State Estimation for Nonlinear Dynamic Systems using Gaussian Processes and Pre-computed Local Linear Models

Xiaoke Yang, Bo Peng, Hang Zhou, and Lingyu Yang

Abstract— **State estimation of nonlinear dynamic systems is an important problem in practice. This paper proposes a recursive state estimation method for nonlinear dynamic systems using Gaussian processes (GP) and pre-computed local linear models. Gaussian processes exhibit remarkable learning, or nonlinear regression capabilities from measurement data. The incorporation of pre-computed local linear models reduces the amount of data required and improves the regression performance of the GP. Based on such an improved GP model for nonlinear dynamic systems, a recursive Bayesian filtering method is implemented for the estimation of the unknown states of the system. Simulations on an aircraft benchmark model demonstrate that the proposed method is capable of estimating the unknown angle of attack of the aircraft, and the existence of the local linear models significantly improves the estimation performance of the filter. This method is especially useful in a control systems design context in which local linearisations of nonlinear dynamic systems are usually readily available.**

Index Terms— **State Estimation, Nonlinear Dynamic Systems, Gaussian Process, Bayesian Filtering Method, Derivative Observation.**

I. Introduction

State estimation of dynamic systems is a crucial element of many industrial applications. For example, in the pose estimation of robots, the integrated navigation systems, and the motion capturing and tracking systems, the position, velocity or orientation of an object need to be constructed from a multitude of sensor measurements. In linear dynamic systems contaminated by white Gaussian noise, the Kalman filter is the proven optimal solution. While in nonlinear dynamic systems, due to the nonlinearity in the state transition and the output functions, closed-form analytical solutions are usually not available. Various approximations have been extensively discussed in literature, such as the extended Kalman filter (EKF) [1], the unscented Kalman filter (UKF) [2], and the particle filter (PF) or sequential Monte Carlo (SMC) method [3]. Most of these estimation or filtering methods are based on a description or model of the system in order to interpret the relations between the unknown state and the output measurement. Gaussian processes (GP), due to their flexibility and expressiveness [4], [5], have been widely used in modelling nonlinear systems and thus naturally serve as the model for various state estimation methods.

A cluster of Bayesian filters is proposed in [6] for dynamic systems modelled by a GP. The idea is that for systems without readily available parametric models, with sufficient offline training data and pre-trained hyper-parameters, the GP model could provide a reasonably good data-based representation of the system. Thus, a two-step Bayesian filter could be implemented over this GP. This cluster of GP-based filters included a GP-EKF, a GP-UKF, and a GP-PF. [7] used a moment-matching method or an assumed density filter (ADF) to propagate the full state distribution and approximate the propagated density with a Gaussian. This method was further improved by [8] which proposed the use of expectation propagation for further refinement of the propagated state distribution. Additionally, [5] proposed the use of a robust Rach-Tung-Striebel smoother for the smoothing problems of Gaussian process dynamic systems.

The above methods are all based on GPs trained with offline data [1]. Thus the offline data is critical for the performance of the methods. Apart from the ordinary input-output data, in literature, derivative observations or pre-computed local linear models have also been proposed as another type of data for the GP [9]. Remarkable performance improvement was demonstrated both in the GP model itself and various control methods utilising the GP model with derivative observations [10], [11].

This paper then proposes the use of derivative observations in the state estimation of nonlinear dynamic systems modelled by a GP. The key idea is that by training over the pre-computed local linear models, the GP fits into these linear models and generates a nonlinear mapping which describes the system's input-output behaviour. Since linearisation is a highly effective and compact form to describe the local behaviour of a nonlinear function, it then leads to both the improvement of the regression performance of the GP and the reduction in the size of the data stored. Similar to the Bayesian filters proposed in literature, this paper also employs an online recursive estimation framework.

The structure of the paper is organised as follows. Section I introduces the background for the proposed method and reviews relevant existing work. Section II briefly summaries the necessary preliminary knowledge on Gaussian processes and derivative observations. Section III then discusses in detail on Gaussian process state-space models for nonlinear

This research was supported by National Natural Science Foundation of China (Grant No. 61273099 and 61304030).

Xiaoke Yang, Bo Peng, Hang Zhou, and Lingyu Yang are with the School of Automation Science and Electrical Engineering, Beihang University, Beijing, 100191, China. das.xiaoke@hotmail.com, nake_peng@163.com, zh506206182@163.com, yanglingyu@buaa.edu.cn.

Code for a Matlab/Octave implementation of Gaussian process regression with derivative observations is available online at https://github.com/ teancake/gpr_dob/.

[1]There are also methods in literature which do not solely rely on offline data and accomplish online inference and learning simultaneously. A discussion on those methods is included in the conclusions.

dynamic systems with local linearisations and the proposed state estimation method. Section IV gives a case study of the proposed method on an aircraft benchmark model. Conclusions and further work then follow in Section V.

