

# Learning Based Model Predictive Control for Quadcopters with Dual Gaussian Process

Yuhan Liu and Roland Tóth

**Abstract**—An important issue in quadcopter control is that an accurate dynamic model of the system is nonlinear, complex, and costly to obtain. This limits achievable control performance in practice. Gaussian process (GP) based estimation is an effective tool to learn unknown dynamics from input/output data. However, conventional GP-based control methods often ignore the computational cost associated with accumulating data during operation of the system and how to handle forgetting in continuous adaption. In this paper, we present a novel Dual Gaussian Process (DGP) based model predictive control strategy that improves the performance of a quadcopter during trajectory tracking. The bio-inspired DGP structure is a combination of a long-term GP and a short-term GP, where the long-term GP is used to keep the learned knowledge in memory and the short-term GP is employed to rapidly compensate unknown dynamics during online operation. Furthermore, a novel recursive online update strategy for the short-term GP is proposed to successively improve the learnt model during online operation in an efficient way. Numerical simulations are used to demonstrate the effectiveness of the proposed strategy.

## I. INTRODUCTION

Over the past decades, quadcopters have attracted significant attention in many research studies due to their wide applicability [1], [2]. Flight control design for a quadcopter for agile maneuvering is not a simple task due to the quadcopter having six degrees of freedom with only four inputs to control the vehicle, which also leads to strong nonlinear couplings in the dynamics. Moreover, time-varying uncertainties and unmodeled dynamics, such as external wind, complex interactions of rotor airflows with the ground and objects such as walls, friction and flapping dynamics of the rotor blades make the control design problem quite complex. Hence a comprehensive and precise model of a quadcopter is still costly to obtain. Therefore, quadcopters mainly rely on robust control solutions against various disturbances and unmodeled dynamics, often limiting maneuvering capabilities to relatively small pitch and roll angles.

Recently, machine learning-based control has shown potential to save effort in understanding unmodeled dynamics and achieving superior performance in compensatory control over physical model-based control. Especially, *Gaussian Process* (GP) regression [3], which is a Bayesian nonparametric

data-driven modeling method, is promising to automatically extract important features of the unknown dynamics from system input and output data. Gaussian process regression is a powerful nonparametric framework using distribution over functions for nonlinear regression. The main advantage of GP regression is that it does not only provide a single function estimate, but also a variance, i.e. confidence interval, which indicates the accuracy of the regression result. In [4], a GP-based data-driven dynamic model was identified for reducing uncertainty of a given model and adapting the feedback gain by taking into account the model confidence. Furthermore, feedback linearization based control has been also achieved by learning a control affine GP model [5].

In general, GP is not suitable for large training sets because the computational cost to train a GP is cubic in terms of the number of data points. To solve this issue, various methods have been developed to reduce the computational load, but preserve the prediction accuracy. The simplest way is to choose a *subset of data* (SoD) of size  $M \ll N$  randomly to represent the full training set. However, this strategy can lead to serious underfitting as the probability of discarding crucial information included in the full training set grows with the reduction rate. In [6], instead of choosing subset randomly, an *information vector machine* based method is proposed to select the subset in a more intelligent way. Besides the SoD method, many complicated techniques regarding more efficient model learning approaches have been explored, including *Nyström method* [7], *deterministic training conditional approximation* (DTC) [8], [9], *partially independent training conditional approximation* (PITC) [10], and *fully independent training conditional approximation* (FITC) [11]. A unifying framework of the mentioned algorithms are discussed in [12]. Recently, several studies for GP-MPC were carried out with the applications of race cars [13], [14], robot arms [15], and quadcopters [16]. However, most GP-based control methods do not perform online updating to avoid the issue of data bloating, although this would allow online adaptation with GP-based control during the system operation. By using data approximation techniques to counteract data bloating in online updating, the GP can gradually lose its knowledge learned during the training phase, and morph into a purely online adaptive regressor. Thus, designing a "memory"-based GP structure with online learning ability deserves further study.

Inspired by the biological concept of "long/short term memory" of human beings, we propose a novel *dual Gaussian process* (DGP) structure based model predictive controller for quadcopter trajectory tracking. The main contri-

Y. Liu is with the Harbin Institute of Technology, Department of Control Science and Engineering, Harbin, China and the Eindhoven University of Tech., Dep. of Electrical Eng., Control Systems Group, Eindhoven, The Netherlands. Email: y.liu11@tue.nl

R. Tóth is with the Institute for Computer Science and Control, Budapest, Hungary and the Eindhoven University of Tech., Dep. of Electrical Eng., Control Systems Group, Eindhoven, The Netherlands. Email: r.toth@tue.nl

This research was supported by the Ministry of Innovation and Technology NRDI Office within the framework of the Autonomous Systems National Laboratory Program and China Scholarship Council (CSC).

butions are: 1) By the authors' knowledge, the proposed DGP structure is the first framework that emphasizes both "memory" and "learning" ability by a combination of long/short-term GPs. As we show, such a method is capable of preventing forgetting relevant process information without the need of periodic reinforcement, but at the same time ensure exploration and adaptation to uncertain or varying aspects of the dynamics. A long-term GP is utilized for the former while a short-term GP is used for the latter objectives. 2) A novel recursive online updating strategy for the short-term GP is presented to continuously learn the unknown time-varying uncertainties during control operation. Numerical simulations are used to show the improvements regarding the performance of the baseline controller.

The paper is organized as follows. Section II introduces the mathematical model of the quadcopter and preliminaries of GP based regression and control. The novel recursive online sparse GP with dual structure is detailed in Section III. In Section IV, a DGP-MPC design procedure for quadcopters is presented. In Section V, simulation studies are provided to demonstrate the capabilities of the proposed method, while in Section VI conclusions on the achieved results are drawn.

## II. PROBLEM FORMULATION AND PRELIMINARIES

### A. Problem Formulation

The quadcopter is modeled as a rigid body with four rotors. Let  $\mathcal{I}$  denote the inertial frame and  $\mathcal{B}$  denote the body fixed frame which is attached to the *center of mass* (COM) of the quadcopter and oriented according to  $\mathcal{I}$ . Define  $\mathbf{p} \in \mathbb{R}^3$  and  $\mathbf{v} \in \mathbb{R}^3$  as the position and velocity of the COM. Denoting  $\zeta = [\phi \ \theta \ \psi]^\top$  as the Euler angles of the quadcopter, then from the Newton-Euler formulation, one can obtain the system model for the quadcopter:

$$\begin{aligned} \dot{\mathbf{p}} &= \mathbf{v}, & m\dot{\mathbf{v}} &= m\mathbf{g}\mathbf{e}_3 - T\mathbf{R}\mathbf{e}_3 + \mathbf{F}_\Delta, \\ \dot{\zeta} &= \Theta\boldsymbol{\omega}, & \mathbf{J}\dot{\boldsymbol{\omega}} &= -\boldsymbol{\omega}^\times \mathbf{J}\boldsymbol{\omega} + \boldsymbol{\tau} + \boldsymbol{\tau}_\Delta \end{aligned} \quad (1)$$

where  $\mathbf{R}$  is the rotation matrix from  $\mathcal{B}$  to  $\mathcal{I}$  with Z-Y-X sequence,  $\boldsymbol{\omega} \in \mathbb{R}^3$  denotes the angular velocity of  $\mathcal{B}$  with respect to  $\mathcal{I}$ , and the matrix  $\Theta \in \mathbb{R}^{3 \times 3}$  is described as  $\Theta = [c\theta \ 0 \ -s\theta c\phi, 0 \ 1 \ s\phi, s\theta \ 0 \ c\theta c\phi]^{-1}$ ,  $s$  and  $c$  are short hands for sine and cosine, respectively.  $m$  is the total mass of quadcopter,  $\mathbf{e}_3 = [0 \ 0 \ 1]^\top$  denotes the unit vector aligning with the gravity  $g$  in  $\mathcal{I}$ ,  $[\cdot]^\times : \mathbb{R}^3 \rightarrow \mathbb{R}^{3 \times 3}$  denotes the cross-product operator,  $\mathbf{J} \in \mathbb{R}^{3 \times 3}$  is the inertia matrix of the quadcopter,  $T \in \mathbb{R}^+$  is the thrust force generated by the four rotors, and  $\boldsymbol{\tau} \in \mathbb{R}^3$  represents the control torque expressed in  $\mathcal{B}$ . The terms  $\mathbf{F}_\Delta$  and  $\boldsymbol{\tau}_\Delta$  represent unknown force and torque applied on the quadcopter resulting due to time-varying uncertainties and unmodeled dynamics, such as external wind, complex interactions of rotor airflows affected by the ground and walls, frictions and flapping dynamics of the rotor blades.

