman filter, Equations (14) and (15), and the energy function iteration step in Equation (34). Initial values to mean and energy are zero. Initial value to covariance can be computed from Equation (16).
4. If there exist more measurement points, go back to step 2. If step size has not changed, you can go straight back to step 3. The evaluation of energy function uses only measurement points, it is independent from test points.

2.3 Additional methods for state-space models

In this section, we shortly introduce two additional methods for state-space models that are used in Experiment 2.

2.3.1 Superposition of multiple state-space models

With state-space models, it is easy to make superposition of multiple models. It allows us to use more complicated models. Here we show how superposition model can be constructed. Define two state-space models as follows:

$$\begin{aligned}
\frac{\mathrm{d}\boldsymbol{f}_1(t)}{\mathrm{d}t} &= \mathbf{F}_1\boldsymbol{f}_1(t) + \mathbf{L}_1\boldsymbol{w}_1(t),\\
y_{k,1} &= \mathbf{H}_1\boldsymbol{f}_1(t_k) + \epsilon_{k,1},\\
\frac{\mathrm{d}\boldsymbol{f}_2(t)}{\mathrm{d}t} &= \mathbf{F}_2\boldsymbol{f}_2(t) + \mathbf{L}_2\boldsymbol{w}_2(t),\\
y_{k,2} &= \mathbf{H}_2\boldsymbol{f}_2(t_k) + \epsilon_{k,2},
\end{aligned} \tag{35}$$

where $\boldsymbol{w}_1(t)$ and $\boldsymbol{w}_2(t)$ are Gaussian white noise processes with spectral densities $\mathbf{Q}_{c,1}$ and $\mathbf{Q}_{c,2}$. Derive superposition of the models as follows:

$$\begin{aligned}
y_k = y_{k,1} + y_{k,2} &= \underbrace{\begin{pmatrix}\mathbf{H}_1 & \mathbf{H}_2\end{pmatrix}}_{\mathbf{H}}\underbrace{\begin{pmatrix}\boldsymbol{f}_1(t_k)\\ \boldsymbol{f}_2(t_k)\end{pmatrix}}_{\boldsymbol{f}(t_k)} + \underbrace{(\epsilon_{k,1} + \epsilon_{k,2})}_{\epsilon_k},\\
\frac{\mathrm{d}\boldsymbol{f}(t)}{\mathrm{d}t} &= \begin{pmatrix}\frac{\mathrm{d}\boldsymbol{f}_1(t)}{\mathrm{d}t}\\ \frac{\mathrm{d}\boldsymbol{f}_2(t)}{\mathrm{d}t}\end{pmatrix} = \begin{pmatrix}\mathbf{F}_1\boldsymbol{f}_1(t)\\ \mathbf{F}_2\boldsymbol{f}_2(t)\end{pmatrix} + \begin{pmatrix}\mathbf{L}_1\boldsymbol{w}_1(t)\\ \mathbf{L}_2\boldsymbol{w}_2(t)\end{pmatrix}\\
&= \underbrace{\begin{pmatrix}\mathbf{F}_1 & \mathbf{0}\\ \mathbf{0} & \mathbf{F}_2\end{pmatrix}}_{\mathbf{F}}\underbrace{\begin{pmatrix}\boldsymbol{f}_1(t)\\ \boldsymbol{f}_2(t)\end{pmatrix}}_{\boldsymbol{f}(t)} + \underbrace{\begin{pmatrix}\mathbf{L}_1 & \mathbf{0}\\ \mathbf{0} & \mathbf{L}_2\end{pmatrix}}_{\mathbf{L}}\underbrace{\begin{pmatrix}\boldsymbol{w}_1(t)\\ \boldsymbol{w}_2(t)\end{pmatrix}}_{\boldsymbol{w}(t)},\\
\mathbf{Q}_c &= \begin{pmatrix}\mathbf{Q}_{c,1} & \mathbf{0}\\ \mathbf{0} & \mathbf{Q}_{c,2}\end{pmatrix},
\end{aligned} \tag{36}$$

where $\boldsymbol{w}(t)$ is a Gaussian white noise process with a spectral density $\mathbf{Q}_c$. If we need a third model, we can sum it as earlier to the superposition model, Equation (36), and we can continue this as long as we need to.