II. Gaussian Processes and Derivative Observations

A. Gaussian Processes

A Gaussian process [4], [12] can be treated as a distribution over functions $f(\mathbf{z}) : \mathbb{R}^D \mapsto R$, and is denoted as

$$f \sim \mathcal{GP}(m(\mathbf{z}), k(\mathbf{z}^m, \mathbf{z}^n)), \tag{1}$$

where $m(\mathbf{z}) : \mathbb{R}^D \mapsto \mathbb{R}$ is the mean function, $k(\mathbf{z}^m, \mathbf{z}^n) : \mathbb{R}^D \times \mathbb{R}^D \mapsto \mathbb{R}$ is the covariance function, and $\mathbf{z}^m, \mathbf{z}^n \in \mathbb{R}^D$ are two input points. In this paper a specific combination, the zero mean function and the squared exponential covariance function, will be used, i.e.

$$m(\mathbf{z}) \equiv 0, \tag{2}$$

$$k_f(\mathbf{z}^m, \mathbf{z}^n) = \alpha \exp\left(-\frac{1}{2}\|\mathbf{z}^m - \mathbf{z}^n\|^2_{\mathbf{\Gamma}}\right), \tag{3}$$

since a GP with the two of them describes a universal approximator to nonlinear smooth functions [13]. In (3), $\mathbf{\Gamma} = \mathrm{diag}\left(\begin{bmatrix}\gamma_1 & \gamma_2 & \cdots & \gamma_D\end{bmatrix}\right)$, notation $\|\cdot\|^2_{\mathbf{\Gamma}}$ is defined as $\|\mathbf{z}\|^2_{\mathbf{\Gamma}} \triangleq \mathbf{z}^\top \mathbf{\Gamma} \mathbf{z}$.

In practice, the measurement of function f is usually contaminated by noise, i.e. $y = f + v$, $v \sim \mathcal{N}(0, \nu)$. In order to capture the measurement noise, an additional term is added to k_f, giving

$$k(\mathbf{z}^m, \mathbf{z}^n) = \alpha \exp\left(-\frac{1}{2}\|\mathbf{z}^m - \mathbf{z}^n\|^2_{\mathbf{\Gamma}}\right) + \nu \delta_{m,n}, \tag{4}$$

where the Kronecker delta is defined as

$$\delta_{m,n} = \begin{cases} 1 & m = n \\ 0 & m \neq n \end{cases}. \tag{5}$$

Parameters in (4) are called hyper-parameters, and are represented as $\boldsymbol{\theta} = \begin{bmatrix}\gamma_1 & \gamma_2 & \cdots & \gamma_D & \alpha & \nu\end{bmatrix}^\top$. Given some input-output data $\mathbf{Z} = \begin{bmatrix}\mathbf{z}^1 & \cdots & \mathbf{z}^N\end{bmatrix}$ and $\mathbf{Y} = \begin{bmatrix}y^1 & \cdots & y^N\end{bmatrix}$, an optimal value for $\boldsymbol{\theta}$ can be found which maximises the log marginal likelihood

$$\log p(\mathbf{Y}|\mathbf{Z}, \boldsymbol{\theta}) = -\frac{1}{2}\mathbf{Y}^\top \mathbf{K}^{-1}\mathbf{Y} - \frac{1}{2}\log|\mathbf{K}| - \frac{N}{2}\log 2\pi, \tag{6}$$

where $\mathbf{K} \in \mathbb{R}^{N\times N}$ is the covariance matrix of the output data $\mathbf{Y}$ and is constructed from the covariance function as

$$\mathbf{K} = \left[k(\mathbf{z}^m, \mathbf{z}^n)\right]_{\substack{m\in\{1,\ldots,N\}\\ n\in\{1,\ldots,N\}}}. \tag{7}$$

Maximisation of (6) is usually referred to as the training of the GP.

Given the GP with the input-output data and the trained hyper-parameters, an output prediction could be computed when a test input $\mathbf{z}^*$ is provided, as $p(y^*|\mathbf{Z}, \mathbf{Y}, \mathbf{z}^*) \sim \mathcal{N}(\mu, \Sigma)$, where

$$\mu = k(\mathbf{z}^*, \mathbf{Z})\mathbf{K}^{-1}\mathbf{Y}, \tag{8}$$

$$\Sigma = k(\mathbf{z}^*, \mathbf{z}^*) - k(\mathbf{z}^*, \mathbf{Z})\mathbf{K}^{-1}k(\mathbf{Z}, \mathbf{z}^*), \tag{9}$$

and

$$k(\mathbf{z}^*, \mathbf{Z}) = \left[k(\mathbf{z}^*, \mathbf{z}^n)\right]_{n\in\{1,\ldots,N\}}. \tag{10}$$

B. Gaussian Processes with Derivative Observations

Derivative observations for Gaussian processes was proposed in [9] and refers to the linearisations of the underlying nonlinear function which is modelled by the GP. The term 'observation' simply means the linearisations of the underlying function pre-computed or 'observed' *offline*.