By denoting  $\mathbf{x} = [\mathbf{p}^\top \ \mathbf{v}^\top \ \zeta^\top \ \boldsymbol{\omega}^\top]^\top \in \mathbb{R}^{12}$  and  $\mathbf{u} = [T \ \boldsymbol{\tau}^\top]^\top \in \mathbb{R}^4$ , the continuous-time model (1) can be linearized and discretized resulting in the LTI model:

$$\mathbf{x}(k+1) = \underbrace{\mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k)}_{\text{nominal}} + \underbrace{\mathbf{B}_d\boldsymbol{\Delta}(\mathbf{x}(k), \mathbf{u}(k))}_{\text{unknown}} \quad (2)$$

where  $k \in \mathbb{Z}$  is the discrete time satisfying  $t = kT_s$  with sampling time  $T_s > 0$ ,  $\mathbf{A} \in \mathbb{R}^{12 \times 12}$  and  $\mathbf{B} \in \mathbb{R}^{12 \times 4}$  are the known system matrices derived from the linearization of the idealistic rigid body dynamics,  $\boldsymbol{\Delta}(\mathbf{x}(k), \mathbf{u}(k)) = [\mathbf{F}_\Delta^\top, \boldsymbol{\tau}_\Delta^\top]^\top \in \mathbb{R}^6$  denotes the unknown dynamics that are either resulted due to the approximation error of the applied linearization and discretization or they cannot be captured reliably by the idealistic rigid body model (1) (i.e., both nonlinear effects and external disturbances),  $\mathbf{B}_d \in \mathbb{R}^{12 \times 6}$  specifies that which states are influenced by  $\boldsymbol{\Delta}$ .

In this paper, the control objective is to design an appropriate trajectory tracking controller for the system (1) where the robust satisfaction of state and input constraints are guaranteed. In the simulation, the state constraint refers to a limited safe-space where the quadcopter could explore and the input constraint corresponding to the maximal thrust force and torque that the quadcopter could provide.

### B. Learning with Gaussian Processes

To guarantee the aforementioned safety constraints, the unknown model  $\boldsymbol{\Delta}(\mathbf{x}(k), \mathbf{u}(k))$  should be identified reliably. Our goal is to construct a probabilistic model of  $\boldsymbol{\Delta}(\mathbf{x}(k), \mathbf{u}(k))$  from the system measurement data, and to improve accuracy gradually as more data are collected. We model  $\boldsymbol{\Delta}(\mathbf{x}(k), \mathbf{u}(k))$  as a GP where the state-input pairs  $\tilde{\mathbf{x}}(k) \triangleq (\mathbf{x}^\top(k) \ \mathbf{u}^\top(k))^\top \in \mathbb{R}^{12+4}$  and  $\boldsymbol{\Delta}(\tilde{\mathbf{x}}(k)) \triangleq \mathbf{x}(k+1) - \mathbf{A}\mathbf{x}(k) - \mathbf{B}\mathbf{u}(k)$  are denoted as training inputs and outputs, respectively. A GP is a distribution over functions, which is full specified by:

$$\begin{aligned} \mu(\tilde{\mathbf{x}}) &= \mathbb{E}[\boldsymbol{\Delta}(\tilde{\mathbf{x}})], \\ k(\tilde{\mathbf{x}}, \tilde{\mathbf{x}}') &= \mathbb{E}[(\boldsymbol{\Delta}(\tilde{\mathbf{x}}) - \mu(\tilde{\mathbf{x}}))(\boldsymbol{\Delta}(\tilde{\mathbf{x}}') - \mu(\tilde{\mathbf{x}}'))], \end{aligned} \quad (3)$$

where  $\mu(\tilde{\mathbf{x}})$  is the mean function and  $k(\tilde{\mathbf{x}}, \tilde{\mathbf{x}}') \triangleq \text{cov}(\boldsymbol{\Delta}(\tilde{\mathbf{x}}), \boldsymbol{\Delta}(\tilde{\mathbf{x}}'))$  is the positive semi-definite covariance function which denotes a measure for the correlation (or similarity) of any two data points  $(\tilde{\mathbf{x}}, \tilde{\mathbf{x}}')$ . Then we assume that:  $\boldsymbol{\Delta}(\tilde{\mathbf{x}}) \sim \mathcal{GP}(\mu(\tilde{\mathbf{x}}), k(\tilde{\mathbf{x}}, \tilde{\mathbf{x}}'))$ . The prior mean function is usually set to zero as no priori knowledge of  $\boldsymbol{\Delta}(\tilde{\mathbf{x}})$  is given. Furthermore, there is a wide selection of the covariance functions, such as sinusoidal and Matérn kernels, and in this paper we use the squared exponential function  $k(\tilde{\mathbf{x}}, \tilde{\mathbf{x}}') = \sigma_f^2 \exp(-\frac{1}{2}(\tilde{\mathbf{x}} - \tilde{\mathbf{x}}')^\top \boldsymbol{\Lambda}^{-2}(\tilde{\mathbf{x}} - \tilde{\mathbf{x}}'))$  as a priori due to its universal approximation capability, where  $\sigma_f^2 \in \mathbb{R}_{\geq 0}$  and  $\boldsymbol{\Lambda} = \text{diag}(\lambda_1, \dots, \lambda_d)$  are signal variance and length-scale hyperparameters, respectively.

Consider the training set  $\mathcal{D} = \{\tilde{\mathbf{X}} = [\tilde{\mathbf{x}}_1, \dots, \tilde{\mathbf{x}}_N], \mathbf{Y} = [\mathbf{y}_1, \dots, \mathbf{y}_N]\}$  with  $N$  noisy output  $\mathbf{y}_i = \boldsymbol{\Delta}(\tilde{\mathbf{x}}_i) + \boldsymbol{\epsilon}$  (i.e., corrupted by an *i.i.d.* Gaussian noise  $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \sigma_\epsilon^2 \mathbf{I})$  with standard deviation  $\sigma_\epsilon$ ). The Gaussian prior distribution for the latent function  $\boldsymbol{\Delta}$  and the model likelihood of the training set  $\mathcal{D}$  are given by  $P(\boldsymbol{\Delta}) = \mathcal{N}(\boldsymbol{\Delta}|\mathbf{0}, \mathbf{K}_N)$  and  $P(\mathbf{Y}|\boldsymbol{\Delta}) = \mathcal{N}(\mathbf{Y}|\boldsymbol{\Delta}, \sigma_\epsilon^2 \mathbf{I}_N)$  respectively, where  $\mathbf{K}_N \in \mathbb{R}^{N \times N}$  denotes the kernel matrix with  $[\mathbf{K}]_{ij} = k(\tilde{\mathbf{x}}_i, \tilde{\mathbf{x}}_j)$ . Thus, according to the Bayes' rule, one can obtain the posterior distribution by *maximum a posterior* (MAP) estimation  $P(\boldsymbol{\Delta}|\mathbf{Y}) \propto P(\mathbf{Y}|\boldsymbol{\Delta})P(\boldsymbol{\Delta}) \propto \mathcal{N}(\boldsymbol{\Delta}|\mathbf{K}_N(\mathbf{K}_N + \sigma_\epsilon^2 \mathbf{I}_N)^{-1}\mathbf{Y}, (\mathbf{K}_N^{-1} + \sigma_\epsilon^{-2} \mathbf{I}_N)^{-1})$ . The associated posterior distribution of the

function value  $\Delta(\tilde{\mathbf{x}}^*)$  at a new test point  $\tilde{\mathbf{x}}^*$  can be derived by  $P(\Delta^*|\mathcal{D}, \tilde{\mathbf{x}}^*) = \int P(\Delta^*|\tilde{\mathbf{X}}, \Delta, \tilde{\mathbf{x}}^*)P(\Delta|\mathbf{Y})d\Delta = \mathcal{N}(\boldsymbol{\mu}_\Delta(\tilde{\mathbf{x}}^*), \boldsymbol{\Sigma}_\Delta(\tilde{\mathbf{x}}^*))$  with the predictive mean and variance:

