Journal of Physics: Conference Series **659** (2015) 012006 doi:10.1088/1742-6596/659/1/012006

Functional Dual Adaptive Control with Recursive Gaussian Process Model

Jakub Prüher and Ladislav Král
University of West Bohemia, Pilsen, Czech Republic
E-mail: jacobnzw@ntis.zcu.cz, ladkral@ntis.zcu.cz

Abstract. The paper deals with dual adaptive control problem, where the functional uncertainties in the system description are modelled by a non-parametric Gaussian process regression model. Current approaches to adaptive control based on Gaussian process models are severely limited in their practical applicability, because the model is re-adjusted using all the currently available data, which keeps growing with every time step. We propose the use of recursive Gaussian process regression algorithm for significant reduction in computational requirements, thus bringing the Gaussian process-based adaptive controllers closer to their practical applicability. In this work, we design a bi-criterial dual controller based on recursive Gaussian process model for discrete-time stochastic dynamic systems given in an affine-in-control form. Using Monte Carlo simulations, we show that the proposed controller achieves comparable performance with the full Gaussian process-based controller in terms of control quality while keeping the computational demands bounded.

1. Introduction

The problem of adaptive control of non-linear stochastic systems with unknown functions offers an interesting challenge [1–4]. It can be understood as a natural extension of the adaptive control from a class of linear systems and non-linear systems with unknown parameters to the complex systems with functional uncertainty. Sometimes this attractive direction of adaptive control is also called a functional adaptive control [1].

Since an optimal solution to this problem is practically impossible to find, a considerable attention has been focused on various sub-optimal solutions, which preserve a key aspects of the stochastic control principles originated from the work of Feldbaum [5]. It means, that a controller should respect the following conflicting goals:

- achieve tracking performance - a controller should be cautious and respects the uncertainty in model knowledge,
- reduce the uncertainty in the future - a controller should probe the system by an appropriate exciting signal.

and is referred to as a dual control [5, 6].

In recent years, several sub-optimal dual controller methods have been applied successfully in the functional approach for their positive qualities (superior control quality and admissible computational demands) and subsequently extended in several directions [7–11]. Although a partial progress was achieved in a functional adaptive control, one of the open and challenging issues remains; namely, finding of a suitable model of the controlled non-linear system. A common

feature of most of the above mentioned solutions to the functional approach is that they are based on parametric models represented by various types of neural networks. These models bring difficulties such as the need to optimise the model structure or the necessity to solve a non-linear estimation problem.

Recently, in the system identification community, non-parametric Gaussian process (GP) regression models have attracted a marked interest as an alternative tool to the parametric models. A GP can be effectively thought of as a prior distribution over functions themselves, where functions can be informally regarded as infinite dimensional vectors of function values. Gaussian process (GP) models do not possess a fixed structure and thus are very flexible. They allow to combine tractable Bayesian inference with non-parametric nature of the model and are highly effective in modelling strongly non-linear phenomena. GP models have been utilised in many applications including non-linear filtering [12–15], model predictive control [16], non-linear system identification [17], time series forecasting [18], reinforcement learning [19, 20] or numerical quadrature [21]. Great advantage of parametric models is that they lend themselves nicely to the possibility of devising a recursive estimation algorithm. As GPs are non-parametric models, the development of a recursive algorithm is much more complicated. Typically, it is necessary to consider some kind of GP approximation to counteract the problem of increasing data size. Recently increased effort has been given to the design of original algorithms, which eliminate this drawback [22–26].

So far, GP modelling was utilised in functional adaptive approach only in a few pioneering works [3, 27]. It should be noted, that the GP models were implemented exclusively in a non-recursive form, causing a continuous increase of the computational demands with time, which significantly reduces their practical applicability. In another words, an effective dual controller based on GP model has not been realised yet.

Based on the motivation point, the main goal of the paper is twofold: 1) Design a functional adaptive control using the recursive GP model suggested in [28] and 2) Compare the proposed solution with the non-recursive case presented in [27] through extensive Monte-Carlo analysis in a numerical example.

The rest of the paper will present a description of the problem formulation in section 2, followed by a recursive GP model based non-linear system identification in section 3. In section 4, dual adaptive control design is completed. A numerical illustration is presented in section 5 and finally, section 6 concludes the paper.

2. Problem Formulation

The dynamical system to be controlled is a non-linear stochastic discrete time-invariant system given in an input-output representation

$$\mathcal{S}: \qquad y_{k+1} = f(\boldsymbol{x}_k^a) \; + \; g(\boldsymbol{x}_k^a)u_k \; + \; e_{k+1}, \tag{1}$$

where $f, g : \mathbb{R}^{n_y+n_u+1} \to \mathbb{R}$ are unknown non-linear functions, $\boldsymbol{x}_k^a = [y_k, \ldots, y_{k-n_y}, u_{k-1}, \ldots, u_{k-n_u}]^\top \in \mathbb{R}^{n_y+n_u+1}$ is the state vector, u_k and y_k are input and output signals at discrete time instants $k \in 0, 1, \ldots,$ N-1 and $\{e_k\}$ is an additive noise and the following assumptions A1-A4 are considered:

A1: The non-linear functions are smooth (infinitely differentiable), i.e. $f(\boldsymbol{x}_k^a)$, $g(\boldsymbol{x}_k^a) \in \mathcal{C}^\infty$.

A2: The structural parameters n_y and n_u of the system are known.

A3: The system has a globally uniformly asymptotically stable zero dynamics and the non-linear function $g(\boldsymbol{x}_k^a)$ is bounded away from zero for all $\boldsymbol{x}_k^a$ [29].