Derivative observations serve as part of the offline data for the GP, and a GP with derivative observations contains a series of such linearisations, in addition to the conventional input-output data, i.e.

$$\mathbf{Z} = \begin{bmatrix}\mathbf{z}_d^1 & \cdots & \mathbf{z}_d^M & \mathbf{z}^1 & \cdots & \mathbf{z}^N\end{bmatrix}, \tag{11}$$

$$\mathbf{Y} = \begin{bmatrix}(\mathbf{y}_d^1)^\top & \cdots & (\mathbf{y}_d^M)^\top & y^1 & \cdots & y^N\end{bmatrix}, \tag{12}$$

where $\mathbf{y}_d \in \mathbb{R}^D$ is the derivative of the function (1) taken at the input location of $\mathbf{z}_d \in \mathbb{R}^D$, i.e.

$$\mathbf{y}_d^i = \frac{\mathrm{d}f}{\mathrm{d}\mathbf{z}}\bigg|_{\mathbf{z}=\mathbf{z}_d^i}, \quad i \in \{1, \ldots, M\}. \tag{13}$$

M is the number of derivative observations and N is the number of function output data points. The covariance matrix of the augmented output data $\mathbf{Y}$, when written in a block matrix form, becomes

$$\mathbf{K} = \begin{bmatrix}\mathbf{K}_{dd} & \mathbf{K}_{dz} \\ \mathbf{K}_{zd} & \mathbf{K}_{zz}\end{bmatrix}, \tag{14}$$

where

$$\mathbf{K}_{dd} = \left[k_{dd}(\mathbf{z}_d^m, \mathbf{z}_d^n)\right]_{\substack{m\in\{1,\ldots,M\}\\ n\in\{1,\ldots,M\}}}, \tag{15}$$

$$\mathbf{K}_{dz} = \left[k_{dz}(\mathbf{z}_d^m, \mathbf{z}^n)\right]_{\substack{m\in\{1,\ldots,M\}\\ n\in\{1,\ldots,N\}}}, \tag{16}$$

$$\mathbf{K}_{zd} = \left[k_{zd}(\mathbf{z}^m, \mathbf{z}_d^n)\right]_{\substack{m\in\{1,\ldots,N\}\\ n\in\{1,\ldots,M\}}}, \tag{17}$$

$$\mathbf{K}_{zz} = \left[k(\mathbf{z}^m, \mathbf{z}^n)\right]_{\substack{m\in\{1,\ldots,N\}\\ n\in\{1,\ldots,N\}}}. \tag{18}$$

Formulae for $k_{dd}(\mathbf{z}_d^m, \mathbf{z}_d^n) \in \mathbb{R}^{D\times D}$, $k_{dz}(\mathbf{z}_d^m, \mathbf{z}^n) \in \mathbb{R}^{D\times 1}$, and $k_{zd}(\mathbf{z}^n, \mathbf{z}_d^m) \in \mathbb{R}^{1\times D}$ are

$$k_{dd}(\mathbf{z}_d^m, \mathbf{z}_d^n) = k_f(\mathbf{z}_d^m, \mathbf{z}_d^n) \left(\mathbf{\Gamma} - \mathbf{\Gamma}(\mathbf{z}_d^m - \mathbf{z}_d^n)(\mathbf{z}_d^m - \mathbf{z}_d^n)^\top \mathbf{\Gamma}\right), \tag{19}$$

$$k_{dz}(\mathbf{z}_d^m, \mathbf{z}^n) = -k_f(\mathbf{z}_d^m, \mathbf{z}^n)\mathbf{\Gamma}(\mathbf{z}_d^m - \mathbf{z}^n), \tag{20}$$

$$k_{zd}(\mathbf{z}^n, \mathbf{z}_d^m) = k_{dz}(\mathbf{z}_d^m, \mathbf{z}^n)^\top. \tag{21}$$

Apart from the changes in the data and the covariance matrix listed above, the inclusion of derivative observations will not affect other aspects of a GP, such as the the training and the prediction, as given in the previous subsection. More information on derivative observations for GP could be found in [9], [10], and [11].

To give an intuition of how derivative observations could improve the regression performance of a GP, a simple illustrious example of a one dimensional function regression problem is shown as follows.

Suppose there is an unknown function $y = f(z) : \mathbb{R} \mapsto \mathbb{R}$, and f is modelled by a GP, i.e. $f \sim \mathcal{GP}(0, k(z, z'))$. For this function, the following output data and derivative observations are available at $z_o^1 = -3$, $z_o^2 = 0$, and $z_o^3 = 3$,

$$y_o^1 = 3, \quad y_o^2 = -1, \quad y_o^3 = -3, \tag{22}$$

$$\left.\frac{\mathrm{d}y}{\mathrm{d}z}\right|_{z_o^1} = -2, \quad \left.\frac{\mathrm{d}y}{\mathrm{d}z}\right|_{z_o^2} = -1, \quad \left.\frac{\mathrm{d}y}{\mathrm{d}z}\right|_{z_o^3} = 0. \tag{23}$$

With the function output data (22) as the only data for the GP, the expression of f given by the GP is shown in Fig. 1, in which the solid line indicates the mean and the shaded area represents the 95% confidence interval. With both the function output (22) and the derivative observation (23) as the data, the expression of f given by the GP is shown in Fig. 2. The hyper-parameters of the GP are fixed in these two cases.