$$\boldsymbol{\mu}_\Delta(\tilde{\mathbf{x}}^*) = \mathbf{K}_{*N}(\mathbf{K}_N + \sigma_\epsilon^2 \mathbf{I}_N)^{-1} \mathbf{Y} \quad (4)$$

$$\boldsymbol{\Sigma}_\Delta(\tilde{\mathbf{x}}^*) = k_{**} - \mathbf{K}_{*N}(\mathbf{K}_N + \sigma_\epsilon^2 \mathbf{I}_N)^{-1} \mathbf{K}_{N*} \quad (5)$$

where  $[\mathbf{K}_{N*}]_j = k(\tilde{\mathbf{X}}_j, \tilde{\mathbf{x}}^*)$ ,  $\mathbf{K}_{*N} = \mathbf{K}_{N*}^\top$  and  $k_{**} = k(\tilde{\mathbf{x}}^*, \tilde{\mathbf{x}}^*)$ . Furthermore, the marginal likelihood  $P(\mathbf{Y}) = \mathcal{N}(\mathbf{0}, \mathbf{K}_N + \sigma_\epsilon^2 \mathbf{I}_N)$  and the hyperparameters  $\boldsymbol{\theta} = [\sigma_\epsilon^2, \sigma_f^2, \lambda_1, \dots, \lambda_d]^\top$  can be optimized by maximizing  $\log P(\mathbf{Y}) = -\frac{N}{2} \log 2\pi - \log |\mathbf{K}_N + \sigma_\epsilon^2 \mathbf{I}_N| - \frac{1}{2} \mathbf{Y}^\top (\mathbf{K}_N + \sigma_\epsilon^2 \mathbf{I}_N)^{-1} \mathbf{Y}$  by means of conjugate gradient-based algorithm [3], which has a computational load of  $\mathcal{O}(N^3)$  per iteration.

### III. RECURSIVE ONLINE SPARSE GAUSSIAN PROCESS WITH DUAL STRUCTURE

#### A. Sparse Gaussian Process

The computational load per test case for the full GP in (4) and (5) is  $\mathcal{O}(N)$  and  $\mathcal{O}(N^2)$  in terms of mean and variance, respectively. In this paper, we focus on the generalization of *sparse variational GP regression* (SVGP), which has been first introduced in [17].

The goal of sparse GP is to find a set of pseudo inputs  $\tilde{\mathbf{X}}_u = [\tilde{\mathbf{x}}_{u,1}, \dots, \tilde{\mathbf{x}}_{u,M}]$  corresponding to pseudo outputs  $\Delta_u = [\Delta_{u,1}, \dots, \Delta_{u,M}]$  of size  $M \ll N$ . In SVGP, the true posterior distribution  $P(\Delta|\mathcal{D})$  is approximated by a Gaussian distribution  $q(\Delta, \Delta_u) = P(\Delta|\Delta_u)q(\Delta_u)$  by means of variational inference, where a tractable Gaussian  $q(\Delta_u) = \mathcal{N}(\Delta_u|\mathbf{m}_u, \mathbf{S}_u)$  is used, corresponding to the maximization of the *evidence lower bound* (ELBO) of  $\log P(\mathbf{Y})$ :

$$\begin{aligned} \mathcal{L}(q) &\triangleq \int q(\Delta, \Delta_u) \log \frac{P(\mathbf{Y}, \Delta, \Delta_u)}{q(\Delta, \Delta_u)} d\Delta d\Delta_u \\ &= F(q) - \mathcal{KL}(q(\Delta_u) || P(\Delta_u)), \end{aligned} \quad (6)$$

where  $F(q) = \int q(\Delta_u) P(\Delta|\Delta_u) \log P(\mathbf{Y}|\Delta) d\Delta d\Delta_u$ , and  $\mathcal{KL}(\cdot||\cdot)$  represents the Kullback-Leibler divergence which quantifies the similarity between two distributions. Then the optimal  $q^*(\Delta_u)$  in terms of minimum of (6) with mean  $\mathbf{m}_u$  and variance  $\mathbf{S}_u$  can be derived as:

$$\begin{aligned} \mathbf{m}_u &= \sigma_\epsilon^{-2} \mathbf{S}_u \mathbf{K}_M^{-1} \mathbf{K}_{MN} \mathbf{Y} \\ \mathbf{S}_u &= \mathbf{K}_M (\mathbf{K}_M + \sigma_\epsilon^{-2} \mathbf{K}_{MN} \mathbf{K}_{NM})^{-1} \mathbf{K}_M \end{aligned} \quad (7)$$

where  $[\mathbf{K}_{MN}]_{ij} = k(\tilde{\mathbf{x}}_{ui}, \tilde{\mathbf{x}}_j)$  denotes the covariance matrix between pseudo inputs  $\tilde{\mathbf{X}}_u$  and training inputs  $\tilde{\mathbf{X}}$ ,  $\mathbf{K}_{MN} = \mathbf{K}_{MN}^\top$ ,  $\mathbf{K}_M$  is the covariance matrix of pseudo inputs. Merging the optimal distribution  $q^*(\Delta_u)$  into  $\mathcal{L}(q)$ , we get:

$$\mathcal{L}(q) = \log \mathcal{N}(\mathbf{Y}|\mathbf{0}, \mathbf{Q}_N + \sigma_\epsilon^2 \mathbf{I}_N) - \frac{1}{2\sigma_\epsilon^2} \text{tr}(\mathbf{K}_N - \mathbf{Q}_N) \quad (8)$$

where  $\mathbf{Q}_N = \mathbf{K}_{NM} \mathbf{K}_M^{-1} \mathbf{K}_{MN}$ . The pseudo inputs  $\tilde{\mathbf{X}}_u$  and hyperparameters can be optimized by maximizing (8), which results in a computational load of  $\mathcal{O}(NM^2 + M^3)$ .

With the approximated posterior  $q(\Delta, \Delta_u)$ , the marginal distribution of  $\Delta$  is  $q(\Delta) = \int P(\Delta|\Delta_u)q(\Delta_u) d\Delta_u = \mathcal{N}(\boldsymbol{\mu}_\Delta(\tilde{\mathbf{x}}), \boldsymbol{\Sigma}_\Delta(\tilde{\mathbf{x}}))$  with mean and variance:

$$\boldsymbol{\mu}_\Delta(\tilde{\mathbf{x}}^*) = \mathbf{K}_{*M} \mathbf{K}_M^{-1} \mathbf{m}_u \quad (9)$$

$$\boldsymbol{\Sigma}_\Delta(\tilde{\mathbf{x}}^*) = k(\tilde{\mathbf{x}}^*, \tilde{\mathbf{x}}^*) - \mathbf{K}_{*M} (\mathbf{K}_M^{-1} - \mathbf{K}_M^{-1} \mathbf{S}_u \mathbf{K}_M^{-1}) \mathbf{K}_{M*} \quad (10)$$

where  $[\mathbf{K}_{*M}]_j = k(\tilde{\mathbf{x}}^*, \tilde{\mathbf{x}}_{u,j})$ .

It is worth mentioning that, the hyperparameters and pseudo points are fixed after the ELBO maximization, and the GP is used to predict the mean and variance at a new data point during the operation. However, due to the fact that the flying environment for the quadcopter is complicated and time-varying, the offline pre-trained GP model is required to be successively improved to ensure continuous adaptation.

#### B. Recursive Online Sparse Gaussian Process

Next a novel online update strategy for sparse Gaussian processes is proposed. The proposed method updates the posterior mean and variance of  $q(\Delta_u)$  recursively based on the regression error at the current time step  $k$ .