2.3.2 Stochastic resonator model

A benefit of conversion from Gaussian process regression models to state-space model is that we can easily use these Gaussian processes with other state-space models. For example, many physical models are state-space models. In the second experiment we use a resonator model. A resonator model

is denoted as follows, in terms of state-space model, as in Equation (10) (Särkkä et al., 2012):

$$\begin{aligned} \mathbf{F} &= \begin{pmatrix} 0 & 2\pi f \\ -2\pi f & -\zeta \end{pmatrix}, \\ \mathbf{L} &= \begin{pmatrix} 0 & 1 \end{pmatrix}^{\mathsf{T}}, \\ \mathbf{H} &= \begin{pmatrix} 1 & 0 \end{pmatrix}, \end{aligned} \tag{37}$$

where $f \in \mathbb{R}$ is frequency and $\zeta \geq 0$ is damping. In this case, $Q_c \geq 0$ is an independent parameter, which does not effect the model matrices in Equation (37).

3 Results

In the following subsections, we use methods of this thesis in two different experiments.

3.1 Experiment 1

This experiment demonstrates the equivalence of regression results in the original Gaussian process regression and regression after the conversion to state space form. It also shows the difference in computation times with increasing number of training points. When the number points grows, computation time in original Gaussian process regression should grow cubically and computation time in state-space model should grow linearly.

The Gaussian process has zero mean and Matérn covariance function with parameters $\nu = 7/2$, $\ell = 1$, and $\sigma^2 = 1$. The regression results should also be the same with both the approaches when using a Matérn covariance function with ν as a finite half-integer, but that is not proved in this thesis.

At first, random draws are generated from the Gaussian process. From the Gaussian process model, it can be done by generating values from a multivariate Gaussian distribution. The covariance matrix used in the multivariate distribution is evaluated from the covariance function and the mean is zero. For example, MATLAB® has function "randn" to generate multivariate Gaussian distributed values (MathWorks, 2013b). Then we add some Gaussian noise to that random draw, so that we have noisy measurements for the regression. This Gaussian noise can also be generated with this "randn" function from MATLAB®. In this thesis, measurement noise is expected to be IID and zero-mean.

We compute Gaussian process regression with Equations (6) and (7). These equations give mean and covariance of estimation. Deriving state-space solution has more steps. First we convert covariance of Gaussian process to continuous LTI state-space model, we do this with Equations (28) and (29). We discretize it with Equation (12). We solve the needed stationary covariance from Equation (16). From the discretized solution we can get the regression result with Kalman filter and RTS smoother in Equations (14) to (17). Note that Kalman filter's update step in Equation (15) is executed only when there exist data and all the other Kalman filtering and smoothing equations are executed for all the test points. The discretized solution depends on step size, so if test points has different step sizes, discretized solution has to be computed for all test points.

Figure 5 shows differences between computation times. We generated data as described earlier, for 5000 test points. Then we computed the regression with

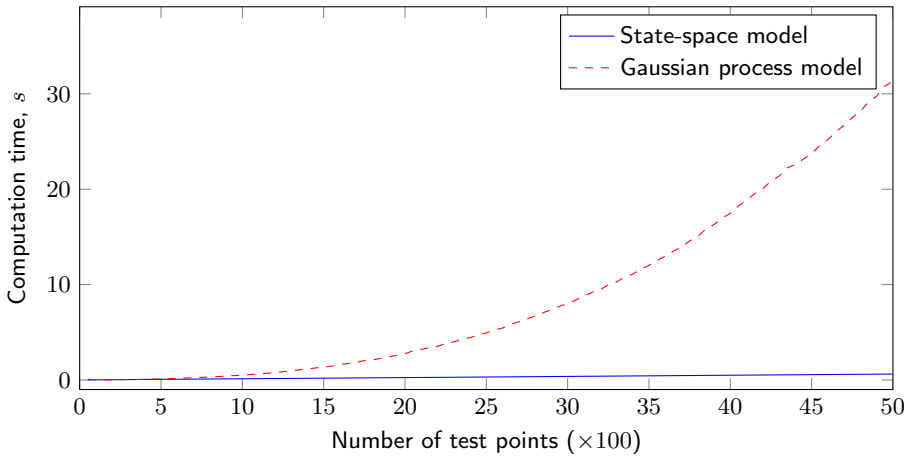
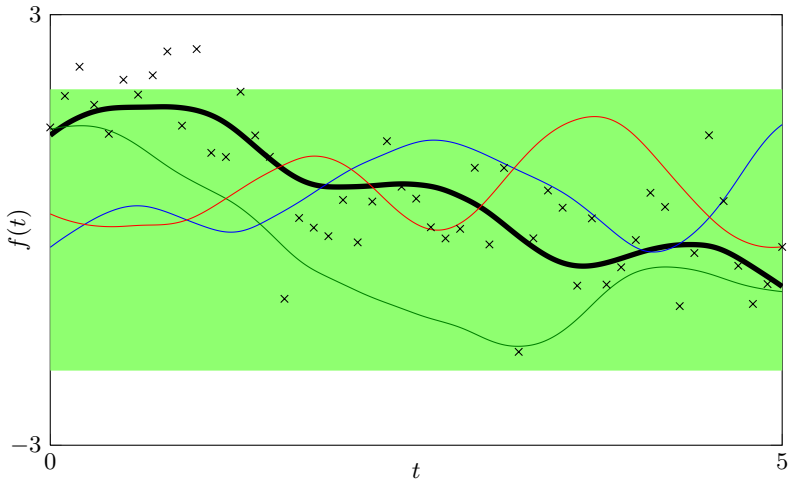


