Robust pose tracking control for a fully-actuated hexarotor UAV based on Gaussian processes

Tatsuya Ibuki †, Hiroto Yoshioka ‡ and Mitsuji Sampei ⊙

†School of Science and Technology, Meiji University, Kanagawa, Japan; ‡eSOL Co., Ltd., Tokyo, Japan; ⊙School of Engineering, Tokyo Institute of Technology, Tokyo, Japan

ABSTRACT

This paper presents a robust position/attitude tracking control method for a fully-actuated hexarotor unmanned aerial vehicle (UAV) based on Gaussian processes. Multirotor UAVs suffer from modeling errors due to their structure complexity and aerodynamical disturbances whose perfect mathematical formulation is intractable. To handle this issue, this paper incorporates a data-based learning technique with model-based control. The hexarotor UAV dynamical model, considering modelling errors and aerodynamic disturbances as unknown dynamics, is first derived. Gaussian process regression is next introduced as a learning method for the unknown dynamics, which provides probabilistic distributions of the predicted values. The predicted means are regarded as deterministic information and cancelled out by feedforward control inputs. The predicted variances are considered as the bounds of the model uncertainties with high probability, and a robust control method to ensure ultimate boundedness of the tracking control error is proposed for the uncertain system. The effectiveness of the proposed method is demonstrated via experiments with a self-developed hexarotor UAV testbed.

1. Introduction

A multirotor unmanned aerial vehicle (UAV) has become one of the most common vehicles for these decades, thanks to its high mobility in three-dimensional (3D) space. Its industrial application is expected in various fields including agriculture monitoring [1], building inspection [2], load transportation [3], and work with a robotic arm [4]. In standard structures of multirotor UAVs, each rotor is placed at the vertex of a regular polygon and in the same direction as shown in Figure 1(a). This same direction property causes underactuation, that is, UAVs cannot move in the horizontal direction without tilting their bodies. This motion limitation is often undesired, e.g. in the aforementioned application. To guarantee the full actuation property, i.e. to control the 6D pose (3D position and 3D attitude) individually, special structures have been recently developed [5–9]. One of the common approaches is to introduce tilted rotors as shown in Figure 1(b). Here, at least six rotors are required to individually control the 6D pose, and multirotor UAVs with six rotors are called hexarotors.

However, this kind of fully-actuated structure brings a new technical issue that tilted rotors cause aerodynamical interference between the rotors, which does not appear in standard structures. Multirotor UAV structures also have the model complexity such as rotor dynamics and aerodynamics associated with thrust generation. Therefore, perfect mathematical formulation of fully-actuated multirotor UAV dynamics is intractable. Moreover, when the UAV flies near buildings (walls) or the ground, counter wind, called ground effects, strongly affects the UAV motion, and modelling of this effect is also quite challenging. In view of these facts, this paper incorporates a data-based learning method for these kinds of unknown dynamics with model-based control.

Multirotor UAV motion control considering the effect of aerodynamical interference from surrounded environments is investigated in [10–12]. The works [10,11] tackle landing control problems taking account of ground effects, and [12] proposes flight control methods near a wall. They achieve desired control tasks based on adaptive robust control techniques but focus only on standard multirotor UAV structures. Since fully-actuated UAVs with tilted rotors suffer from different types of aerodynamical interferences, we have to design other new controllers. Additionally, while the methods in [10–12] succeed in attenuating aerodynamical effects, the characteristics of those unknown dynamics are not directly estimated. Compared with such an adaptive control approach, Gaussian Process Regression (GPR) [13] directly provides unknown dynamics prediction as a probabilistic distribution with mean and variance information. The prediction results are also provided in the closed form.
based on current states, which enables us to easily utilize the predicted values for control. With hope for this versatility, this paper introduces GPR as a learning method for unknown dynamics of the fully-actuated UAVs.

Another advantage of GPR is that it is a nonparametric regression model, i.e. the structure of a target model is not restricted. Thanks to these versatile properties, GPR has been recently utilized as a statistical modelling methodology in robotics and control engineering fields [14–20]. Especially, similarly to these works, the authors in [14, 15] employ GPR for multirotor UAV control. The work [14] approximates an evaluation index of control performance by GPR, and the optimization of controller gains is achieved while guaranteeing the stability of the UAV. The authors in [15] guarantee the safety during learning process by introducing control barrier functions, where GPR is used for learning the disturbances of the UAV and the improvement of tracking control performance is confirmed via simulation. The work [16] also designs control barrier functions for safe flight control based on GPR learning results, where more general uncertainties are learned and the predicted variance modifies the control barrier function values. Whereas these studies handle standard structures of multirotor UAVs, we expect that GPR can be also applied to fully-actuated structures thanks to its versatility. Moreover, since the predicted variance can be regarded as uncertainties on the system, GPR is useful for robust controller design.