On the basis of recursive Bayesian regression, and given a new online measurement output  $\mathbf{y}_k = \Delta(\tilde{\mathbf{x}}_k) + \epsilon$ , the posterior mean and variance (7) at the  $k^{\text{th}}$ -step  $q_k(\Delta_u) = \mathcal{N}(\Delta_u|\mathbf{m}_u^k, \mathbf{S}_u^k)$  can be rewritten in terms of the posterior  $q_{1:k-1}(\Delta_u) = \mathcal{N}(\Delta_u|\mathbf{m}_u^{k-1}, \mathbf{S}_u^{k-1})$ :

$$\begin{aligned} \mathbf{m}_u^k &= \mathbf{S}_u^k (\mathbf{S}_u^{k-1} \mathbf{m}_u^{k-1} + \sigma_\epsilon^{-2} \boldsymbol{\Phi}_k^\top \mathbf{y}_k) \\ \mathbf{S}_u^k &= (\mathbf{S}_u^{k-1} + \sigma_\epsilon^{-2} \boldsymbol{\Phi}_k^\top \boldsymbol{\Phi}_k)^{-1} \end{aligned} \quad (11)$$

with kernel  $\boldsymbol{\Phi}_k = \mathbf{K}_{xM} \mathbf{K}_M^{-1}|_{\tilde{\mathbf{x}}=\tilde{\mathbf{x}}_k}$  and  $[\boldsymbol{\Phi}]_j = \phi(\tilde{\mathbf{x}}, \tilde{\mathbf{x}}_{uj})$ . Then (9) can be seen as a linear combination of  $M$  kernel functions, each one is corresponding to a pseudo input:

$$\boldsymbol{\mu}_\Delta(\tilde{\mathbf{x}}) = \sum_{j=1}^M m_{uj} \phi(\tilde{\mathbf{x}}, \tilde{\mathbf{x}}_{uj}) = \boldsymbol{\Phi} \mathbf{m}_u \quad (12)$$

Note that (12) can be interpreted as a weight-space representation of (9). Thus, recursive least squares constitutes an efficient way to update (9) and (10) with online streaming data points. Given a new data point  $\{\tilde{\mathbf{x}}_k, \mathbf{y}_k\}$ , the posterior mean and variance can be updated by:

$$\begin{cases} \mathbf{m}_u^k &= \mathbf{m}_u^{k-1} + \mathbf{L}_k \mathbf{r}_k \\ \mathbf{r}_k &= \mathbf{y}_k - \boldsymbol{\Phi}_k \mathbf{m}_u^{k-1} \\ \mathbf{L}_k &= \mathbf{S}_u^{k-1} \boldsymbol{\Phi}_k^\top \mathbf{G}_k^{-1} \\ \mathbf{G}_k &= \lambda + \boldsymbol{\Phi}_k \mathbf{S}_u^{k-1} \boldsymbol{\Phi}_k^\top \\ \mathbf{S}_u^k &= \lambda^{-1} (\mathbf{S}_u^{k-1} - \mathbf{L}_k \mathbf{G}_k \mathbf{L}_k^\top) \end{cases} \quad (13)$$

where  $\lambda$  is the forgetting factor. The recursion starts from  $\mathbf{m}_u^0$  and  $\mathbf{S}_u^0$ , which can be seen as the prior of the GP. This recursive online sparse GP can be embedded into the predictor of an MPC scheme, and is capable of dealing with varying disturbances and sudden changes in the dynamics.

#### C. Dual Gaussian Process Structure

Despite the adapting capability of the recursive online sparse GP, the learned dynamics obtained during the offline training phase, will be forgotten during the online learning phase. This means that, if the current evidence during the online control phase does not support the dynamics seen, they will gradually disappear via the forgetting factor  $\lambda$  according to (13). For example, the "knowledge" for the uncertainties with large roll and pitch angles will fade in case the quadcopter is hovering for a while. Inspired by the biological concept of "long/short term memory", we present

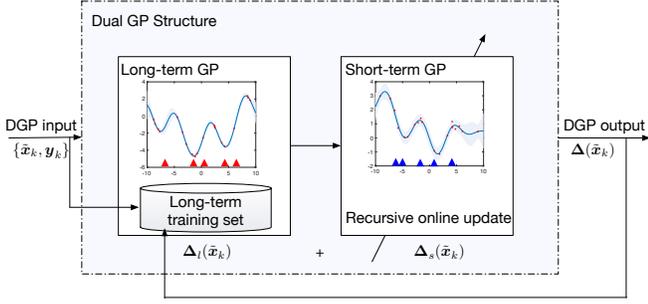


Fig. 1: An illustration of the Dual Gaussian Process structure

a novel structure named *Dual Gaussian Process* (see Fig. 1), which allows the storage of collected experience and is able to prevent knowledge forgetting.

The DGP structure mainly consists of two GPs: The first GP corresponds to the long-term GP, which is employed to learn all the time-independent uncertainties and disturbances, including aerodynamics besides the rigid body dynamics and indoor effects. The hyperparameters and the pseudo points of the long-term GP keeps fixed during online learning phase, and will be batch updated after each mission. This makes the long-term GP to have the ability to keep the offline trained memory in mind and evolve from mission to mission. On the other hand, the short-term GP is utilized to adapt with online time-varying disturbances and (sudden) changes in dynamics. It is worth mentioning that, the short-term GP learns around on the predicted mean value of the long-term GP, and the posterior is recursively updated during the online learning phase. Hence, the unknown dynamics  $\Delta(\tilde{x})$  can be rewritten as a sum of two independent GPs:

$$\Delta(\tilde{x}) = \Delta_l(\tilde{x}) + \Delta_s(\tilde{x}) \quad (14a)$$

$$\Delta_l(\tilde{x}) \sim \mathcal{GP}(\boldsymbol{\mu}_l(\tilde{x}), k(\tilde{x}, \tilde{x}')), \quad (14b)$$

$$\Delta_s(\tilde{x}) \sim \mathcal{GP}(\mathbf{0}, v(\tilde{x}, \tilde{x}')) \quad (14c)$$

where the subscript  $l$  and  $s$  represent "long-term" and "short-term", respectively.  $v(\tilde{x}, \tilde{x}')$  denotes the covariance function for the short-term GP. For the simplicity of notation, the prior mean function is set as zero in the following derivation. Furthermore, the mean function  $\boldsymbol{\mu}_l(\tilde{x})$  for the long-term GP is obtained from the offline training phase. Then, considering the training set  $\mathcal{D}$ , the prior distribution for  $\Delta_l$  and  $\Delta_s$  are given by  $P(\Delta_l) = \mathcal{N}(\Delta_l | \boldsymbol{\mu}_l, \mathbf{K}_N)$  and  $P(\Delta_s) = \mathcal{N}(\Delta_s | \mathbf{0}, \mathbf{V}_N)$ . Moreover, the model likelihood is denoted as  $P(\mathbf{Y} | \Delta_l, \Delta_s) = \mathcal{N}(\mathbf{Y} | \Delta_l + \Delta_s, \sigma_\epsilon^2 \mathbf{I}_N)$  where  $\mathbf{V}_N$  represents the kernel matrix with  $[\mathbf{V}]_{ij} = v(\tilde{x}_i, \tilde{x}_j)$ . To deal with the large data sets, we introduce two sets of pseudo points for the DGP, i.e.,  $\mathbf{u}_l = \Delta_l(z_l)$  and  $\mathbf{u}_s = \Delta_s(z_s)$  where  $z_{(\cdot)} = [z_{(\cdot)1}, \dots, z_{(\cdot)M}]$  denote the pseudo inputs for the long/short-term GP, respectively. Subsequently, one can obtain the following conditional distributions:  $P(\Delta_l | \mathbf{u}_l) = \mathcal{N}(\Delta_l | \mathbf{K}_{NM} \mathbf{K}_M^{-1} \mathbf{u}_l, \mathbf{K}_N - \mathbf{Q}_N)$  and  $P(\Delta_s | \mathbf{u}_s) = \mathcal{N}(\Delta_s | \mathbf{V}_{NM} \mathbf{V}_M^{-1} \mathbf{u}_s, \mathbf{V}_N - \mathbf{O}_N)$ , where  $\mathbf{O}_N = \mathbf{V}_{NM} \mathbf{V}_M^{-1} \mathbf{V}_{MN}$ . The marginal likelihood gives  $\log P(\mathbf{Y}) = \log \int P(\mathbf{Y} | \Delta_l, \Delta_s) P(\Delta_l | \mathbf{u}_l) P(\Delta_s | \mathbf{u}_s) P(\mathbf{u}_l) P(\mathbf{u}_s) d\Delta_l d\Delta_s d\mathbf{u}_l d\mathbf{u}_s$ . Next, two variational distributions  $q(\mathbf{u}_l) = \mathcal{N}(\mathbf{u}_l | \mathbf{m}_{ul}, \mathbf{S}_{ul})$  and  $q(\mathbf{u}_s) = \mathcal{N}(\mathbf{u}_s | \mathbf{m}_{us}, \mathbf{S}_{us})$  are intro-