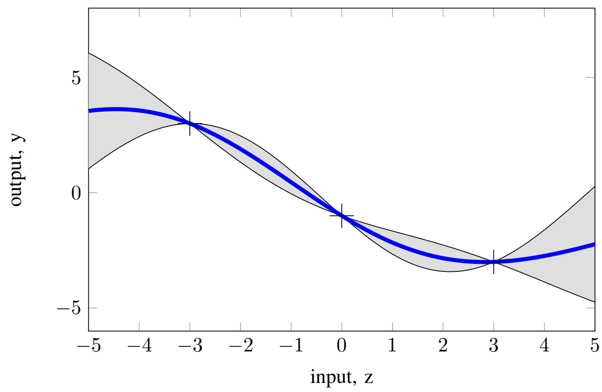

Fig. 1: Expression of the function given by the GP with only function output data, no derivative observations are included.

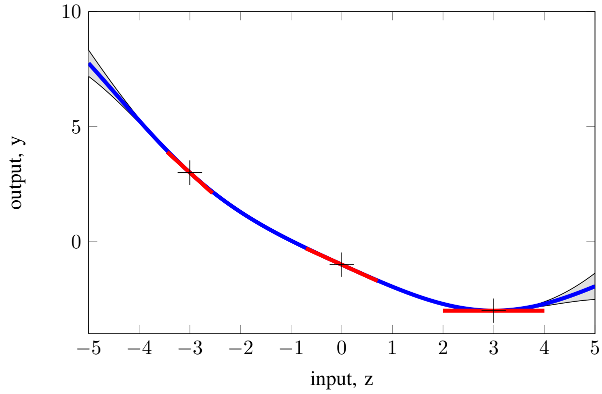

Fig. 2: Expression of the function given by the GP with both the function output and the derivative observations.

It can be seen from the comparison that the derivative observations not only change the shape of the mean to satisfy the derivatives at the given inputs, the uncertainty over the expression is also significantly reduced.

III. Nonlinear Dynamic Systems State Estimation

A. Nonlinear Dynamic Systems Modelling with GP and Pre-computed Local Linear Models

Various forms of GP models for dynamic systems exist in literature, such as the nonlinear auto-regressive with exogenous input (NARX) model, the nonlinear auto-regressive moving-average with exogenous input (NARMAX) model, etc. Here we use a state-space model to represent the underlying nonlinear system, i.e.

$$\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k), \tag{24}$$

$$\mathbf{y}_k = \mathbf{g}(\mathbf{x}_k, \mathbf{u}_k), \tag{25}$$

where $\mathbf{x} \in \mathbb{R}^{N_x}$ is the state variable, $\mathbf{u} \in \mathbb{R}^{N_u}$ is the input variable, $\mathbf{y} \in \mathbb{R}^{N_y}$ is the output variable, k is the discrete time instant, $\mathbf{f}(\cdot)$ and $\mathbf{g}(\cdot)$ represent the state transition function and the output function, respectively. Both functions are modelled by Gaussian processes, i.e. $\mathbf{f} \sim \mathcal{GP}$, $\mathbf{g} \sim \mathcal{GP}$.

Note that in (24) and (25), there are no additive system noise or measurement noise as in a standard state-space model, the reason is that the two GPs modelling $\mathbf{f}$ and $\mathbf{g}$ could conveniently incorporate the noise through an additional term in the covariance function as in (4). Furthermore, since $\mathbf{f}$ is a vector function, $\mathbf{f} \sim \mathcal{GP}$ means that each dimension of the function $\mathbf{f}$ is a GP, i.e.

$$f_j \sim \mathcal{GP}\left(0, k(\mathbf{z}, \mathbf{z}')\right), \quad j \in \{1, 2, \ldots, N_x\}, \tag{26}$$

where $\mathbf{z} = \begin{bmatrix}\mathbf{x}^\top & \mathbf{u}^\top\end{bmatrix}^\top$.

To simplify the problem, we restrict our discussion to a specific yet common case in practice, that is, the output function (25) is an indexing function which extracts certain elements from the state variable. If we define $\boldsymbol{\zeta} \in \mathbb{R}^{N_\zeta}$ as a sub-vector in $\mathbf{x}$ whose measurements are available, and the vector containing the rest of the unknown variables in $\mathbf{x}$ is denoted as $\boldsymbol{\rho} \in \mathbb{R}^{N_x - N_\zeta}$. Then the state vector can be written as $\mathbf{x} = [\boldsymbol{\zeta}^\top \quad \boldsymbol{\rho}^\top]^\top$, and

$$\boldsymbol{\zeta}_{k+1} = \mathbf{f}_{\boldsymbol{\zeta}}(\mathbf{x}_k, \mathbf{u}_k) \tag{27}$$

$$\boldsymbol{\rho}_{k+1} = \mathbf{f}_{\boldsymbol{\rho}}(\mathbf{x}_k, \mathbf{u}_k) \tag{28}$$

and

$$\mathbf{f} = \begin{bmatrix}\mathbf{f}_{\boldsymbol{\zeta}} \\ \mathbf{f}_{\boldsymbol{\rho}}\end{bmatrix}, \quad \mathbf{g} = \mathbf{f}_{\boldsymbol{\zeta}}. \tag{29}$$

Also, all of the N_x GPs use the same covariance function.