A4: $\{e_k\} \in \mathbb{R}$ is a known Gaussian sequence with zero mean and variance σ_e^2.

The proposed control law is of an explicit-type, suboptimal dual cost function based on the bi-criterial dual (BD) controller developed by Filatov et al. [30] for linear systems. This cost function explicitly includes two separate criteria, where each criterion introduces one of the mutually opposing goals between estimation and control; *probing* and *caution*. The final control law will be obtained by a subsequent minimisation of the criteria (2) and (4).

The first criterion in the bi-criterial approach is suggested in the following form

$$J_k^c = \mathrm{E}\left\{(y_{k+1} - r_{k+1})^2 + q\, u_k^2 \middle| \boldsymbol{I}^k\right\}, \tag{2}$$

where $\boldsymbol{I}^k$ is the information state containing all measurable inputs and outputs available up to time instant k and r_{k+1} is a bounded reference signal and $q \geq 0$ is a design parameter penalising the control action.

The criterion (2) evaluates quality of the control and involves minimisation of the expected value of the tracking error. The resulting control

$$u_k^c = \underset{\mathrm{u_k}}{\operatorname{argmin}}\, J_k^c \tag{3}$$

respects uncertainties in the knowledge of the unknown system functions (1), and is equal to *cautious control*.

The second criterion in the bi-criterial approach is chosen as

$$J_k^a = -\mathrm{E}\left\{(y_{k+1} - \hat{y}_{k+1})^2 \,\middle|\, \boldsymbol{I}^k\right\}, \tag{4}$$

where $\hat{y}_{k+1}$ is a one step prediction of the output, which will be obtained from the GP model. This criterion evaluates the quality. It should accelerate the estimation process for future control improvement by increasing the predictive error value. The controller provides an optimal excitation added to the cautious control and determines magnitude of the *probing signal*.

Finally, the BD control u_k is then obtained by

$$u_k = \underset{\mathrm{u_k}\in\Omega}{\operatorname{argmin}}\, J_k^a, \tag{5}$$

where Ω defines a Pareto set as a region of permissible values of u_k representing an efficient trade-off between criteria J_k^a and J_k^c. The region Ω is a design parameter. The J_k^c contribution should be proportional to the uncertainty of the model to provide a probing component of the control signal u_k rich enough.

From relationships describing adaptive controller (2) – (5) it is clear that a suitable model description is necessary to complete the control design. The system model will be based on a non-parametric GP regression and control design will be performed as follows. First, a GP model of the system (1) will be specified. Then a recursive algorithm for the GP model will be proposed. Subsequently, the recursive GP model will be used to complete derivation of the control law. Finally, the whole procedure will be algorithmically summarised.

3. Gaussian Process Regression Model

This section covers the identification of a non-linear stochastic system (1) by means of the GP regression algorithm. The full Gaussian process (FGP) model is described first, in order to show its main difficulties, when applied in situations where data arrive sequentially (which is exactly the case in adaptive control). Introducing the non-recursive FGP model also serves as a motivation for the recursive GP regression algorithm proposed in the subsequent part.

3.1. Full Gaussian process model

The GP can be seen as an extension of the multivariate normal distribution to the normal distribution over infinite-dimensional random vectors (random functions). This means, that the GP, akin to the Gaussian distribution, is fully determined by its first two moments; mean and variance. Let $\boldsymbol{x}_k = [\,(\boldsymbol{x}_k^a)^\top,\, u_k\,]^\top$ be a vector of dimension $D = ny + nu + 2$ and let $\boldsymbol{y}_k = [y_1, \ldots, y_k]^\top$, $\boldsymbol{u}_k = [\,u_0,\, \ldots,\, u_{k-1}\,]$ and $\boldsymbol{X}_k = [\,\boldsymbol{x}_0,\, , \ldots,\, \boldsymbol{x}_{k-1}]$ be a $D \times k$ matrix. Then, a non-parametric GP model of the unknown non-linear stochastic system (1) can be defined in the following general form

$$\mathcal{M}: \qquad \hat{y}_{k+1} = \mathcal{GP}(m^+,\, k^+), \tag{6}$$

where posterior mean m^+ and variance k^+ fully describe the GP posterior $\mathcal{GP}\,(m^+,\, k^+)$ and are given by

$$m^+ = m(\boldsymbol{x}_k\,;\boldsymbol{\theta}) \;+\; k(\boldsymbol{x}_k,\, \boldsymbol{X}_k\,;\boldsymbol{\theta})\, k(\boldsymbol{X}_k,\, \boldsymbol{X}_k\,;\boldsymbol{\theta})^{-1}\boldsymbol{y}_k, \tag{7}$$

$$k^+ = k(\boldsymbol{x}_k,\, \boldsymbol{x}_k\,;\boldsymbol{\theta}) \;-\; k(\boldsymbol{x}_k,\, \boldsymbol{X}_k\,;\boldsymbol{\theta})\, k(\boldsymbol{X}_k,\, \boldsymbol{X}_k\,;\boldsymbol{\theta})^{-1} k(\boldsymbol{X}_k,\, \boldsymbol{x}_k\,;\boldsymbol{\theta}), \tag{8}$$