duced to approximate the posterior:  $P(\Delta_l, \Delta_s, \mathbf{u}_l, \mathbf{u}_s | \mathbf{Y}) \approx P(\Delta_l | \mathbf{u}_l) P(\Delta_s | \mathbf{u}_s) q(\mathbf{u}_l) q(\mathbf{u}_s)$ . The marginalized distribution of  $q(\Delta_l) = \int P(\Delta_l | \mathbf{u}_l) q(\mathbf{u}_l) d\mathbf{u}_l$  and  $q(\Delta_s) = \int P(\Delta_s | \mathbf{u}_s) q(\mathbf{u}_s) d\mathbf{u}_s$  can be simply computed as

$$q(\Delta_l) = \mathcal{N}(\mathbf{K}_{NM} \mathbf{K}_M^{-1} \mathbf{m}_{ul}, \mathbf{K}_N + \hat{\mathbf{Q}}_N) \triangleq \mathcal{N}(\mathbf{m}_l, \mathbf{S}_l) \quad (15a)$$

$$q(\Delta_s) = \mathcal{N}(\mathbf{V}_{NM} \mathbf{V}_M^{-1} \mathbf{m}_{us}, \mathbf{V}_N + \hat{\mathbf{O}}_N) \triangleq \mathcal{N}(\mathbf{m}_s, \mathbf{S}_s) \quad (15b)$$

where  $\hat{\mathbf{Q}}_N = \mathbf{K}_{NM} \mathbf{K}_M^{-1} (\mathbf{S}_{ul} - \mathbf{K}_M) \mathbf{K}_M^{-1} \mathbf{K}_{MN}$ , and  $\hat{\mathbf{O}}_N = \mathbf{V}_{NM} \mathbf{V}_M^{-1} (\mathbf{S}_{us} - \mathbf{V}_M) \mathbf{V}_M^{-1} \mathbf{V}_{MN}$ . It is worth mentioning that the posterior for long-term GP  $q(\Delta_l)$  is obtained after the offline training phase and it is fixed during the inference. Furthermore, the posterior for the short-term GP  $q(\mathbf{u}_s)$  is updated with (13). Then we can obtain the lower bound of  $\log P(\mathbf{Y})$  for DGP as follows:

$$\begin{aligned} \mathcal{L}(q) &\triangleq \int q(\Delta_l, \Delta_s, \mathbf{u}_l, \mathbf{u}_s) \log \frac{P(\mathbf{Y}, \Delta_l, \Delta_s, \mathbf{u}_l, \mathbf{u}_s)}{q(\Delta_l, \Delta_s, \mathbf{u}_l, \mathbf{u}_s)} \\ &\quad d\Delta_l d\Delta_s d\mathbf{u}_l d\mathbf{u}_s \\ &= \int q(\mathbf{u}_l) \int P(\Delta_s | \mathbf{u}_s) q(\mathbf{u}_s) \int P(\Delta_l | \mathbf{u}_l) \\ &\quad \log P(\mathbf{Y} | \Delta_l, \Delta_s) d\Delta_l d\Delta_s d\mathbf{u}_s d\mathbf{u}_l \\ &\quad - \mathcal{KL}(q(\mathbf{u}_l) || P(\mathbf{u}_l)) - \mathcal{KL}(q(\mathbf{u}_s) || P(\mathbf{u}_s)) \quad (16) \end{aligned}$$

Due to the fact that the posterior of the long-term GP is fixed during the online learning phase, we can equate the variational parameters of  $q(\mathbf{u}_l)$  to (7):  $\mathbf{m}_{ul} = \mathbf{m}_u$  and  $\mathbf{S}_{ul} = \mathbf{S}_u$ . Next we will compute the analytical form of (16) to get the optimal variational parameters for  $q(\mathbf{u}_s)$ . The inner integral for the first term of (16) can be derived as:

$$\begin{aligned} &\int P(\Delta_l | \mathbf{u}_l) \log P(\mathbf{Y} | \Delta_l, \Delta_s) d\Delta_l \\ &= \int \mathcal{N}(\Delta_l | \mathbf{K}_{NM} \mathbf{K}_M^{-1} \mathbf{u}_l, \mathbf{K}_N - \mathbf{Q}_N) \log P(\mathbf{Y} | \Delta_l, \Delta_s) d\Delta_l \\ &= \log \mathcal{N}(\mathbf{Y} | \mathbf{K}_{NM} \mathbf{K}_M^{-1} \mathbf{u}_l + \Delta_s, \sigma_\epsilon^2 \mathbf{I}_N) - \frac{1}{2\sigma_\epsilon^2} \text{tr}(\mathbf{K}_N - \mathbf{Q}_N) \quad (17) \end{aligned}$$

By merging inner integral into the second integral in (16):

$$\begin{aligned} &\int P(\Delta_s | \mathbf{u}_s) q(\mathbf{u}_s) \int P(\Delta_l | \mathbf{u}_l) \log P(\mathbf{Y} | \Delta_l, \Delta_s) d\Delta_l d\mathbf{u}_s \\ &= \int q(\Delta_s) \log \mathcal{N}(\mathbf{Y} | \mathbf{K}_{NM} \mathbf{K}_M^{-1} \mathbf{u}_l + \Delta_s, \sigma_\epsilon^2 \mathbf{I}_N) d\Delta_s \\ &\quad - \frac{1}{2\sigma_\epsilon^2} \text{tr}(\mathbf{K}_N - \mathbf{Q}_N) \\ &= \log \mathcal{N}(\mathbf{Y} | \mathbf{K}_{NM} \mathbf{K}_M^{-1} \mathbf{u}_l + \mathbf{m}_s, \sigma_\epsilon^2 \mathbf{I}_N) \\ &\quad - \frac{1}{2\sigma_\epsilon^2} \text{tr}(\mathbf{K}_N - \mathbf{Q}_N) - \frac{1}{2\sigma_\epsilon^2} \text{tr}(\mathbf{S}_s). \quad (18) \end{aligned}$$

Substituting (18) back to (16), one has:

$$\begin{aligned} \mathcal{L}(q) &= \int q(\mathbf{u}_l) \log \mathcal{N}(\mathbf{Y} | \mathbf{K}_{NM} \mathbf{K}_M^{-1} \mathbf{u}_l + \mathbf{m}_s, \sigma_\epsilon^2 \mathbf{I}_N) d\mathbf{u}_l \\ &\quad - \mathcal{KL}(q(\mathbf{u}_l) || P(\mathbf{u}_l)) - \mathcal{KL}(q(\mathbf{u}_s) || P(\mathbf{u}_s)) \\ &\quad - \frac{1}{2\sigma_\epsilon^2} \text{tr}(\mathbf{K}_N - \mathbf{Q}_N) - \frac{1}{2\sigma_\epsilon^2} \text{tr}(\mathbf{S}_s) \quad (19) \\ &\leq \log \int \mathcal{N}(\mathbf{Y} | \mathbf{K}_{NM} \mathbf{K}_M^{-1} \mathbf{u}_l + \mathbf{m}_s, \sigma_\epsilon^2 \mathbf{I}_N) P(\mathbf{u}_l) d\mathbf{u}_l \\ &\quad - \mathcal{KL}(q(\mathbf{u}_s) || P(\mathbf{u}_s)) - \frac{1}{2\sigma_\epsilon^2} \text{tr}(\mathbf{K}_N - \mathbf{Q}_N) - \frac{1}{2\sigma_\epsilon^2} \text{tr}(\mathbf{S}_s) \end{aligned}$$