At some pre-specified values (usually the operating points or equilibria of the system) of $\mathbf{x}$ and $\mathbf{u}$, say $(\mathbf{x}_o^i, \mathbf{u}_o^i)$, $i \in \{1, 2, \ldots, I\}$, where I is the number of operating points, linearisation of the system dynamics (24) is written as

$$\mathbf{x}_{k+1} = \mathbf{A}^i(\mathbf{x}_k - \mathbf{x}_o^i) + \mathbf{B}^i(\mathbf{u}_k - \mathbf{u}_o^i) + \mathbf{F}^i, \tag{30}$$

where

$$\mathbf{A}^i = \left.\frac{\partial \mathbf{f}(\mathbf{x}, \mathbf{u})}{\partial \mathbf{x}}\right|_{\mathbf{x}_o^i, \mathbf{u}_o^i}, \tag{31}$$

$$\mathbf{B}^i = \left.\frac{\partial \mathbf{f}(\mathbf{x}, \mathbf{u})}{\partial \mathbf{u}}\right|_{\mathbf{x}_o^i, \mathbf{u}_o^i}, \tag{32}$$

$$\mathbf{F}^i = \mathbf{f}(\mathbf{x}_o^i, \mathbf{u}_o^i). \tag{33}$$

(30) suggests that the derivative observations for the GP should be $[\mathbf{A}^i \quad \mathbf{B}^i]$ and the function output data, i.e. the output of $\mathbf{f}_d$ at $(\mathbf{x}_o^i, \mathbf{u}_o^i)$, is $\mathbf{F}^i$, $i \in \{1, 2, \ldots, I\}$. To be specific, the local linear models provide two types of data for the GP, derivative observations $\begin{bmatrix}\mathbf{Z}_{Ld} \\ \mathbf{Y}_{Ld}\end{bmatrix}$ and function output data $\begin{bmatrix}\mathbf{Z}_{L} \\ \mathbf{Y}_{L}\end{bmatrix}$, where

$$\mathbf{Z}_{Ld} = \mathbf{Z}_L = \begin{bmatrix} (\mathbf{x}_o^1)^\top & (\mathbf{u}_o^1)^\top \\ \vdots & \vdots \\ (\mathbf{x}_o^I)^\top & (\mathbf{u}_o^I)^\top \end{bmatrix}^\top, \tag{34}$$

$$\mathbf{Y}_{Ld} = \begin{bmatrix}\mathbf{A}^1 & \mathbf{B}^1 & \cdots & \mathbf{A}^I & \mathbf{B}^I\end{bmatrix}, \tag{35}$$

$$\mathbf{Y}_{L} = \begin{bmatrix}\mathbf{F}^1 & \cdots & \mathbf{F}^I\end{bmatrix}. \tag{36}$$

Note that $\mathbf{Y}_{Ld}$ and $\mathbf{Y}_d$ are also compact representations for the output data of the N_x individual GPs. For the j^{th} GP modelling function f_j, the corresponding rows of the two matrices should be extracted for that GP.

B. Estimation of the Unknown State Variables

A description of the estimation problem to be tackled is as follows. There is a nonlinear dynamic system which is modelled by a GP in a state-space form (27) and (28). The GP is trained with a series of local linearisations of the nonlinear system $\{\mathbf{Z}_{Ld}, \mathbf{Z}_L, \mathbf{Y}_{Ld}, \mathbf{Y}_L\}$ and additional input-output data. Given the GP with the training data and the trained hyper-parameters, when the measurement of $\boldsymbol{\zeta}_k$ is available at time instant k , estimate the state $\mathbf{x}_{k-1}$, which includes the unknown state variable $\boldsymbol{\rho}_{k-1}$ and the state variable $\boldsymbol{\zeta}_{k-1}$ whose measurement is available but needs to be filtered.

A classical two-step Bayesian filtering framework is adopted, which includes a prediction step and an update step. By denoting all historical measurement as

$$\boldsymbol{\zeta}_{0:k} = \{\boldsymbol{\zeta}_0, \boldsymbol{\zeta}_1, \ldots, \boldsymbol{\zeta}_k\} \tag{37}$$

and omitting the dependency of the state on the input for clarity, the prediction of the density of $\mathbf{x}_{k-1}$ can be given by

$$p(\mathbf{x}_{k-1}|\boldsymbol{\zeta}_{0:k-1}) = \int p(\mathbf{x}_{k-1}|\mathbf{x}_{k-2}) p(\mathbf{x}_{k-2}|\boldsymbol{\zeta}_{0:k-1}) \mathrm{d}\mathbf{x}_{k-2}, \tag{38}$$

where $p(\mathbf{x}_{k-2}|\boldsymbol{\zeta}_{0:k-1})$ is the estimated state distribution at the previous time step $k-1$, and $p(\mathbf{x}_{k-1}|\mathbf{x}_{k-2}) = \mathbf{f}(\mathbf{x}_{k-2}, \mathbf{u}_{k-2})$ is the state transition density given by the GP. This integral will generally results a non-Gaussian distribution, and in practice a Gaussian distribution can be used as an approximate through moment matching method [14], [11].