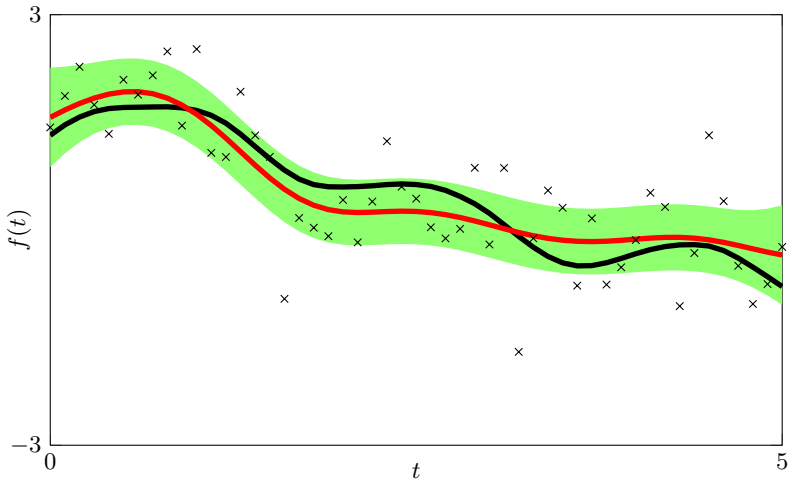
Figure 5: Comparing computation time of the ordinary Gaussian process regression with computation time of the same regression when using state-space model.

the same data, first only for 100 points, then 200 points and so on, until the last regressions were computed with the whole 5000 points. We smoothed the computation times taking average of 30 repeated regressions. Figure 5 shows that computation time grows cubically when using the original Gaussian process regression model and linearly when using the state-space model.

Figure 6a shows random draws form prior and 95% confidence region. It shows that the random draws are almost inside the 95% confidence region. Figure 6b shows the Gaussian process regression mean and 95% confidence region. We computed the regression for one of the random draws added some Gaussian noise. Now, when we have the data, 95% confidence region of posterior is much smaller than the confidence region of prior. We show both regression results, from state-space model and from Gaussian process regression model. You cannot see the difference. Sum of the squared differences between variances and means of the results are $\mathrm{O}(10^{-28})$. It is in the same order of magnitude as numerical precision of the computation program. Figure 6b shows that the actual values are inside the 95% confidence region of the posterior.



(a) Random draws from the prior of Gaussian process regression. Green shows 95% confidence region. Crosses represent noise measurements from a random draw. Bold black line represents the random draw.



(b) Gaussian process regression to noise measurements (crosses) of random realization of Gaussian process (bold black line). The same regression is computed after conversion to state-space form. Sum of squared differences between these regressions are $O(10^{-28})$. You cannot see the differences between the regression results: mean (bold red line) and confidence interval (green area).

Figure 6: Example of Gaussian process regression. Figure 6b: Crosses represent data used for regression. We generated the data from Gaussian process with Matérn covariance function and parameter values $\nu = 7/2$, σ^2 and $\ell = 1$. We added Gaussian noise to measurements. Gaussian noise had constant variance $\sigma_n^2 = 1$. Figure 6a shows 4 random realization from the Gaussian prior.