Using Gaussian identities, the integral in the log-term can be further simplified as:  $\int \mathcal{N}(\mathbf{Y}|\mathbf{K}_{NM}\mathbf{K}_M^{-1}\mathbf{u}_l + \mathbf{m}_s, \sigma_\epsilon^2 \mathbf{I}_N) P(\mathbf{u}_l) d\mathbf{u}_l = \mathcal{N}(\mathbf{Y}|\mathbf{m}_s, \mathbf{Q}_N + \sigma_\epsilon^2 \mathbf{I}_N)$ . Furthermore, the KL-divergence between two Gaussians is analytical, that is:  $\mathcal{KL}(q(\mathbf{u}_s)||P(\mathbf{u}_s)) = \frac{1}{2} \log |\mathbf{V}_M| + \frac{1}{2} \mathbf{m}_{us}^\top \mathbf{V}_M^{-1} \mathbf{m}_{us} + \frac{1}{2} \text{tr}(\mathbf{S}_{us} \mathbf{V}_M^{-1}) - \frac{1}{2} M - \frac{1}{2} \log |\mathbf{S}_{us}|$ . Consequently, one gets the analytical form of  $\mathcal{L}(q)$ :

$$\begin{aligned} \mathcal{L}(q) = & \log \mathcal{N}(\mathbf{Y}|\mathbf{m}_s, \mathbf{Q}_N + \sigma_\epsilon^2 \mathbf{I}_N) - \frac{1}{2\sigma_\epsilon^2} \text{tr}(\mathbf{K}_N - \mathbf{Q}_N) \\ & - \frac{1}{2\sigma_\epsilon^2} \text{tr}(\mathbf{S}_s) - \frac{1}{2} \log |\mathbf{V}_M| - \frac{1}{2} \mathbf{m}_{us}^\top \mathbf{V}_M^{-1} \mathbf{m}_{us} \\ & - \frac{1}{2} \text{tr}(\mathbf{S}_{us} \mathbf{V}_M^{-1}) + \frac{1}{2} M + \frac{1}{2} \log |\mathbf{S}_{us}|. \end{aligned} \quad (20)$$

Taking the derivative of  $\mathcal{L}(q)$  with respect to the variational parameters of  $q(\mathbf{u}_s)$  and set them as zeros, we can obtain the optimal  $q^*(\mathbf{u}_s)$  with mean and variance satisfy:

$$\mathbf{m}_{us} = \mathbf{V}_M (\mathbf{V}_M + \mathbf{V}_{MN} \tilde{\mathbf{Q}}_N^{-1} \mathbf{V}_{NM})^{-1} \mathbf{V}_{MN} \tilde{\mathbf{Q}}_N^{-1} \mathbf{Y} \quad (21)$$

$\mathbf{S}_{us} = \mathbf{V}_M (\mathbf{V}_M + \sigma_\epsilon^{-2} \mathbf{V}_{MN} \mathbf{V}_{NM})^{-1} \mathbf{V}_M$  where  $\tilde{\mathbf{Q}}_N = \mathbf{Q}_N + \sigma_\epsilon^2 \mathbf{I}_N$ . Combining (15) with the learned variational parameters, one can derive the predictive distribution of both of the GPs:

$$\begin{aligned} q(\Delta_l^*) &= \int P(\Delta_l^*|\mathbf{u}_l) q^*(\mathbf{u}_l) d\mathbf{u}_l^* \triangleq \mathcal{N}(\boldsymbol{\mu}_l^*, \boldsymbol{\Sigma}_l^*) \\ q(\Delta_s^*) &= \int P(\Delta_s^*|\mathbf{u}_s) q^*(\mathbf{u}_s) d\mathbf{u}_s^* \triangleq \mathcal{N}(\boldsymbol{\mu}_s^*, \boldsymbol{\Sigma}_s^*) \end{aligned} \quad (22)$$

with  $\boldsymbol{\mu}_l^* = \mathbf{K}_{*M} \mathbf{K}_M^{-1} \mathbf{m}_{ul}$ ,  $\boldsymbol{\mu}_s^* = \mathbf{V}_{*M} \mathbf{V}_M^{-1} \mathbf{m}_{us}$ ,  $\boldsymbol{\Sigma}_l^* = k_{**} - \mathbf{K}_{*M} \mathbf{K}_M^{-1} \mathbf{K}_{M*} + \mathbf{K}_{*M} \mathbf{K}_M^{-1} \mathbf{S}_{ul} \mathbf{K}_M^{-1} \mathbf{K}_{M*}$ , and  $\boldsymbol{\Sigma}_s^* = v_{**} - \mathbf{V}_{*M} \mathbf{V}_M^{-1} \mathbf{V}_{M*} + \mathbf{V}_{*M} \mathbf{V}_M^{-1} \mathbf{S}_{us} \mathbf{V}_M^{-1} \mathbf{V}_{M*}$ .

Finally, the predictive distribution of the latent function  $\Delta$  at a new test point  $\tilde{\mathbf{x}}^*$  is given by  $q(\Delta^*) = \int P(\Delta^*|\Delta_l^*, \Delta_s^*) q(\Delta_l^*) q(\Delta_s^*) d\Delta_l^* d\Delta_s^*$  which leads to the following predictive mean and variance of the DGP:

$$\begin{aligned} \boldsymbol{\mu}_\Delta(\tilde{\mathbf{x}}^*) &= \mathbf{K}_{*M} \mathbf{K}_M^{-1} \mathbf{m}_{ul} + \mathbf{V}_{*M} \mathbf{V}_M^{-1} \mathbf{m}_{us} \\ \boldsymbol{\Sigma}_\Delta(\tilde{\mathbf{x}}^*) &= k_{**} + v_{**} - \mathbf{K}_{*M} \mathbf{K}_M^{-1} \mathbf{K}_{M*} - \mathbf{V}_{*M} \mathbf{V}_M^{-1} \mathbf{V}_{M*} \\ &+ \mathbf{K}_{*M} \mathbf{K}_M^{-1} \mathbf{S}_{ul} \mathbf{K}_M^{-1} \mathbf{K}_{M*} + \mathbf{V}_{*M} \mathbf{V}_M^{-1} \mathbf{S}_{us} \mathbf{V}_M^{-1} \mathbf{V}_{M*} \end{aligned} \quad (23)$$

The predictive distribution  $q(\Delta^*)$  of the latent function  $\Delta$  can be used for multi-step prediction over a given horizon as it is done in the MPC scheme in the next section. Furthermore, to update the predictive mean and variance of the short-term GP, one can first recursively update the posterior distribution  $q(\mathbf{u}_s)$  with (13) using online data points, then marginalize the posterior on  $\mathbf{u}_s$  as in (15), leading to  $q(\Delta^*)$ .

#### IV. GP-MPC FOR QUADCOPTERS

To design a stabilizing tracking controller for the system (2) subject to the robust satisfaction of state and control input constraints  $\mathcal{X} \subseteq \mathbb{R}^{12}$  and  $\mathcal{U} \subseteq \mathbb{R}^4$ , the following constrained optimal control problem is formulated as:

$$\begin{aligned} \min_{\mathbf{u}(k)} J &= \mathbb{E} \left[ l_T(\mathbf{x}_{H|k}, \mathbf{r}_{H|k}) + \sum_{i=0}^{H-1} l(\mathbf{x}_{i|k}, \mathbf{u}_{i|k}, \mathbf{r}_{i|k}) \right] \\ \text{s.t. } \mathbf{x}_{0|k} &= \mathbf{x}(k), \mathbf{u}_{0|k} = \mathbf{u}(k), \\ \mathbf{x}_{i+1|k} &= \mathbf{A} \mathbf{x}_{i|k} + \mathbf{B} \mathbf{u}_{i|k} + \mathbf{B}_d \Delta(\mathbf{x}_{i|k}, \mathbf{u}_{i|k}), \\ \mathbf{x}_{i|k} &\in \mathcal{X}, \mathbf{u}_{i|k} \in \mathcal{U}, \end{aligned} \quad (24)$$

where  $H$  is the prediction horizon, and  $\mathbf{r}(\cdot)$  represents the desired trajectory to be tracked.  $l(\cdot)$  and  $l_T(\cdot)$  are the stage and terminal cost functions, which are defined in a quadratic form with weight matrices  $\mathbf{Q}$ ,  $\mathbf{Q}_T$  and  $\mathbf{R}$ :  $l(\cdot) \triangleq \|\mathbf{x}_{i|k} - \mathbf{r}_{i|k}\|_{\mathbf{Q}}^2 + \|\mathbf{u}_{i|k}\|_{\mathbf{R}}^2$ , and  $l_T(\cdot) \triangleq \|\mathbf{x}_{H|k} - \mathbf{r}_{H|k}\|_{\mathbf{Q}_T}^2$ .