This paper presents a new robust pose tracking control method of a fully-actuated multirotor UAV based on GPR for learning of its unknown dynamics. The proposed technique explicitly utilizes the predicted mean and variance information for robust controller design. By regarding the predicted means by GPR as deterministic information and cancelling them out by feedforward inputs, the control performance is improved. On the other hand, the predicted variances are considered as the bounds of the model uncertainties with high probability, and a robust control method to ensure ultimate boundedness of the tracking control error is proposed for the uncertain system. The main contribution is to show the utility of the proposed scheme via experimental verification with a self-developed fully-actuated hexarotor UAV testbed. Here, we demonstrate learning of unknown dynamics caused by modelling errors, aerodynamical interference between the tilted rotors, and ground effects.

2. Problem settings

2.1. Fully-actuated hexarotor structure

Throughout this paper, we consider the hexarotor UAV structure shown in Figure 2, which is proposed by our previous works [8, 9]. This structure guarantees the full actuation of the hexarotor UAV, namely, can generate 6D wrench (3D force and 3D torque) inputs independently.

Let the world frame be $\Sigma_w$ and body-fixed frame $\Sigma_b$ whose origin coincides with the centre of mass of the vehicle. The position of each rotor $i \in \{1, \ldots, 6\}$ in $\Sigma_b$ is denoted by $r_i \in \mathbb{R}^3$, the tilting angle around the direction of $r_i$ is represented by $\gamma_i \in (-\pi/2, \pi/2)$, and the direction of the thrust force generated by each rotor is denoted by $s_i \in \mathbb{R}^3$ ($\|s_i\| = 1$). We also represent each counter torque coefficient by $c_i \in \mathbb{R}$, where $c_i > 0$ $(c_i < 0)$ means that the direction of the counter torque generated by rotor $i$ is the same as (opposite to) that of the thrust force [21].

Then, the fully-actuated hexarotor UAV structure in Figure 2 is formulated as follows:

\begin{align}
\|r_i\| &= R > 0, \quad r_i \perp s_i \quad \forall \ i \in \{1, \ldots, 6\}, \quad (1a) \\
\gamma_{i+3} &= -\gamma_i, \quad s_{i+3} = s_i \quad \forall \ i \in \{1, 2, 3\}, \quad (1b) \\
\gamma_1 &= \gamma_2 = \gamma_3 = -\gamma_4 = -\gamma_5 = -\gamma_6 = \gamma \in \left(0, \frac{\pi}{2}\right), \quad (1c) \\
s_1 \neq s_2, \ s_2 \neq s_3, \ s_3 \neq s_1, \quad (1d) \\
c_1 = c_2 = c_3 = -c_4 = -c_5 = -c_6 = c > 0. \quad (1e)
\end{align}

Here, (1a)–(1c) mean the symmetry of the structure, (1d) guarantees the full-actuation property, and
The rotation matrix is defined as $R = \begin{bmatrix} \cos \psi & \cos \theta & 0 \\ -\cos \psi \tan \theta & \sin \psi \tan \theta & 1 \end{bmatrix}$.

Remark 2.1: Although we provide a special structure here for reasons of brevity and according to the experimental testbed used in Section 5, the subsequent discussion holds for any kinds of multi-rotor UAVs as far as they are fully actuated.

2.2. Hexarotor model with unknown dynamics

In this work, we consider translational force and angular velocity as the control inputs as in [22, 23]. The usage of the angular velocity inputs is motivated by the fact that not torque inputs but angular velocity ones are used for many multirotor UAVs now in use, i.e. desired angular velocity is relatively easily achieved by a well-tuned local controller. In fact, we can also directly command the angular velocity inputs in the experiments in Section 5.