where $m(\boldsymbol{x}_k \mid \boldsymbol{\theta}) : \mathbb{R}^D \to \mathbb{R}$ is a *mean function* and $k(\boldsymbol{x}_k,\, \boldsymbol{x}_k \mid \boldsymbol{\theta}) : \mathbb{R}^D \times \mathbb{R}^D \to \mathbb{R}$ is a *covariance function*. The mean and covariance function are parametrised by a vector of free *hyper-parameters* $\boldsymbol{\theta}$. The setting of the optimal values of $\boldsymbol{\theta}$ is discussed in Section 3.2.2 and the dependence of $m(\cdot)$ and $k(\cdot,\cdot)$ on $\boldsymbol{\theta}$ will be omitted for a brevity of notation. Moreover, it holds that $k(\boldsymbol{x}_k,\, \boldsymbol{X}_k) = [\,k(\boldsymbol{x}_k,\, \boldsymbol{x}_0),\, \ldots,\, k(\boldsymbol{x}_k,\, \boldsymbol{x}_{k-1})\,]$, $k(\boldsymbol{X}_k,\, \boldsymbol{x}_k) = k(\boldsymbol{x}_k,\, \boldsymbol{X}_k)^\top$ and $k(\boldsymbol{X}_k,\, \boldsymbol{X}_k)$ is an $k \times k$ prior covariance matrix of elements $k(\boldsymbol{x}_i,\, \boldsymbol{x}_j)$ for $i, j \in 0, \ldots, N-1$.

The mean and covariance function are chosen by the designer and imposes certain assumptions on the model. Compared to the parametric regression models, where the assumptions on the modelled function are typically expressed in a fixed model structure, the assumptions of non-parametric GP models, expressed by the choice of covariance function, are much weaker. There are many covariance functions to choose from, each expressing different assumptions about the modelled phenomena. For comprehensive account of covariance functions and their relationship to other models see [31].

Without loss of generality, zero mean function $m(\cdot)$ will be further assumed. The covariance function $k(\cdot,\cdot)$ is chosen to ease the design of the adaptive BD controller in later sections and is inspired by [32]. The mean and covariance functions are thus given by

$$m(\boldsymbol{x}_k) \equiv 0, \tag{9}$$

$$k(\boldsymbol{x}_i,\, \boldsymbol{x}_j) = k_f(\boldsymbol{x}_i^a,\, \boldsymbol{x}_j^a) \;+\; k_g(\boldsymbol{x}_i^a,\, \boldsymbol{x}_j^a) u_i u_j \;+\; k_n(\boldsymbol{x}_i^a,\, \boldsymbol{x}_j^a), \tag{10}$$

where

$$k_f(\boldsymbol{x}_i^a,\, \boldsymbol{x}_j^a) = \ell_f^2 \exp\left(-\frac{1}{2}(\boldsymbol{x}_i^a - \boldsymbol{x}_j^a)^\top \boldsymbol{\Lambda}_f^{-1}(\boldsymbol{x}_i^a - \boldsymbol{x}_j^a)\right), \tag{11}$$

$$k_g(\boldsymbol{x}_i^a,\, \boldsymbol{x}_j^a) = \ell_g^2 \exp\left(-\frac{1}{2}(\boldsymbol{x}_i^a - \boldsymbol{x}_j^a)^\top \boldsymbol{\Lambda}_g^{-1}(\boldsymbol{x}_i^a - \boldsymbol{x}_j^a)\right), \tag{12}$$

$$k_n(\boldsymbol{x}_i,\, \boldsymbol{x}_j) = \sigma_e^2\, \delta_{ij}, \tag{13}$$

and where ℓ_f, ℓ_g are vertical lengthscales, $\boldsymbol{\Lambda}_f = \mathrm{diag}(\lambda_{f1}^2, \ldots, \lambda_{fD}^2)$ (analogously for $\boldsymbol{\Lambda}_g$) are horizontal lengthscales and σ_e^2 is a noise variance. The symbol δ_{ij} is Kronecker delta. The above mentioned quantities are hyper-parameters forming a vector $\boldsymbol{\theta} = [\,\ell_f^2,\, \lambda_{f1},\, \ldots,\, \lambda_{fD},\, \ell_g^2,\, \lambda_{g1},\, \ldots,\, \lambda_{gD}\,]^\top$. The term k_n is a covariance function of a white noise with variance σ_e^2 and the covariance functions k_f and k_g are selected as *squared exponential* covariance function with *automatic relevance determination*, which is a common choice in a number of applications [14, 17].

The equations (6)–(13) describe model of the system (1) and provide a procedure how to obtain an one-step prediction $\hat{y}_{k+1}$ and variance $\mathrm{var}(y_{k+1})$, which may be useful component in the control law derivation [27]. Note that, the equations (7)–(8) show that the prediction is calculated using *all* the currently available data. This is the distinguishing feature of non-parametric models in general. In offline settings, where all the data are readily available, calculating model prediction calls for the inversion of large covariance matrix $k(\boldsymbol{X}_k^a, \boldsymbol{X}_k^a)$. In cases where the data arrives sequentially, the situation is even worse, because the size of the covariance matrix $k(\boldsymbol{X}_k, \boldsymbol{X}_k)$ grows with every new measurement, leading to ever increasing computational demands for matrix inversion. This is why controllers based on this non-recursive model form cannot be seriously considered for practical applications. To alleviate this problem, the recursive Gaussian process regression algorithm is proposed in the next section.

3.2. Recursive Gaussian process model

The proposed recursive Gaussian process algorithm is described in two steps. The recursive relations for mean and variance functions calculation is briefly outlined first. This is then followed by a procedure for hyper-parameter learning. To keep the presentation succinct, only the basic results obtained in [25] and [28] are described. For detailed derivation of the algorithm, interested reader is referred to the previously cited references.