Then the update step can be formulated using Bayes' rule

$$p(\mathbf{x}_{k-1}|\boldsymbol{\zeta}_{0:k}) = \frac{p(\boldsymbol{\zeta}_k|\mathbf{x}_{k-1}) p(\mathbf{x}_{k-1}|\boldsymbol{\zeta}_{0:k-1})}{\int p(\boldsymbol{\zeta}_k|\mathbf{x}_{k-1}) p(\mathbf{x}_{k-1}|\boldsymbol{\zeta}_{0:k-1}) \mathrm{d}\mathbf{x}_{k-1}}, \tag{39}$$

where $p(\boldsymbol{\zeta}_k|\mathbf{x}_{k-1}) = \mathbf{f}_{\zeta}(\mathbf{x}_{k-1}, \mathbf{u}_{k-1})$ is the likelihood term given by the GP, and $p(\mathbf{x}_{k-1}|\boldsymbol{\zeta}_{0:k-1})$ is the prior given by the prediction step (38). Again, the product of the two densities in the numerator is not Gaussian in general and a Gaussian approximation through moment-matching could be used.

Given an initial condition

$$p(\mathbf{x}_0|\boldsymbol{\zeta}_0) = p(\mathbf{x}_0) = \mathcal{N}(\boldsymbol{\mu}_{\mathbf{x}_0}, \boldsymbol{\Sigma}_{\mathbf{x}_0}), \tag{40}$$

the computation of (38) and (39) can be carried out iteratively to form a recursive estimation procedure, and in the process, the following MAP estimate of the state could also be computed

$$\hat{\mathbf{x}}_{k-1,\text{MAP}} = \arg\max_{\mathbf{x}_{k-1}} p(\boldsymbol{\zeta}_k|\mathbf{x}_{k-1}) p(\mathbf{x}_{k-1}|\boldsymbol{\zeta}_{0:k-1}). \tag{41}$$

An algorithmic description of the above filtering method is listed in Algorithm 1.

Algorithm 1 Nonlinear Dynamic Systems State Estimation with Gaussian Processes and Local Linear Models

Initialisation of parameters
Set the initial time instant $k = 0$
Train the $\mathcal{GP}$ with $\mathbf{Z}_{Ld}, \mathbf{Y}_{Ld}, \mathbf{Z}_L$, $\mathbf{Y}_L$, and additional input-output data
Set the initial distribution of $\mathbf{x}_0$ as in (40)
while $k < k_T$, k_T is the terminating time instant **do**
    Obtain $\mathbf{u}_k$ and $\boldsymbol{\zeta}_k$
    if $k \geq 1$ **then**
        if $k > 1$ **then**
            Compute the prediction step (38)
        end if
        Compute the update step (39)
        Compute the MAP estimate (41)
    end if
    $k = k + 1$
end while

IV. Case Study: State Estimation of Aircraft Longitudinal Dynamics

A. Aircraft Description

In this section, a preliminary case study is carried out on a high-fidelity nonlinear aircraft simulation model from NASA called the generic transportation model (GTM) [15]. The longitudinal dynamics of the aircraft model is investigated in this case study, which includes 4 states and 1 input. The states consist of the true airspeed V_{TAS}, the angle of attack (AoA) α, the pitch rate q, and the pitch angle θ. The left and the right elevators are bound together to form a single elevator input δ_e. Notations α and θ are reused here to follow the flight dynamics modelling convention.

To describe the aircraft longitudinal dynamics, a GP state-space model is constructed as in (24). The model has 4 single-output GPs, each of which has 5 inputs. Two local linear models are pre-computed at straight and level trimmed flight at an altitude of 800 feet. The first one is linearised at an airspeed of 75 knots, with a trimmed angle of attack of 5.6 degrees. The other is linearised at an airspeed of 52.8 knots, with a trimmed angle of attack of 17 degrees. 10 pairs of noisy input-output data around the first trim point are also provided.

B. The Simulation Scenario and Results

In this testing scenario, the angle of attack of the aircraft is assumed to be unknown and measurements of all other longitudinal states are available. Then, by using the notations in (27) and (28), we have $\boldsymbol{\zeta} = [V_{\text{TAS}} \quad q \quad \theta]^\top$ and $\boldsymbol{\rho} = [\alpha]$. The units of the variables in $\boldsymbol{\zeta}$ are knots, radians per second, and radians, respectively, and the unit of α is degrees. Measurement of $\boldsymbol{\zeta}$ is contaminated by zero-mean white Gaussian noise with a covariance matrix of $\text{diag}([1\ 0.001\ 0.001])$.

The simulated aircraft starts at straight and level trimmed flight at an altitude of 800 feet with an airspeed of 75 knots, the same as the trimming condition for the first local linear model. At 0.5 seconds, a doublet command with a magnitude of 5 degrees lasting for 1 second is issued to the elevator. The elevator command is shown in Fig. 3.