In (24), the model uncertainty  $\Delta(\mathbf{x}_{i|k}, \mathbf{u}_{i|k}) = \mathcal{N}(\Delta|\boldsymbol{\mu}_\Delta^k, \boldsymbol{\Sigma}_\Delta^k)$  is a stochastic term which is inferred by the DGP structure. This directly leads to a probabilistic prediction model  $P(\mathbf{x}_{i+1|k}) = \mathcal{N}(\boldsymbol{\mu}_x^{i+1}, \boldsymbol{\Sigma}_x^{i+1})$ . Assuming that the predictive state is time independent, then one can obtain the mean and variance of the prediction model as:

$$\begin{aligned} \boldsymbol{\mu}_x^{i+1} &= \mathbf{A} \boldsymbol{\mu}_x^i + \mathbf{B} \mathbf{u}_{i|k} + \mathbf{B}_d \boldsymbol{\mu}_\Delta^i, \\ \boldsymbol{\Sigma}_x^{i+1} &= \mathbf{A} \boldsymbol{\Sigma}_x^i \mathbf{A}^\top + \mathbf{B}_d \boldsymbol{\Sigma}_\Delta^i \mathbf{B}_d^\top \end{aligned} \quad (25)$$

It can be observed from (25) that the input for GP  $\Delta(\mathbf{x}_k, \mathbf{u}_k)$  becomes a random variable during the state propagation, which results in a non-Gaussian posterior and is intractable to be computed analytically. We adopt the exact moment matching approach [18], [19] to fit the posterior optimally with the first and second moments. One can use other moment matching methods to obtain the cost distribution as well. This allows us to express the stochastic cost function as a deterministic form:

$$\begin{aligned} J &= (\boldsymbol{\mu}_x^{H|k} - \mathbf{r}_{H|k})^\top \mathbf{Q}_T (\boldsymbol{\mu}_x^{H|k} - \mathbf{r}_{H|k}) + \text{tr}(\mathbf{Q}_T \boldsymbol{\Sigma}_x^{H|k}) + \\ & \sum_{i=0}^{H-1} ((\boldsymbol{\mu}_x^{i|k} - \mathbf{r}_{i|k})^\top \mathbf{Q} (\boldsymbol{\mu}_x^{i|k} - \mathbf{r}_{i|k}) + \mathbf{u}_{i|k}^\top \mathbf{R} \mathbf{u}_{i|k} + \text{tr}(\mathbf{Q} \boldsymbol{\Sigma}_x^{i|k})) \end{aligned} \quad (26)$$

Next, we discuss the satisfaction of the constraints on the state and the control input. Consider linear state constraints  $\mathbf{H} \mathbf{x}_{i|k} \leq \mathbf{h}$  with  $\mathbf{H}_j \mathbf{x}_{i|k} \leq h_j$ ,  $j = 1, \dots, m$ ,  $\mathbf{H} \in \mathbb{R}^{m \times 12}$  and  $\mathbf{h} \in \mathbb{R}^m$ . As  $\mathbf{x}_{i|k}$  is a random variable with mean  $\boldsymbol{\mu}_x^i$  and variance  $\boldsymbol{\Sigma}_x^i$ , we intend to satisfy the constraints in probabilistic sense, i.e. by the chance constraint:  $P(\mathbf{H} \mathbf{x}_{i|k} \leq \mathbf{h}) \geq \gamma$ . The quantile function  $\varpi(\gamma)$ , which can be seen as the inverse CDF, is utilized to convert the above chance constraints into deterministic ones. Denoting  $\mathbf{z} = \mathbf{H} \mathbf{x}_{i|k} - \mathbf{h}$ , then we have  $\mathbf{z} \sim \mathcal{N}(\mathbf{H} \boldsymbol{\mu}_x^{i|k} - \mathbf{h}, \mathbf{H} \boldsymbol{\Sigma}_x^{i|k} \mathbf{H}^\top)$ . Thus, according to the Galois inequalities, the chance constraint is equivalent to  $\varpi(\gamma) \leq 0$  and finally leads to

$$\mathbf{H} \boldsymbol{\mu}_x^{i|k} \leq \mathbf{h} - \varpi(\gamma) \sqrt{\mathbf{H} \boldsymbol{\Sigma}_x^{i|k} \mathbf{H}^\top}.$$

On the other hand, the control input constraints  $\mathcal{U}$  is simply defined as  $\mathbf{u}_{\min} \leq \mathbf{u}_{i|k} \leq \mathbf{u}_{\max}$ . Finally, the GP-MPC problem in this paper is formulated as:

$$\begin{aligned} \min_{\mathbf{u}(k)} & \quad (26) \\ \text{s.t. } \mathbf{x}_{0|k} &= \mathbf{x}(k), \mathbf{u}_{0|k} = \mathbf{u}(k) \\ \boldsymbol{\mu}_x^{i+1} &= \mathbf{A} \boldsymbol{\mu}_x^i + \mathbf{B} \mathbf{u}_i + \mathbf{B}_d \boldsymbol{\mu}_\Delta^i, \\ \boldsymbol{\Sigma}_x^{i+1} &= \mathbf{A} \boldsymbol{\Sigma}_x^i \mathbf{A}^\top + \mathbf{B}_d \boldsymbol{\Sigma}_\Delta^i \mathbf{B}_d^\top \\ \mathbf{H} \boldsymbol{\mu}_x^{i|k} &\leq \mathbf{h} - \varpi(\gamma) \sqrt{\mathbf{H} \boldsymbol{\Sigma}_x^{i|k} \mathbf{H}^\top} \\ \mathbf{u}_{\min} &\leq \mathbf{u}_{i|k} \leq \mathbf{u}_{\max} \end{aligned} \quad (27)$$

Because the stochastic cost has been transformed into a deterministic form, most nonlinear optimization algorithms can be utilized to solve the problem.

## V. SIMULATION STUDY

In this section, numerical simulation results are provided to illustrate the effectiveness of the proposed dual GP based MPC strategy for quadcopters. The parameters of the quadcopter model are considered according to [20]. The quadcopter is forced to track a helix-like trajectory in 3D space, which is described by:  $x(t) = 2 \sin(t)$ ,  $y(t) = 2 \cos(t)$ , and  $z(t) = \{2, 3, 4\}$  for every third of total operation time  $T$ . The sampling time  $T_s$  is 0.1 s. The discrete-time nonlinear system dynamics of the quadcopter are linearized and decomposed into two LTI systems, i.e., the outer-loop translational subsystem and the inner-loop rotational subsystem. To simplify the controller design and speed up the simulation, we adopt the PD controller for the rotational subsystem and use the MPC for the translational subsystem. It is worth mentioning that, despite the linearizing effect feedback control in the rotational subsystem driven by PD controller, the nonlinearities and additional dynamics experienced there affect the translational system, which interprets the necessity of a GP compensation. For the MPC scheme, the prediction horizon is chosen as  $H = 5$ , and the weight matrices in the cost function are:  $\mathbf{Q} = \text{diag}(1 \ 1 \ 20 \ 1 \ 1 \ 20)$ , and  $\mathbf{R} = \text{diag}(1 \ 1 \ 1)$ .

First, to construct the training data set, the quadcopter is enabled to track a random reference trajectory and explore the space as much as possible. In general, the data can also be collected by flying the quadcopter manually. We use the linear MPC as the exciting input for the quadcopter for 50s, and then collect the data with 20Hz, resulting in a training data set with size  $N = 1000$ . Furthermore, we employ a constant wind with speed  $[1 \ 3 \ -2]^T$  m/s in frame  $\mathcal{I}$  which occurs as a heading dependent uncertainty in dynamics (1) during the offline training phase, and a time-varying wind with speed  $[1 \ 3 \ -2]^T + \sin(\frac{\pi}{10}(t - 10)) * [-2 \ -3 \ 3]^T$  m/s, which appears as a time-varying orientation dependent uncertainty into the system at  $t = 10$  s. Then, a SVGP is trained offline with  $M = 20$  pseudo inputs, which gives the long-term GP for the first iteration. At the same time, the short-term GP is initialized as a zero-mean GP.

To show the advantage of the proposed DGP-MPC strategy, a baseline GP-MPC scheme without online recursive update is employed in the simulations. The simulation results are shown in Figs. 2 - 4. The trajectory tracking result of the proposed DGP-MPC after the 3rd iteration is depicted in Fig. 2. The tracking performance is adequate though the constant wind from 0s - 10s turns into time-varying wind at 10s. In addition, the DGP estimation results and error comparison among the baseline GPMPC, online recursive GPMPC and the proposed DGP-MPC are shown in Figs. 3 - 4. It is obvious that both the online GP and DGP can recursively update with the new measurement data after 10s, and DGP shows better performance on estimating the uncertain model owing to the "memory" data. Furthermore, to quantify the performance and tracking errors of the proposed controller and baseline controller, the average cost  $\bar{J}$  within the prediction horizon and the mean square error (MSE) are recorded in Table. 1.