3.2.1. RGP algorithm Huber [25] proposed the recursive Gaussian process (RGP) regression algorithm as a solution to the problem of GP regression for sequential data processing. The main idea employed in RGP is to use a predefined set of *basis vectors*, which are sequentially updated and effectively summarise the information obtained from currently available measurements about the approximated function on a user-defined domain. Instead of using all the data for prediction, as in the case of FGP, predictions of RGP model are computed from the basis vectors. The number of basis vectors $s \ll N$ (where N is the number of maximum data points) is fixed throughout the operation of the algorithm, which enables to keep the computational demands in check. On the other hand, using only fixed amount of basis vectors acts as an approximation to the full GP approach characterised by the equations (7) and (8), because the prediction is no longer a function of all the previously seen data.

Let $\tilde{\boldsymbol{X}} = [\tilde{\boldsymbol{x}}_1, \tilde{\boldsymbol{x}}_2, \ldots, \tilde{\boldsymbol{x}}_s]$ denote a $D \times s$ matrix of the basis vectors. The RGP algorithm then calculates moments m^+ and k^+ at every time step $k > 0$ by incorporating information from the new observation y_k. In summary, the RGP algorithm operates by means of the following two sets of equations

$$\text{update} \begin{cases} \boldsymbol{G}_k &= \boldsymbol{C}_{k-1}^h \boldsymbol{J}_{k-1}^\top \left(C_k^p + \sigma_e^2 \right)^{-1}, & \text{(14a)} \\ \boldsymbol{\mu}_k^h &= \boldsymbol{\mu}_{k-1}^h + \boldsymbol{G}_k \left(y_k - \mu_k^p \right), & \text{(14b)} \\ \boldsymbol{C}_k^h &= \boldsymbol{C}_{k-1}^h - \boldsymbol{G}_k \boldsymbol{J}_{k-1} \boldsymbol{C}_{k-1}^h, & \text{(14c)} \end{cases}$$

$$\text{predict} \begin{cases} \boldsymbol{J}_k &= \boldsymbol{k}_k^s \, \boldsymbol{K}^{-1}, & \text{(15a)} \\ \mu_{k+1}^p &= \boldsymbol{J}_k \, \boldsymbol{\mu}_k^h, & \text{(15b)} \\ C_{k+1}^p &= k_k^{ss} + \boldsymbol{J}_k \left(\boldsymbol{C}_k^h - \boldsymbol{K} \right) \boldsymbol{J}_k^\top, & \text{(15c)} \end{cases}$$

where $\boldsymbol{K} = k(\tilde{\boldsymbol{X}}, \tilde{\boldsymbol{X}})$ is an $s \times s$ matrix of elements $k(\tilde{\boldsymbol{x}}_p, \tilde{\boldsymbol{x}}_q)$ for $p, q \in 1, \ldots, s$. The quantities $\boldsymbol{k}_k^s$ and k_k^{ss} are computed as

$$\boldsymbol{k}_k^s = k(\boldsymbol{x}_k, \tilde{\boldsymbol{X}}) = \boldsymbol{k}_k^{sf} + \boldsymbol{k}_k^{sg} \, u_k, \quad (16)$$

$$k_k^{ss} = k(\boldsymbol{x}_k, \boldsymbol{x}_k) = k_k^{ssf} + k_k^{ssg} \, u_k^2 + k_k^{ssn} \quad (17)$$

with

$$\boldsymbol{k}_k^{sf} = k_f(\boldsymbol{x}_k^a,\, \tilde{\boldsymbol{X}}^a), \quad (18) \qquad k_k^{ssf} = k_f(\boldsymbol{x}_k^a,\, \boldsymbol{x}_k^a), \quad (20)$$

$$\boldsymbol{k}_k^{sg} = k_g(\boldsymbol{x}_k^a,\, \tilde{\boldsymbol{X}}^a) \circ \boldsymbol{u}, \quad (19) \qquad k_k^{ssg} = k_g(\boldsymbol{x}_k^a,\, \boldsymbol{x}_k^a), \quad (21)$$

$$k_k^{ssn} = k_n(\boldsymbol{x}_k^a,\, \boldsymbol{x}_k^a), \quad (22)$$

where the symbol $\circ$ denotes element-wise product of two vectors, $\tilde{\boldsymbol{X}}^a$ is the matrix of $D-1$ rows of $\tilde{\boldsymbol{X}}$ and $\boldsymbol{u}$ denotes the last row of $\tilde{\boldsymbol{X}}$. The equations (14a) – (14c) update the function value estimates $\boldsymbol{\mu}_k^h$ and their covariance $\boldsymbol{C}_k^h$ using the last output measurement y_k. The equations (15a) – (15c) are used to calculate the RGP model output prediction $\hat{y}_{k+1} = \mu_{k+1}^p$ and predictive variance $\mathrm{var}(y_{k+1}) = C_{k+1}^p$.

3.2.2. Hyper-parameter learning procedure The RGP algorithm is able to reduce the computational demands for GP regression in sequential data processing. Note however, that in order to compute the RGP model prediction given by equations (15a) – (15c), the hyper-parameters $\boldsymbol{\theta}$ have to be determined.