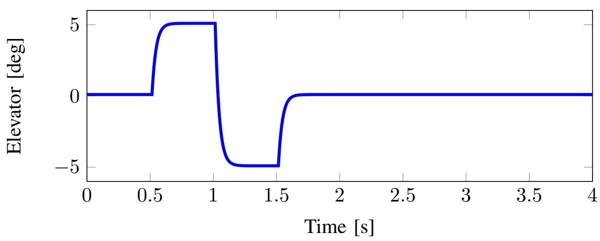

Fig. 3: The doublet elevator command.

In order to demonstrate the performance of the proposed method and the advantages of incorporating local linear models in GP, three estimators are compared. They include a Kalman filter designed at the first trim point, an MAP estimator as in (41) using both the local linear models and the input-output data points, and an MAP estimator as in (41) using only the 10 input-output data points.

For the Kalman filter, the covariance matrix of the process noise is chosen to be $10^{-6}\mathbf{I}_{4\times4}$, since there are no process noise involved in the simulation. The output covariance matrix is chosen to be the actual covariance matrix of the measurement noise. For the GP models, the information of the measurement noise is automatically computed from the 10 pairs of noisy data during the training procedure.

The comparison of the angle of attack estimates is shown in Fig. 4.

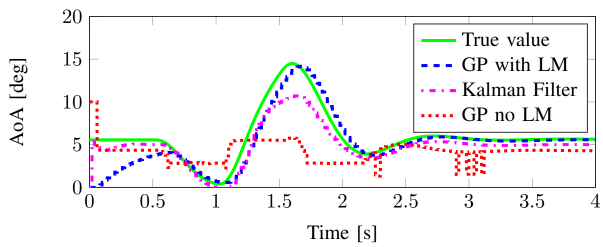

Fig. 4: A comparison of the angle of attack estimates by various estimators with the true values.

It can be seen that without the local linear models, the 10 input-output data points give the GP some rough information about the aircraft's input-output behaviour, and the estimates of the angle of attack wiggle around the equilibrium with pretty large error. In contrast, with the information from the local models, the performance of the GP-based estimator is significantly improved, and the estimates closely follow the true values of the angle of attack.

Furthermore, the Kalman filter also gives fairly good estimates when the angle of attack is around the equilibrium at which the filter is derived, but the estimates gradually deviate from the true values as the angle of attack gets larger and the aircraft's behaviour changes significantly from the equilibrium. In contrast, with two local linear models, one covers a relatively small angle of attack and the other a larger one, the GP-based filter gives a pretty good estimates over the entire range of the angle of attack.

Comparisons of the estimates of other states are shown in Fig. 5 to 7. It can be seen that the additive noise is largely filtered out from the measurements by the GP-based filter.

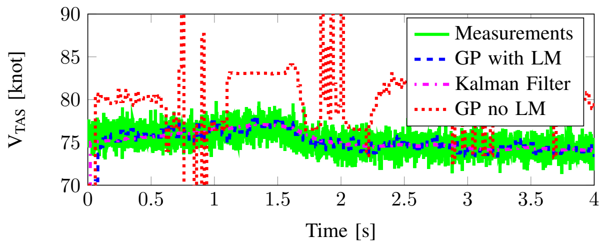

Fig. 5: A comparison of the airspeed estimates by various estimators with the actual measurements.

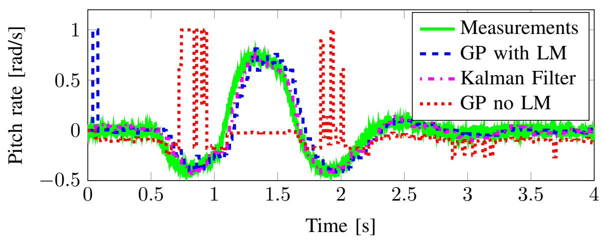

Fig. 6: A comparison of the pitch rate estimates by various estimators with the actual measurements.

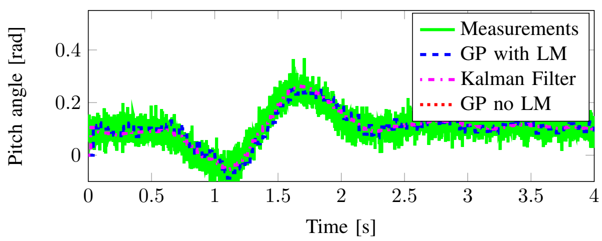

Fig. 7: A comparison of the pitch angle estimates by various estimators with the actual measurements.

V. Conclusions and Further Research

This paper proposes a state estimation method for nonlinear dynamic systems based on Gaussian processes and pre-computed local linear models. Since a linear model is an effective and compact form to describe the local behaviours of nonlinear dynamic systems, using local linear models reduces the amount of data required for GP and improves the performance of the GP-based filters. Simulations on an aircraft benchmark model show that with a couple of linearisations and a few input-output data points, the GP-based filter is capable of estimating the unknown aircraft state. This method is especially useful in a control systems design context in which local linearisations are usually readily available.