3.2 Experiment 2

This experiment demonstrates real data regression with state-space model. This experiment uses Mauna Loa Observatory data from Hawaii. Data is average monthly measurements of CO_2 concentration from year 1970 to the end of 2012. Data can be downloaded from Tans and Keeling (2013). We predict this data 20 years into the future. Same data, but different time period, was also used by Rasmussen and Williams (2006).

In this experiment, we have the three main steps. First we need to choose which models we want to use and initial values for hyperparameters. Then we need to find the optimal hyperparameters. Finally, we can predict the future values with optimized model.

As the prior, we use two squared exponential covariance functions and a resonator model. The first SE explains the longer distance variation, the second SE explains the smaller distance variation and resonator explains the annual periodic variation. The prior mean is set to zero.

Initial values for the models are as follows:

- The first squared exponential covariance function has initial values: $\sigma_1^2 = 10^5$, $\ell_1 = 10^2$ and $N_1 = 6$. It explains large-scale smooth variation.
- The second squared exponential covariance function has initial values: $\sigma_2^2 = 1$, $\ell_2 = 1$ and $N_2 = 6$. It explains small-scale smooth variation. Without this model, optimization will not converge, σ_2 will be too large.
- The resonator model has initial values: $Q_{c,3} = 0.01$ and $f_3 = 1$. This model explains variation during the year.

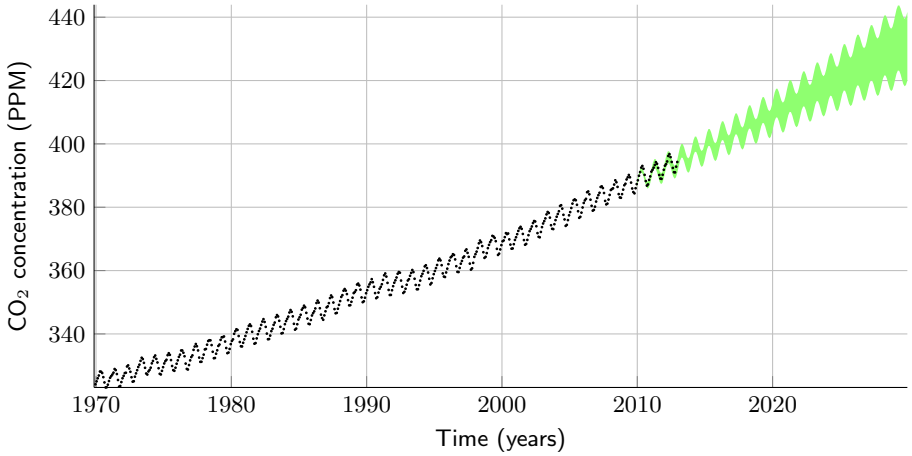
All the parameters are chosen by manually minimizing sensitivity of initial values and at the same time so that the result looks reasonable. Parameters N_1 and N_2 are not optimized and these are chosen so that approximation is close enough to the real SE covariance function. Differences between the exact SE covariance and the approximation can be seen in Figure 4. The parameter f_3 is not optimized, because optimization algorithm did not change the value, but it disturbed the optimization.

After choosing the initial values and models, parameter optimization is done by the methods explained in Section 2.2.4. Optimization gives the values:

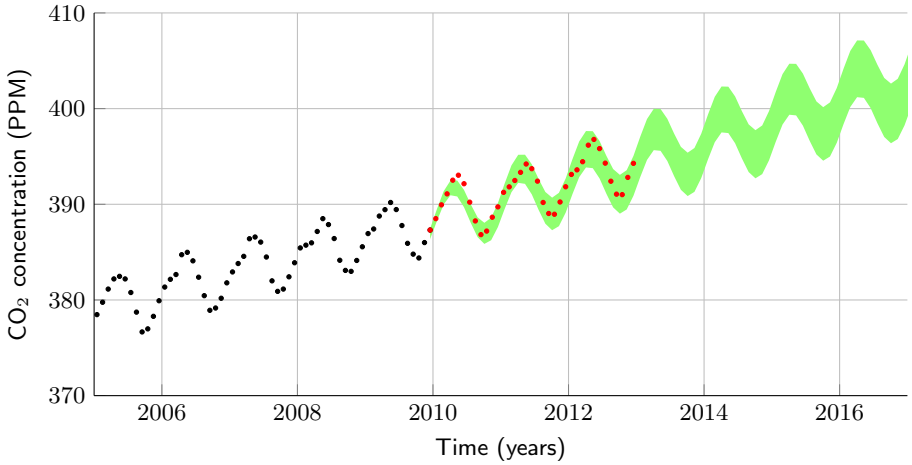
$$\begin{aligned} \sigma_1^2 &\approx 9.28 \cdot 10^4, & \ell_1 &\approx 126, \\ \sigma_2^2 &\approx 0.367, & \ell_2 &\approx 1.17, \\ Q_{c,3} &\approx 3.35 \cdot 10^{-3}. \end{aligned} \tag{38}$$