One can see that comparing with the baseline controller,

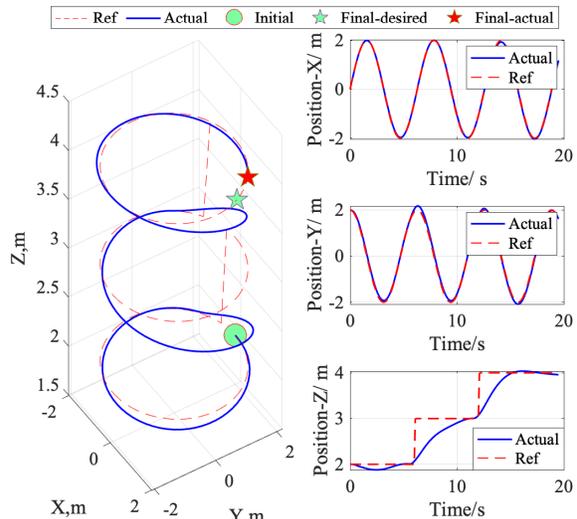


Fig. 2: Trajectory tracking with DGP-MPC (after the 3rd iteration).

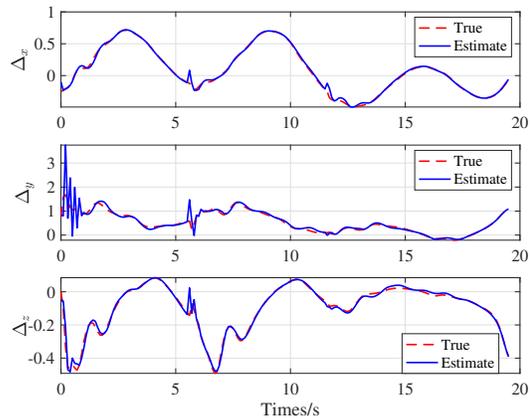


Fig. 3: DGP estimation results.

the performance of the proposed DGP-MPC strategy in the third iteration has been improved. The reason why the  $\bar{J}$  and MSE for baseline GP-MPC are smaller than DGP-MPC in the first iteration is that the estimation of  $\Delta$  needs to converge to the true value during the first seconds of online recursion in DGP-MPC. Moreover, comparing the  $\bar{J}$  and MSE within the iterations of DGP-MPC, one can observe that after incorporating new data into the long-term GP from mission to mission, the tracking error decreases obviously. The reason is that the long-term GP gradually captures more and more model uncertainties after the batch training.

## VI. CONCLUSIONS

This paper presents a novel Dual Gaussian Process based model predictive control for quadcopter trajectory tracking with model uncertainties and time-varying external disturbances. The knowledge of learned model is kept in the long-term GP while the short-term GP performs recursive online adaption to compensate the unlearned uncertainties during control operation. The DGP structure is not limited to MPC schemes, but can be also extended to any GP-based controllers which require both "memory" and "learning". Future work will focus on the real-time implementation of

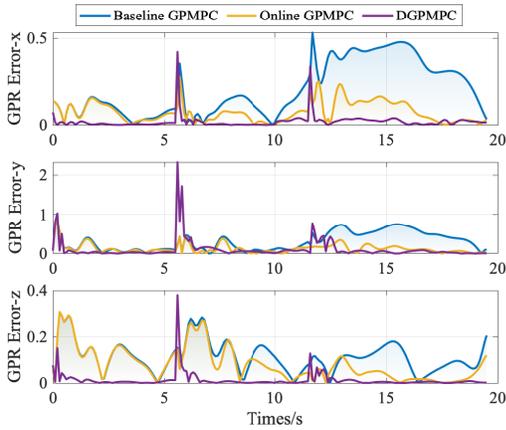


Fig. 4: GP regression error of three methods.

TABLE I: Quantitative evaluation of the two controllers

Controller	$\bar{J}$	MSE		
		$x$	$y$	$z$
Baseline GP-MPC	35.9730	0.0020	0.0053	0.0914
DGP-MPC 1st iter.	36.4046	0.0023	0.0062	0.0928
DGP-MPC 3rd iter.	33.3544	0.0011	0.0035	0.0825

the proposed strategy on Crazyflie nano quadcopters with VICON positioning system.

#### REFERENCES

- [1] Agha-mohammadi, Ali-akbar, N. K. Ure, J. P. How, and J. Vian, "Health aware stochastic planning for persistent package delivery missions using quadrotors," in *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2014, pp. 3389–3396.
- [2] S. Hayat, E. Yanmaz, and R. Muzaffar, "Survey on unmanned aerial vehicle networks for civil applications: A communications viewpoint," *IEEE Com. Surveys & Tutorials*, vol. 18, no. 4, pp. 2624–2661, 2016.
- [3] C. E. Rasmussen, "Gaussian processes in machine learning," in *Summer School on Machine Learning*. Springer, 2003, pp. 63–71.
- [4] T. Beckers, D. Kulić, and S. Hirche, "Stable Gaussian process based tracking control of Euler–Lagrange systems," *Automatica*, vol. 103, pp. 390–397, 2019.
- [5] J. Umlauft, T. Beckers, M. Kimmel, and S. Hirche, "Feedback linearization using Gaussian processes," in *Proc. of the IEEE 56th Conf. on Decision and Control*, 2017, pp. 5249–5255.
- [6] N. Lawrence, M. Seeger, and R. Herbrich, "Fast sparse Gaussian process methods: The informative vector machine," in *Proc. of the Conf. on Neural Information Processing Systems*, 2003, pp. 609–616.
- [7] C. Williams and M. Seeger, "Using the nyström method to speed up kernel machines," in *Proc. of the 14th Conf. on Neural Information Processing Systems*, 2001, pp. 682–688.
- [8] L. Csató, "Gaussian processes: iterative sparse approximations," Ph.D. dissertation, Aston University Birmingham, UK, 2002.
- [9] M. Seeger, "Bayesian Gaussian process models: PAC-Bayesian generalisation error bounds and sparse approximations," University of Edinburgh, Tech. Rep., 2003.
- [10] V. Tresp, "The generalized bayesian committee machine," in *Proc. of the 6th ACM Int. Conf. on Knowledge Discovery and Data Mining*, 2000, pp. 130–139.
- [11] E. Snelson and Z. Ghahramani, "Sparse Gaussian processes using pseudo-inputs," *Advances in Neural Information Processing Systems*, vol. 18, pp. 1257–1264, 2005.
- [12] J. Quinero-Candela and C. E. Rasmussen, "A unifying view of sparse approximate Gaussian process regression," *The Journal of Machine Learning Research*, vol. 6, pp. 1939–1959, 2005.
- [13] L. Hewing, J. Kabzan, and M. N. Zeilinger, "Cautious model predictive control using Gaussian process regression," *IEEE Transactions on Control Systems Technology*, vol. 28, no. 6, pp. 2736–2743, 2019.
- [14] J. Kabzan, L. Hewing, A. Liniger, and M. N. Zeilinger, "Learning-based model predictive control for autonomous racing," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3363–3370, 2019.

- [15] A. Carron, E. Arcari, M. Wermelinger, L. Hewing, M. Hutter, and M. N. Zeilinger, "Data-driven model predictive control for trajectory tracking with a robotic arm," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3758–3765, 2019.
- [16] G. Cao, E. M. Lai, and F. Alam, "Gaussian process based model predictive control for linear time varying systems," in *Proc. of the 14th IEEE Int. Workshop on Advanced Motion Control*, 2016, pp. 251–256.
- [17] M. Titsias, "Variational learning of inducing variables in sparse gaussian processes," in *Artificial Int. and Statistics*, 2009, pp. 567–574.
- [18] J. Quinero-Candela, A. Girard, and C. E. Rasmussen, "Prediction at an uncertain input for Gaussian processes and relevance vector machines-application to multiple-step ahead time-series forecasting," Technical University of Denmark, IMM-2003-18, Tech. Rep., 2003.
- [19] M. P. Deisenroth, *Efficient reinforcement learning using Gaussian processes*. KIT Scientific Publishing, 2010, vol. 9.
- [20] P. E. I. Pounds, "Design, construction and control of a large quadrotor micro air vehicle," Ph.D. dissertation, Australian National Univ., 2007.