A preferred method for setting of the hyper-parameters is based on maximisation of the marginal likelihood. In practice it is convenient to work with the logarithm of marginal likelihood, which for a Gaussian likelihood, has an analytic form

$$\log p(\boldsymbol{y}_k \mid \boldsymbol{X}_k, \boldsymbol{\theta}) = -\frac{1}{2}\underbrace{\boldsymbol{y}_k^\top \left(\boldsymbol{K}(\boldsymbol{\theta}) + \sigma_e^2 \boldsymbol{I}\right)^{-1} \boldsymbol{y}_k}_{\text{data fit}} - \frac{1}{2}\underbrace{\log \left|\boldsymbol{K}(\boldsymbol{\theta}) + \sigma_e^2 \boldsymbol{I}\right|}_{\text{complexity penalty}} - \frac{N}{2}\log 2\pi, \quad (23)$$

where $\boldsymbol{K}(\boldsymbol{\theta}) = k(\boldsymbol{X}_k,\, \boldsymbol{X}_k;\, \boldsymbol{\theta})$. The marginal likelihood consists of a *data fit* term, which encourages complex models that fit data well, and a *complexity penalty* term, which penalises models that are too complex. This ensures that maximisation of (23) will not result in model over-fitting or under-fitting [33]. The disadvantage is that the optimisation problem is non-linear, which makes finding of the global maxima very difficult. It is helpful to point out, that the objective (23) is a function of $\boldsymbol{\theta}$, where the data remain fixed. Note, that using this technique the optimal hyper-parameters are determined directly from the data and therefore, their optimisation takes place before the system output prediction is generated.

Using (23) for setting of hyper-parameters in RGP algorithm is certainly possible, but note, that this would again introduce the problem of increasing computational demands as the covariance matrix $\boldsymbol{K}(\boldsymbol{\theta})$ in the marginal likelihood is constructed from all available data. Evaluating (23) would thus become harder with every new measurement processed.

We propose an approximation of (23) by replacing the noisy observations $\boldsymbol{y}_k$ with estimates of the function values $\boldsymbol{\mu}_k^h$ at the basis vectors $\tilde{\boldsymbol{X}}$. The approximate marginal likelihood is then given by

$$\log p(\boldsymbol{y}_k \mid \boldsymbol{X}_k, \boldsymbol{\theta}) \approx \log p\Big(\boldsymbol{\mu}_k^h \mid \tilde{\boldsymbol{X}}, \boldsymbol{\theta}\Big) = -\frac{1}{2}\Big[(\boldsymbol{\mu}_k^h)^\top \tilde{\boldsymbol{K}}(\boldsymbol{\theta})^{-1}\, \boldsymbol{\mu}_k^h \;+\; \log\Big|\tilde{\boldsymbol{K}}(\boldsymbol{\theta})\Big| \;+\; s\log(2\pi)\Big], \quad (24)$$

where $\tilde{\boldsymbol{K}}(\boldsymbol{\theta}) = k(\tilde{\boldsymbol{X}},\, \tilde{\boldsymbol{X}};\, \boldsymbol{\theta})$. Since the number of basis vectors s is pre-defined by the user and fixed throughout the operation of the algorithm, the computational demands for evaluating the approximated marginal likelihood (24) do not grow with time. In practical implementation, the objective (24) is maximised in every time step using an iterative optimisation solver, where the initial guess are the hyper-parameters from the previous time step. To keep the computational requirements in check, we also limit the maximum number of solver iterations.

4. Bi-criterial Dual Control Design

This section is focused on the BD control derivation based on the idea of a multi-objective optimization introduced in Section 2.

4.1. Control law derivation

Let $\hat{f}_{k+1} = \boldsymbol{k}_k^{sf}\, \boldsymbol{K}^{-1}\boldsymbol{\mu}_k^h$ and $\hat{g}_{k+1} = \boldsymbol{k}_k^{sg}\, \boldsymbol{K}^{-1}\boldsymbol{\mu}_k^h$. Then the GP model mean (15b) can be rewritten to the following form

$$\hat{y}_{k+1} = \hat{f}_{k+1} + \hat{g}_{k+1} u_k. \tag{25}$$

The GP model variance (15c) can be rearranged as

$$\operatorname{var}(y_{k+1}) = \mu_k u_k^2 + \nu_k u_k + c_k, \tag{26}$$

with $\boldsymbol{\Gamma} = \boldsymbol{K}^{-1}(\boldsymbol{C}_k^h - \boldsymbol{K})(\boldsymbol{K}^{-1})^\top$, $\mu_k = k_k^{ssg} + \boldsymbol{k}_k^{sg}\boldsymbol{\Gamma}(\boldsymbol{k}_k^{sg})^\top$, $\nu_k = \boldsymbol{k}_k^{sg}\boldsymbol{\Gamma}(\boldsymbol{k}_k^{sf})^\top + \boldsymbol{k}^{sf}\boldsymbol{\Gamma}(\boldsymbol{k}_k^{sg})^\top$ and $c_k = k_k^{ssn} + k_k^{ssf} + \boldsymbol{k}_k^{sf}\boldsymbol{\Gamma}(\boldsymbol{k}_k^{sf})^\top$. The first of the BD criteria (2) can be minimized using (25), (26) and the cautious control component u_k^c is given as

$$u_k^c = \frac{\left(r_{k+1} - \hat{f}_{k+1}\right)\hat{g}_{k+1} - \frac{1}{2}\nu_k}{\hat{g}_{k+1}^2 + \mu_k + q}. \tag{27}$$

The second BD criterion (4) can be rewritten by reusing (26) as

$$J_k^a = -\operatorname{var}(y_{k+1}) = -\mu_k u_k^2 - \nu_k u_k - c_k. \tag{28}$$

In order to complete the derivation we need to specify the region Ω. A suitable choice is a symmetrically distributed region around the caution control u_k^c defined as

$$\Omega_k = [u_k^c - \delta,\, u_k^c + \delta], \tag{29}$$

where the optional parameter δ fixes the domain Ω as a constant.