The position of the hexarotor UAV in $\Sigma_w$ is denoted by $p = [x\; y\; z]^T \in \mathbb{R}^3$, and the attitude is represented by XYZ Euler angles $\xi = [\phi\; \theta\; \psi]^T \in \mathbb{R}^3$ [24]. The force and angular velocity inputs of the vehicle in $\Sigma_b$, generated by the six rotors, are written by $F \in \mathbb{R}^3$ and $\Omega \in \mathbb{R}^3$, respectively. Then, the nominal model of the UAV is given as follows [21]:

$$m \ddot{p} = R_{wb}(\xi)F - mge_3,$$
$$\dot{\xi} = J(\xi)\Omega.$$  \hspace{1cm} (2a)

By denoting the total states as $x = [p^T \; \dot{p}^T \; \xi^T]^T \in \mathbb{R}^9$ and the total inputs as $u = [F^T \; \Omega^T]^T \in \mathbb{R}^6$, dynamics (2) can be written by the following state equation:

$$\dot{x} = f(\bar{x}) + g(\bar{x})u.$$  \hspace{1cm} (3)

Here, $\bar{x} \in \mathbb{R}^9$ indicates the state of the nominal system.

Let us next consider unknown dynamics for (3). In this work, the modelling error of the nominal system (3), aerodynamical interference between the rotors, and the ground effects are mainly considered as the unknown dynamics, and suppose that they are dependent on the state $x$. In this case, by regarding the total modelling error from the nominal model (3) as the unknown dynamics: $d(x) = \dot{x} - (f(x) + g(x)u) \in \mathbb{R}^9$, we finally obtain the following state equation as the actual system:

$$\dot{x} = f(x) + g(x)u + d(x).$$  \hspace{1cm} (4)

Here, we also suppose that $d(x) = [0_1^T \; d_1(x) \cdot \cdot \cdot d_6(x)]^T$ holds because our focus is learning the modelling error and aerodynamical interference that mainly affect the position dynamics rather than the position kinematics. ($0_n \in \mathbb{R}^n$ is the $n$-dimensional zero vector.)

Remark 2.2: It is mathematically possible to consider the unknown dynamics dependent also on the input $u$, i.e. $d(x, u)$. However, this setting causes time inconsistency in the implementation stage because $d(x, u)$ is used for determining $u$ in the robust controller. On the other hand, since the control input $u$ is formed by a state-feedback-type controller, we expect that the input information is indirectly used for learning of the unknown dynamics $d$. In fact, we show the present setting works in the experimental verification in Section 5.

Figure 2. Fully-actuated hexarotor UAV.
2.3. Control objective

The objective of this paper is to propose a robust tracking controller for given reference pose trajectories so that the tracking errors to the trajectories are bounded. Here, the probability distributions of the unknown dynamics \( \mathbf{d}(\mathbf{x}) \) are learned by GPR and utilized for the controller design.

3. Unknown dynamics learning via Gaussian process regression

3.1. Gaussian process

A Gaussian process (GP) is a kind of probability process whose distribution is given by random variables dependent on its inputs. Suppose that a function \( \delta(\cdot) : \mathbb{R}^n \rightarrow \mathbb{R} \) follows a GP. Then, given any finite input variables \( \chi_i \in \mathbb{R}^n, i = 1, \ldots, N \), the joint probability distribution of \( \delta(\chi_i) \) is provided by

\[
\begin{bmatrix}
\delta(\chi_1) \\
\vdots \\
\delta(\chi_N)
\end{bmatrix} \sim \mathcal{N}
\begin{bmatrix}
\mu(\chi_1) \\
\vdots \\
\mu(\chi_N)
\end{bmatrix},
\begin{bmatrix}
k(\chi_1, \chi_1) & \cdots & k(\chi_1, \chi_N) \\
\vdots & \ddots & \vdots \\
k(\chi_N, \chi_1) & \cdots & k(\chi_N, \chi_N)
\end{bmatrix}
\]

Here, \( \mathcal{N}(\cdot, \cdot) \) is the Gaussian distribution, \( \mu(\cdot) \in \mathbb{R} \) the mean function, and \( k(\cdot, \cdot) > 0 \) covariance one.