We also bear in mind that there is another class of methods in literature focusing on the simultaneous inference (i.e. state estimation) and learning (i.e. identification of the GP model itself) problem, which is far more challenging to tackle. For example, [16] used a two-stage maximum a posteriori (MAP) method over a proposed Gaussian process dynamical model for the estimation of the hyper-parameters and the latent states. The implementation was on a challenging task of human pose and motion estimation from high-dimensional motion capture data. [17] used an expectation maximisation algorithm to iterate between the inference and the learning procedures. [18] proposed a specific particle Markov Chain Monte Carlo (MCMC) method and [19] uses a variational approach to reduce the computational load of [18]. As stated in the introduction, the method in this paper assumes that the GP model is trained with offline data and no online learning, i.e. adaptation of the hyper-parameters according to the online data, is involved. Since the purpose of incorporating derivative data is to improve the model quality, it is reasonable not to involve online adaptation of the hyper-parameters if the model quality can be improved sufficiently using the offline data. However, online simultaneous estimation and learning with pre-computed local linear models is still worth doing from a technical point of view.

VI. Acknowledgement

The authors thank Jan Maciejowski and Edward Hartley for their invaluable comments on the ideas in this paper.

References

[1] B. Anderson and J. Moore, *Optimal Filtering*. Prentice-Hall, Inc., 1979.

[2] S. J. Julier and J. K. Uhlmann, "Unscented filtering and nonlinear estimation," *Proceedings of the IEEE*, vol. 92, pp. 401–422, Mar 2004.

[3] N. J. Gordon, D. J. Salmond, and A. F. M. Smith, "Novel approach to nonlinear/non-Gaussian Bayesian state estimation," *IEE Proceedings F - Radar and Signal Processing*, vol. 140, pp. 107–113, April 1993.

[4] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. the MIT Press, 2006.

[5] M. P. Deisenroth, R. D. Turner, M. F. Huber, U. D. Hanebeck, and C. E. Rasmussen, "Robust filtering and smoothing with Gaussian processes," *IEEE Transactions on Automatic Control*, vol. 57, pp. 1865–1871, July 2012.

[6] J. Ko and D. Fox, "GP-BayesFilters: Bayesian filtering using Gaussian process prediction and observation models," *Autonomous Robots*, vol. 27, no. 1, pp. 75–90, 2009.

[7] M. P. Deisenroth, M. F. Huber, and U. D. Hanebeck, "Analytic moment-based Gaussian process filtering," in *Proceedings of the 26th Annual International Conference on Machine Learning*, (Montreal, Quebec, Canada), pp. 225–232, 2009.

[8] M. Deisenroth and S. Mohamed, "Expectation propagation in Gaussian process dynamical systems," in *Advances in Neural Information Processing Systems 25*, pp. 2609–2617, 2012.

[9] E. Solak, R. Murray-Smith, W. E. Leithead, D. J. Leith, and C. E. Rasmussen, "Derivative observations in Gaussian process models of dynamic systems," in *Advances in Neural Information Processing Systems 15*, (Vancouver, British Columbia, Canada), 2002.

[10] J. Kocijan and D. J. Leigth, "Derivative observations used in predictive control," in *Proceedings of the 12th IEEE Mediterranean Electrotechnical Conference*, vol. 1, (Boston, MA, USA), pp. 379–382, 2004.

[11] K. Ažman and J. Kocijan, "Non-linear model predictive control for models with local information and uncertainties," *Transactions of the Institute of Measurement and Control*, vol. 30, no. 5, pp. 371–396, 2008.

[12] D. J. C. MacKay, "Introduction to Gaussian processes," in *NIPS'97 Tutorial notes*, 1998.

[13] M. P. Deisenroth, *Efficient Reinforcement Learning Using Gaussian Processes*. KIT Scientific Publishing, 2010.

[14] J. Quiñonero-Candela, A. Girard, J. Larsen, and C. E. Rasmussen, "Propagation of uncertainty in Bayesian kernel models - application to multiple-step ahead forecasting," in *Proceedings of the 2003 IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 2, pp. II–701–4, 2003.

[15] K. Cunningham, D. Cox, D. Murri, and S. Riddick, "A piloted evaluation of damage accommodating flight control using a remotely piloted vehicle," in *Proceedings of the 2011 AIAA Guidance, Navigation, and Control Conference*, (Portland, Oregon, USA), Aug 2011.

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

[17] R. D. Turner, *Gaussian Processes for State Space Models and Change Point Detection*. PhD thesis, University of Cambridge, July 2011.

[18] R. Frigola, F. Lindsten, T. B. Schön, and C. E. Rasmussen, "Bayesian inference and learning in Gaussian process state-space models with particle MCMC," in *Advances in Neural Information Processing Systems 26*, (Nevada, USA), 2013.

[19] R. Frigola, Y. Chen, and C. E. Rasmussen, "Variational Gaussian process state-space models," in *Advances in Neural Information Processing Systems 27*, pp. 3680–3688, 2014.