It is obvious, that the criterion J_k^a is a concave function of variable u_k. Hence, the extreme will inevitably be found within the boundary of the domain Ω. Therefore, it is possible to state the following relation

$$u_k = u_k^c + \eta \cdot \operatorname{sign}\left(J_k^a(u_k^c - \delta) - J_k^a(u_k^c + \delta)\right), \tag{30}$$

where $\eta > 0$ is a designer parameter, which represents the gain of the probing component. The expression in brackets can be expressed as

$$J_k^a(u_k^c - \delta) - J_k^a(u_k^c + \delta) = \delta(2\mu_k u_k^c + \nu_k). \tag{31}$$

The control law derivation is finalized using (31) in (30) as

$$u_k = u_k^c + \eta \cdot \operatorname{sign}\left(2\mu_k u_k^c + \nu_k\right). \tag{32}$$

The equations (27) and (32) represent the final BD adaptive control law, where the first term is the cautious control component and second term denotes probing control component. The probing part is a function of variable μ_k and ν_k which represent the uncertainty in the system functions $f(\boldsymbol{x}_k^a)$ and $g(\boldsymbol{x}_k^a)$. As a consequence, a higher value of the probing control component is generated in the case of high model uncertainty and vice versa.

4.2. BD-RGP control algorithm

The following few basic steps summarize the operation of the proposed BD-RGP adaptive controller

RGP-based Bicriterial Dual Control Algorithm	
Step 0:	Initialize GP model hyper-parameters $\boldsymbol{\theta} = \boldsymbol{\theta}_0$, basis vector locations $\tilde{\boldsymbol{X}}$ and set $\boldsymbol{\mu}_0^h \sim \text{Uniform}(0,1)$, $\boldsymbol{C}_0^h = k(\tilde{\boldsymbol{X}}, \tilde{\boldsymbol{X}}\,; \boldsymbol{\theta}_0)$; set system initial condition and controller parameters, set $k = 1$
Step 1:	Measure the system output y_k.
Step 2:	Update the RGP model $\boldsymbol{\mu}_k^h$, $\boldsymbol{C}_k^h$ using (14a) – (14c).
Step 3:	Optimise hyper-parameters as $\boldsymbol{\theta} = \arg\max_{\boldsymbol{\theta}} \log\left(p\left(\boldsymbol{\mu}_k^h \,|\, \tilde{\boldsymbol{X}}, \boldsymbol{\theta}\right)\right)$.
Step 4:	Generate control action u_k from relations (27) and (32).
Step 5:	Predict system output μ_{k+1}^p and predictive variance C_{k+1}^p by (15a)–(15c).
Step 6:	$k = k + 1$, go to Step 1.

5. Numerical Example

We tested the proposed bi-criterial dual controller on the following example of non-linear stochastic discrete-time system

$$y_{k+1} = \frac{1.5 y_k}{1 + y_k^2} + (2 + \cos(y_k))u_k + e_{k+1}, \tag{33}$$

where e_{k+1} is a white Gaussian noise with variance $\sigma_e^2 = 0.0025$. Note, that the vector of regressors $\boldsymbol{x}_k = [y_k, u_k]$ is now two-dimensional. The bi-criterial dual controller with the full Gaussian process model (BD-FGP) as well as the recursive GP model (BD-RGP) used the same set of initial hyperparameter values $\boldsymbol{\theta} = [0, 0, 0, 0]$. The set of basis vectors for the BD-RGP was initialized as a rectangular grid of 7×5 points in the interval $[-2, 2] \times [-1, 1]$.

The main motivation for designing the BD-RGP algorithm was to alleviate the prohibitive computational demands of the BD-FGP. The Figure 1 is a result of averaging execution time measured in each control loop iteration of both algorithms over 100 Monte Carlo runs. Clearly, the demands of the BD-RGP algorithm remain constant, thus making it superior alternative to the BD-FGP in this regard.

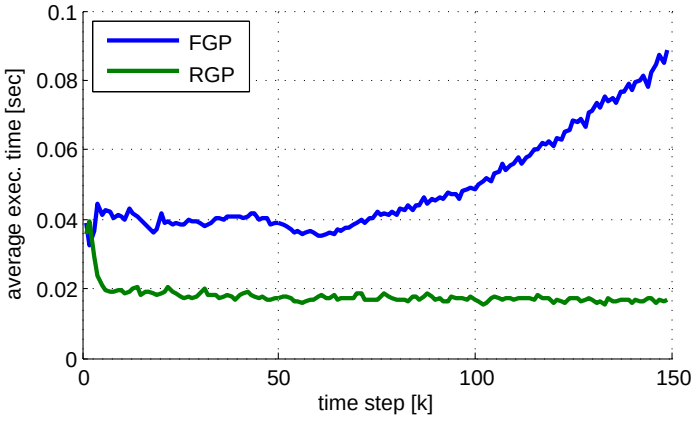

Figure 1: Average execution time per time step of the BD-FGP and BD-RGP control algorithms. The computational requirements of the BD-RGP controller remain constant, while the demands BD-FGP controller continue to grow indefinitely with every time step.

Table 1: Control quality of the bicriterial dual controller based on full (BD-FGP) and recursive Gaussian process (BD-RGP) model. Results are averaged over 100 Monte Carlo simulations. Variance estimates of the means of the criteria were obtained by bootstrap method.