Now we can do the regression as in Experiment 1. First we form the model matrices for every model as in Section 2.2.3 for SE covariances and by Equation (37) for the resonator. Then we sum the models to superposition model using Equations (35) and (36). Then we discretize it with Equation (12) and solve the needed stationary covariance from Equation (16). Finally from the discretized solution, we get the regression result with Kalman filter and RTS smoother in Equations (14) to (17). Actually, in Kalman filtering and smoothing, we need to discretize the continuous superposition model every time the step size changes. Kalman filtering and smoothing are continued for the 20 additional years. Update step of the Kalman filter is skipped for these additional years.

Next we show the Kalman filtering and smoothing solution. Figure 7a shows the result. Green area represents 95% confidence region of prediction and black points represent the data. The predicted mean seems reasonable. Mean of the prediction continues growing with the same speed as earlier. Mean is at center of the green area. The predicted confidence region looks almost the same as in Rasmussen and Williams (2006). We did not use the last three years for the regression but for evaluation of the regression. Figure 7b shows more closely those three years. It shows that prediction is good at least for the first three years. In that sense, we can assume that the prediction is good also for the later years.



(a) Mauna Loa data with prediction



(b) Evaluation of prediction

Figure 7: The 516 observations of monthly averages of the atmosphere concentration of CO_2 from 1970 to the end of 2012, together with 95% predicted confidence region for a Gaussian process regression model, 20 years into the future. We used state-space inference instead of original Gaussian process regression. Green area represents the 95% confidence region of the prediction and black points represent the data. Figure 7a shows the used data with prediction. We did not use the last three years for the regression but for evaluation of the regression. Figure 7b shows the first three predicted ears more closely. For these three years we had also data. Figure 7b shows that prediction for the first three years is close to the data.

4 Conclusion and discussion

We have shown how some Gaussian process regression models can be converted into state-space models and further into Kalman filtering and smoothing problems. The computation complexity in GPR models is $O(n^3)$ and in state-space models computational complexity is $O(n)$, where n is number of training points. This was demonstrated in Experiment 1. Because of the computational complexity, state-space models are much more effective for large data sets than GPR models. In Experiment 1 we also tested that the conversion from a GPR model into state-space model does not effect on the regression result, when using Matérn covariance function as a prior.

In Experiment 2 we showed another benefit from the conversion. When a GPR model is converted into a state-space model, we can combine it with other state-space models and many real processes can be represented as state-space models. In Experiment 2 we used resonator model, which is state-space models. In Experiment 2 we also computed regression for real data. The prediction looks reasonable and it is close to the one computed in Rasmussen and Williams (2006) for the same data. They used a GPR model for prediction.

In Experiment 2 we had two squared exponential covariance functions as the prior. We showed that in this case the model cannot be converted exactly into a state-space model. We chose an approximation degree such that the result was close enough to the original Gaussian process. The model could not be converted exactly because of smoothness. The squared exponential covariance function is infinite smooth. During the conversion we approximated the GP as solution to nth order SDE, where n is finite integer. Because of nth order SDE, infinite smoothness results infinite-dimensional model matrices. The first model had finite smoothness, so we had finite-dimensional model matrices and we needed no approximation with it.

With our method, a spectral density has to be in specific form. It has to be rational function, both denominator and numerator has to be polynomial of squared frequency, and degree of the numerator has to be smaller than degree of the denominator. Spectral densities of all covariance functions are not that form. Some spectral densities can be approximated as such form, even if they are not that form originally. There might exists some covariance functions which cannot be approximated in that form at all. We also found other type of problems during the conversion. The spectral density is Fourier transform of the prior covariance function. Fourier transform does not converge for all functions.