Controller	$\hat{J}_1$	$\text{var}(\hat{J}_1)$	$\hat{J}_{15}$	$\text{var}(\hat{J}_{15})$
BD-FGP	1.08e-01	2.02e-04	1.12e-02	3.79e-05
BD-RGP	1.07e+01	9.66e+00	1.42e-02	1.01e-04

The other experiment was focused on the assessment of control quality, which was measured by

$$J_m = \frac{1}{N} \sum_{k=m}^{N} (y_{k+1} - r_{k+1})^2, \tag{34}$$

where $m \geq 1$ is the offset. We compared the control quality of the BD-FGP controller, which uses the exact non-recursive GP model, with the BD-RGP controller, which uses the approximate recursive GP model. Ideally, we would like the quality of the approximate BD-RGP to remain close enough to the exact BD-FGP. Each controller was simulated for $N = 60$ time steps and the reference signal r_k was chosen such that, a one half is a square wave and the other half is a sine wave.

Table 1 compares the control quality of the BD-FGP and the BD-RGP controllers for offsets $m = 1$, which evaluates the criterion using the whole trajectory, and $m = 15$, which ignores the first few steps to assess the behaviour after the GP model adaptation. The quality of the BD-RGP is two orders of magnitude worse than that of the BD-FGP, when the whole trajectory is taken into account. This is caused by the longer adaptation of the RGP model. However, the values of the offset criteria indicate that this adaptation quickly disappears (after about 15 time steps) and the BD-RGP control quality is then close to the BD-FGP. The reason for this behaviour is the fact, that the RGP model is an approximation of the exact FGP model, and as such takes longer to adapt (slower convergence). This intuition is corroborated by our earlier results published in [34].

Figure 2 compares the two dual controllers in terms of the system output response to the reference signal. The output response for the BD-RGP is more erratic at the start, which explains the results in the Table 1, but then stays close to the BD-FGP output response. Figure 3 shows the evolution of the FGP and the RGP model predictive uncertainty. The RGP model provides more conservative predictive variances across the whole trajectory, which is a desirable behaviour considering the RGP is an approximate model to the FGP.

6. Conclusions

In this work, we proposed the use of the RGP regression model for reducing computational demands of the BD controller based on full GP model, when applied for control of non-linear discrete time-invariant stochastic systems with functional uncertainties. The GP regression models are applied for the approximation of unknown functions in the system description. We utilize an approximation to the marginal likelihood, which makes it possible to keep the computational demands fixed when learning the GP model hyper-parameters. We derived the bi-criterial dual controller with RGP model, whose computational demands do not increase with time. After the model adaptation, the control quality of the proposed BD-RGP controller stays close to the BD-FGP controller.

Acknowledgments

This work was supported by the Czech Science Foundation, project no. GA 15-12068S.

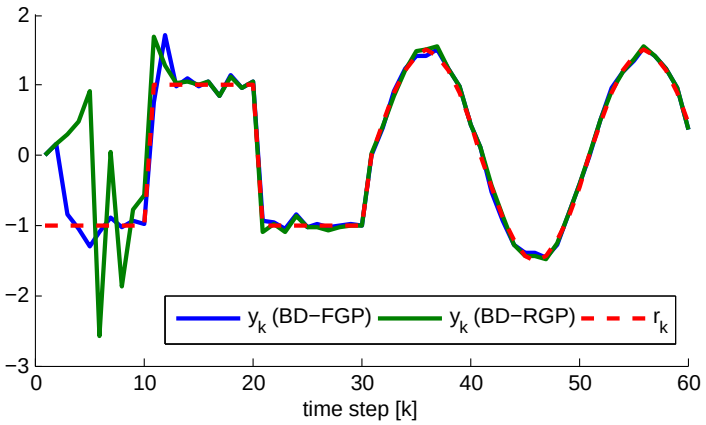

Figure 2: Comparison of the typical system output response y_{k+1} to a reference signal r_k for the BD-FGP and BD-RGP controllers. The controller using the approximate recursive GP model (BD-RGP) exhibits slower adaptation than the BD-FGP.

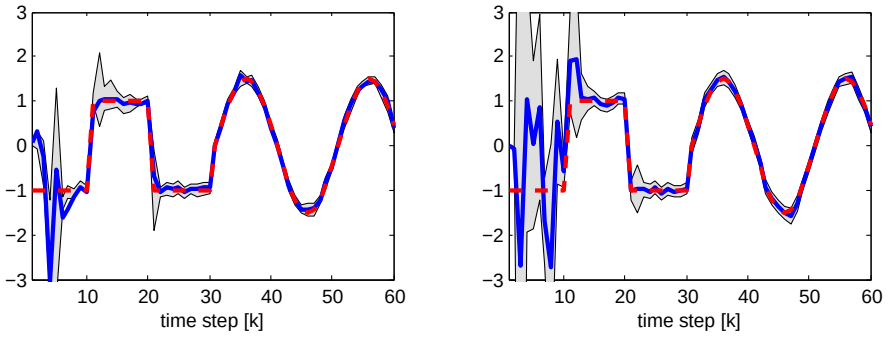

Figure 3: Comparison of model variances (gray band); the RGP model predictive variance (right) is more conservative than that of the FGP model (left). Since RGP model is an approximation of the FGP, this is a desirable behaviour.