We predict that using the same methods backwards we can convert state-space models into Gaussian process regression models. In that case we can

combine converted state-space models with GP models which cannot be converted into state-space models. We can also predict that only some state-space models can be converted into GP models.

We can use the methods represented in this thesis to convert other GPR models into state-space models. The methods can be extended to multidimensional, or *spatio-temporal*, case this is shown by Solin (2012). We can improve also the optimization by computing gradient function of the energy function as was shown by Mbalawata et al. (2012). Some optimization algorithms compute the gradient numerically if the gradient function is not given by user (MathWorks, 2013a). Numerical gradient is ineffective to compute, the original energy function has to be evaluated many times on every iteration and it might be inaccurate. Inaccuracy of the gradient slows the optimization and makes it inaccurate.

In this thesis, we managed to reduce computational complexity of two common Gaussian process regression models. We did this by converting the Gaussian process regression models into the state-space models. The result was exactly the same when we used the Matérn covariance function as a priori. The methods can be used to convert more Gaussian processes into state-space models.

References

M. Abramowitz and I. Stegun. *Handbook of Mathematical Functions: With Formulars, Graphs, and Mathematical Tables*. Dover, New York, 1965.

J. Hartikainen. *Sequential Inference for Latent Temporal Gaussian Process Models*. PhD thesis, Aalto University, 2013.

J. Hartikainen and S. Särkkä. Kalman filtering and smoothing solutions to temporal gaussian process regression models. In *IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 379–384, 2010.

R. Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82(1), 1960.

MathWorks. Matlab documentation: function fmincon, 2013a. URL http://www.mathworks.se/help/optim/ug/fmincon.html. Accessed: September 3, 2013.

MathWorks. Matlab documentation: function randn, 2013b. URL http://www.mathworks.se/help/matlab/ref/randn.html. Accessed: September 3, 2013.

I. S. Mbalawata, S. Särkkä, and H. Haario. Parameter estimation in stochastic differential equations with markov chain monte carlo and non-linear kalman filtering. *Computational Statistics*, 28(3):1–29, 2012.

C. E. Rasmussen and C. K. Williams. *Gaussian Process for Machine Learning*. The MIT Press, 2006.

H. Rauch, F. Tung, and C. Striebel. Maximum likelihood estimates of linear dynamic systems. *AIAA journal*, 3(8), 1965.

S. Särkkä. *Recursive Bayesian Inference on Stochastic Differential Equations*. PhD thesis, Helsinki University of Technology, 2006.

S. Särkkä. *Bayesian Filtering and Smoothing*. Cambridge University Press, 2013.

S. Särkkä and J. Hartikainen. Infinite-dimensional Kalman filtering approach to spatio-temporal Gaussian process regression. *AISTATS*, 22:993–1001, 2012.

S. Särkkä, A. Solin, A. Nummenmaa, A. Vehtari, T. Auranen, S. Vanni, and F.-H. Lin. Dynamic retrospective filtering of physiological noise in BOLD fMRI: DRIFTER. *NeuroImage*, 60(2):1517–1527, 2012.

S. Särkkä, A. Solin, and J. Hartikainen. Spatiotemporal learning via infinite-

dimensional Bayesian filtering and smoothing. *IEEE Signal Prosessing magazine*, 30(4), 2013.

A. Solin. Hilbert space methods in infinite-dimensional Kalman filtering. Master's thesis, Aalto University, 2012.

P. Tans and R. Keeling. Mauna Loa CO_2 monthly mean data, 2013. URL http://www.esrl.noaa.gov/gmd/ccgg/trends/. Accessed: September 3, 2013.

A Summary in Finnish

Tämä työ keskittyi kahteen menetelmään: regressiomalleihin gaussisten prosessien avulla (GPR-malleihin) ja tila-avaruusmalleihin. GPR-mallit ovat viime aikoina lisääntyneet niiden helppouden ja hyvän sovellettavuuden takia. GPR-mallit ovat kuitenkin laskennallisesti raskaita. Niissä laskenta-aika kasvaa kuutiollisesti, kun havaintopisteitä lisätään. Tila-avaruusmallit vastaavat tähän ongelmaan, koska niissä laskenta-aika kasvaa ainoastaan lineaarisesti suhteessa laskentapisteiden määrään. Tila-avaruusmallit ovat myös yleisiä fysikaalisten ilmiöiden selittämisessä. Tämän työn päätavoitteena oli muuntaa kaksi yleistä GPR-mallia vastaavaan tila-avaruusmuotoon. Tila-avaruusmuodot muunnettiin vielä edelleen Kalmanin suodatus- ja silotusongelmiksi, jotka ovat yleisiä ja tehokkaita ratkaisumenetelmiä tila-avaruusmalleille.