References

[1] Fabri S G and Kadirkamanathan V 2001 *Functional Adaptive Control: An Intelligent Systems Approach* (Springer-Verlag)
[2] Herzallah R and Lowe D 2002 A novel approach to modelling and exploiting uncertainty in stochastic control systems *Artificial Neural Networks - ICANN 2002* (Springer Berlin Heidelberg) pp 801–806
[3] Sbarbaro D, Murray-Smith R and Valdes A 2004 *Proceedings of International Symposium on Neural Networks - ISNN 2004* 52–58
[4] Sarangapani J 2006 *Neural Network Control of Nonlinear Disrete-Time Systems* (Taylor & Francis)
[5] Fel'dbaum A A 1965 *Optimal Control Systems* (Academic Press, New York)
[6] Bar-Shalom Y and Tse E T 1976 *Control and dynamic systems* **12** 99–172
[7] Fabri S G and Kadirkamanathan V 1998 *Automatica* **34** 245–253
[8] Šimandl M, Král L and Hering P 2005 Neural network based bicriterial dual control of nonlinear systems *Proceedings of the* 16^{th} *IFAC World Congress* (Prague: IFAC)
[9] Bugeja M K, Fabri S G and Camilleri L 2009 *IEEE Transactions on Systems, Man and Cybernetics - Part B* **39** 129–141
[10] Král L and Šimandl M 2011 *Int. J. Adapt. Control Signal Process.* **25** 949–964
[11] Fabri S G and Bugeja M K 2013 Unscented transform-based dual adaptive control of nonlinear MIMO systems *Proceedings of* 13^{th} *European Control Conference* pp 392–397 Zürich
[12] Ko J and Fox D 2008 GP-BayesFilters: Bayesian Filtering Using Gaussian Process Prediction and Observation Models *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on* vol 27

(Springer) pp 3471–3476 ISBN 9781424420582
[13] Ko J, Klein D J, Fox D and Haehnel D 2007 GP-UKF: Unscented Kalman Filters with Gaussian Process Prediction and Observation Models *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on* vol 54 (IEEE) pp 1901–1907 ISBN 1424409128
[14] Deisenroth M P, Turner R D, Huber M F, Hanebeck U D and Rasmussen C E 2012 *IEEE Trans. on Automatic Control* **57** 1865–1871
[15] Särkkä S, Hartikainen J, Svensson L and Sandblom F 2014 Gaussian Process Quadratures in Nonlinear Sigma-Point Filtering and Smoothing *Information Fusion (FUSION), 2014 17th International Conference on* pp 1–8
[16] Kocijan J, Murray-Smith R, Rasmussen C and Girard A 2004 Gaussian process model based predictive control *Proc. of the American Control Conf.* vol 3 pp 2214–2219
[17] Kocijan J, Girard A, Banko B and Murray-Smith R 2005 *Mathematical and Computer Modelling of Dynamic Systems* **11** 411–424
[18] Girard A, Rasmussen C E, Quiñonero Candela J and Murray-Smith R 2003 Gaussian Process Priors With Uncertain Inputs Application to Multiple-Step Ahead Time Series Forecasting *Advances in Neural Information Processing Systems 15* ed Becker S, Thrun S and Obermayer K (MIT Press) pp 545–552
[19] Deisenroth M and Rasmussen C E 2011 Pilco: A model-based and data-efficient approach to policy search *Proc. of the 28th Int. Conf. on Machine Learning (ICML-11)* pp 465–472
[20] Park S, Mustafa S K and Shimada K 2013 Learning Based Robot Control with Sequential Gaussian Process *Robotic Intelligence In Informationally Structured Space (RiiSS), 2013 IEEE Workshop on* (IEEE) pp 120–127 ISBN 978-1-4673-5877-4
[21] Rasmussen C E and Ghahramani Z 2003 Bayesian Monte Carlo *Advances in Neural Information Processing Systems 15* 1 ed Becker S, Thrun S and Obermayer K (MIT Press) pp 505—-512
[22] Csató L and Opper M 2002 *Neural Computation* **14** 641–668
[23] Ranganathan A, Yang M H and Ho J 2011 *IEEE Transactions on Image Processing* **20** 391–404
[24] Särkkä S, Solin A and Hartikainen J 2013 *IEEE Signal Processing Magazine* **30** 51–61 ISSN 1053-5888
[25] Huber M 2013 Recursive gaussian process regression *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE Int. Conf. on* pp 3362–3366 ISSN 1520-6149
[26] Chowdhary G, Kingravi H A, How J P and Vela P A 2014 *Neural Networks and Learning Systems, IEEE Transactions on* **PP** 1–1 ISSN 2162-237X
[27] Král L, Prüher J and Šimandl M 2014 Gaussian process based dual adaptive control of nonlinear stochastic systems *Proceedings of the* 22*nd Mediterranean Conference on Control and Automation* (Palermo)
[28] Prüher J and Šimandl M 2014 Gaussian process based recursive system identification *Proceedings of the* 11^{th} *European Workshop on Advanced Control and Diagnosis* (Berlin)
[29] Chen F C and Khalil H K 1995 *IEEE Transaction on Automatic Control* **40** 791–801
[30] Filatov N M and Unbehauen H 2004 *Adaptive Dual Control* (Springer-Verlag)
[31] Rasmussen C E and Williams C K 2006 *Gaussian Processes for Machine Learning* (The MIT Press) ISBN 978-262-18253-9
[32] Sbarbaro D and Murray-Smith R 2003 Self-tuning Control of Non-linear Systems Using Gaussian Process Prior Models *Switching and Learning in Feedback Systems* ed Murray-Smith R and Shorten R (Berlin Heidelberg: Springer) pp 140–157
[33] MacKay D J C 2003 *Information Theory, Inference, and Learning Algorithms* (Cambridge University Press) ISBN 978-0521642989
[34] Prüher J and Šimandl M 2014 *Journal of Physics: Conference Series* **570** 1–9