Ensimmäinen koetilanne tehtiin simuloidulla aineistolla, jossa saadaan hyvin esiin muunnoksen ominaisuudet. Toinen koetilanne tehtiin aidolla aineistolla ja menetelmiä käytettiin ennustamiseen. Jälkimmäinen koetilanne oli monimutkaisempi ja näytti samalla, minkälaisiin ongelmiin menetelmiä voi soveltaa.

Gaussinen prosessi on moniulotteisen normaalijakauman yleistys ääretönulotteiseksi. GPR-malleissa estimoidaan funktioita oletuksella, että funktio olisi gaussinen prosessi. GPR-malleissa aineiston sisäisiä riippuvuuksia mallinnetaan kovarianssifunktiolla eikä esimerkiksi lineaarisilla kertoimilla. GPR-mallien tuloksena saadaan suoraan estimoitavan funktion arvoja eikä esimerkiksi lineaarisia kertoimia. Lineaarisessa regressiossa funktion arvot lasketaan jälkikäteen näistä lineaarisista kertoimista. Tämän kertoimien puuttumisen takia GPR mallit ovat joustavampia kuin lineaarinen regressio.

Tila-avaruusmalleissa tilojen välisiä riippuvuuksia mallinnetaan (stokastisella) differentiaaliyhtälöllä. Lisäksi tiloja ei välttämättä tunneta suoraan vaan niistä saadaan häiriöllisiä mittauksia. Näistä häiriöllisistä mittauksista pyritään ratkaisemaan tilat. Tilat pystytään ratkaisemaan optimaalisesti esimerkiksi Kalmanin suodatus- ja silotusalgoritmeilla.

Yksi tapa muuntaa GPR-malli tila-avaruusmuotoon on approksimoida käytettyä gaussisen prosessin prioria stokastisella differentiaaliyhtälöllä. Tällöin alkuperäinen gaussinen prosessi saadaan stokastisen differentiaaliyhtälön ratkaisuna. Tällaisessa stokastisessa differentiaaliyhtälössä summataan gaussisen prosessin derivaattoja yhteen niin, että tulokseksi saadaan valkoista kohinaa. Kun on olemassa lineaarinen stokastinen differentiaaliyhtälö, siitä saadaan yhtälöä uudelleenjärjestämällä tila-avaruusmuoto.

Stokastisen differentiaaliyhtälön kertoimet saadaan spektrin avulla. Sekä priorin kovarianssifunktiosta että stokastisen differentiaaliyhtälön ratkaisus-

ta on laskettavissa spektri. Määräämällä spektrit arvoiltaan yhtä suuriksi voidaan tuntemattomat kertoimet selvittää. Stokastisen differentiaaliyhtälön spektri on muotoa polynomi jaettuna polynomilla. Lisäksi nimittäjän polynomin pitää olla suurempaa astetta kuin osoittajan. Jos priorin kovarianssista saatu spektri ei ole tällaista muotoa, sitä pitää approksimoida sellaisena. Tämän jälkeen stokastisen differentiaaliyhtälön kertoimien arvoilla on vielä useita vaihtoehtoja ja niistä täytyy valita sellaiset arvot, että systeemistä tulee stabiili. Tila-avaruusmuoto ratkaistaan järjestämällä saatu stokastinen differentiaaliyhtälö uudelleen. Tämä tila-avaruusmuoto on yleisesti jatkuva, ja se pitää diskretoida laskentapisteissä. Diskreettiin tila-avaruusmuotoon voidaan käyttää Kalmanin suodatus- ja silotusalgoritmeja ongelman ratkaisemiseksi.

Ensimmäinen muunnettu gaussinen prosessi kuului Matérn luokkaan. Ensimmäisessä sovellusesimerkissä testattiin tämän muunnoksen ominaisuuksia itse luodulla aineistolla. Mallin muunnoksessa ei tehty mitään approksimaatioita ja siten regression tuloksissa ei ollut eroa. Laskenta-aika erosi kuitenkin merkittävästi. Tila-avaruusmallin laskenta-aika kasvoi lineaarisesti ja alkuperäisen GPR-mallin laskenta-aika kasvoi huomattavasti lineaarista nopeammin. Laskennat suoritettiin kolmekymmentä kertaa ja näistä otettiin keskiarvot. Näin saatiin poistettua häiriöt laskenta-ajoista.

Toisella muunnetulla gaussisella prosessilla oli neliöllinen eksponenttikovarianssifunktio, ja sitä käytettiin toisessa sovellusesimerkissä. Tämän GP:n muuntamisessa piti käyttää approksimaatiota. Approksimaatioaste valittiin sillä tavalla että saatu spektri oli riittävän lähellä alkuperäistä. Kaikkiaan tässä esimerkissä oli käytössä kolme mallia: kaksi gaussista prosessia, joilla oli neliölliset eksponentti-kovarianssifunktiot, ja lisänä resonaattorimalli. Resonaattorimalli on tila-avaruusmalli.

Toisessa esimerkissä aineistona käytettiin Mauna Loan mittausaseman kuukausittaisia hiilidioksidipitoisuuksia vuodesta 1970 vuoden 2012 loppuun. Tätä aineistoa ennustettiin kaksikymmentä vuotta eteenpäin. Hyperparametrien suurimman uskottavuuden optimointi löysi selvän (paikallisen) optimin, ja sama optimi löytyi, vaikka hyperparametrien alkuarvoja muunneltiin. Tulos vaikutti järkevältä. Ennusteen mukaan hiilidioksidipitoisuudet kasvavat myös tulevaisuudessa ja kasvunopeus hieman kiihtyy lähivuosina. Ennusteesta näkyi selvästi vuoden sisäinen vaihtelu usealle vuodelle eteenpäin. Lähivuosille ennuste antoi pienen luottamusvälin, ehkä jopa epärealistisen pienen. Myöhempinä vuosina luottamusväli levenee, mutta leveneminenkin oli yllättävän hidasta.

Tarve toisen esimerkin approksimaatiolle tuli siitä, että neliöllinen eksponentti-kovarianssifunktio on äärettömän sileä. Gaussista prosessia approksimoitiin ratkaisuna äärellisulotteiselle stokastiselle differentiaaliyhtälölle, kun GP-malli muunnettiin tila-avaruusmuotoon. Tästä seurasi, että tila-

avaruusmatriisien koko oli verrannollinen kovarianssifunktion sileydelle. Äärettömän sileän kovarianssifunktion tila-avaruusmatriisit olisivat ääretönulotteisia. Lisärajoituksia GP:n priori kovarianssifunktiolle antoi myös Fourierin muunnos. Spektri saatiin Fourierin muunnoksella priori kovarianssifunktiosta. Fourierin muunnos on integraalimuunnos eikä ole laskettavissa kaikille funktioille.

Toisen esimerkin resonaattorimalli havainnollisti sitä, kuinka tila-avaruusmalleja ja GPR-malleja voidaan yhdistää ensin muuntamalla GPR-malli tila-avaruusmuotoon. Tämän työn tulosten perusteella voisi ennustaa, että käyttämällä vastaavia menetelmiä vastakkaiseen suuntaan pystyttäisiin muuntamaan tila-avaruusmalleja GPR-malleiksi. Lisäksi voisi ennustaa, että vain osa tila-avaruusmalleista pystytään muuntamaan GPR-malleiksi. Tällöin myös tila-avaruusmuotoon muuntumattomat GPR-mallit voitaisiin yhdistää tila-avaruusmalleihin.

Tässä työssä esiteltyjä menetelmiä pystyy käyttämään uusien GPR-mallien muuntoon tila-avaruusmuotoon. Lisäksi menetelmät ovat laajennettavissa useampaan ulottuvuuteen, ja esimerkiksi hyperparametrien optimointia pystyy tehostamaan ja tarkentamaan laskemalla energiafunktion gradientin. Jos käyttäjä ei anna gradienttia, osa optimointialgoritmeista laskee gradientin numeerisesti. Tällainen numeerinen gradientin laskenta ei välttämättä tuota tarkkaa gradientin arvoa, ja se häiritsee optimointia. Lisäksi numeerisen gradientin laskenta vaatii useita laskentakertoja alkuperäiselle funktiolle, ja se on laskennallisesti raskasta.