The fact that \( \delta(\cdot) \) follows a GP is represented by

\[
\delta(\chi) \sim \mathcal{GP}(\mu(\chi), k(\chi, \chi')).
\]

The function \( k(\chi, \chi') \), also called the kernel function, represents the covariance of \( \delta(\chi) \) and \( \delta(\chi') \). This paper employs the following squared exponential kernel (see [13] for a review of different kernels):

\[
k(\chi, \chi') = \sigma_l^2 \exp \left( -\frac{1}{2} (\chi - \chi')^T L^{-2} (\chi - \chi') \right),
\]

where \( \sigma_l > 0 \) and \( L = \text{diag}(l_1, \ldots, l_n) \in \mathbb{R}^{n \times n}, l_i > 0 \ \forall \ i \in \{1, \ldots, n\} \) are hyperparameters. This kernel implies that the more similar the values of \( \chi \) and \( \chi' \) are to each other, the stronger the correlation between \( \delta(\chi) \) and \( \delta(\chi') \) is. In addition to the choice of kernel functions, the way to choose the hyperparameters is the design freedom of GPR.

3.2. Gaussian process regression

By using state/input data of the UAV, given (measured) by preliminary experiments, as training data, we predict \( d_i(\mathbf{x}) \), \( i \in \{1, \ldots, 6\} \) of (4) under the assumption that they follow GPs. For the simplicity of the learning process, each \( d_i(\mathbf{x}) \) is individually predicted by supposing that they are independent of each other, i.e. \( d_i(\mathbf{x}) \sim \mathcal{GP}(\mu_i(\mathbf{x}), k_i(\mathbf{x}, \mathbf{x}')) \) holds for all \( i \in \{1, \ldots, 6\} \). In this section, we briefly explain the GPR method for one element of \( \mathbf{d}(\mathbf{x}) \), simply denoted by \( \mathbf{d}(\mathbf{x}) \in \mathbb{R}^9 \).

Suppose that the training data \( \mathcal{D} = \{\mathbf{x}_i, \hat{\mathbf{d}}(\mathbf{x}_i)\}_{i=1}^N \) is given by preliminary experiments. Here, keeping the measurement noise for the training data in mind, we consider \( \hat{\mathbf{d}}(\mathbf{x}_i) \sim \mathcal{N}(\mathbf{d}(\mathbf{x}_i), \sigma_n^2) \). We note that the variance \( \sigma_n^2 \) of the measurement noise can be also handled as a hyperparameter of the GPR. Then the posterior predictive distribution of \( \mathbf{d}(\mathbf{x}) \) for any query state \( \mathbf{x}_s \in \mathbb{R}^9 \) can be calculated by the prior distribution of \( \mathbf{d}(\mathbf{x}) \) and the joint distribution with it.

Since we have no information about the prior distribution of the unknown dynamics, we simply suppose that the mean of the prior distribution is zero. In this case, the joint distribution of the training data \( \hat{\mathbf{d}}(\mathbf{x}_i), i = 1, \ldots, N \) and the predicted value \( \hat{\mathbf{d}}(\mathbf{x}_s) \) is given as follows:

\[
\begin{bmatrix}
\hat{\mathbf{d}}_D \\
\mathbf{d}(\mathbf{x}_s)
\end{bmatrix} \sim \mathcal{N}
\begin{bmatrix}
0_{N+1}, & [I + \sigma_n^2 I_N]^{-1} \mathbf{k}^{T}(\mathbf{x}_s) \\
[I + \sigma_n^2 I_N]^{-1} \mathbf{k}(\mathbf{x}_s, \mathbf{x}_s + \sigma_n^2)
\end{bmatrix}
\]

Here, \( \hat{\mathbf{d}}_D = [\hat{\mathbf{d}}(\mathbf{x}_1) \cdots \hat{\mathbf{d}}(\mathbf{x}_N)]^T \in \mathbb{R}^N \) is the training data, \( \mathbf{k} \in \mathbb{R}^{N \times N} \) the covariance matrix of \( \hat{\mathbf{d}}_D \) with the elements \( \mathbf{k}(i, j) = \mathbf{k}(\mathbf{x}_i, \mathbf{x}_j), i, j \in \{1, \ldots, N\} \), \( \mathbf{k}(\mathbf{x}_s) = [\mathbf{k}(\mathbf{x}_s, \mathbf{x}_1) \cdots \mathbf{k}(\mathbf{x}_s, \mathbf{x}_N)]^T \in \mathbb{R}^N \) the covariance vector of \( \hat{\mathbf{d}}_D \) and \( \mathbf{d}(\mathbf{x}_s) \), and \( \mathbf{k}(\mathbf{x}_s, \mathbf{x}_s) \) the covariance of \( \mathbf{d}(\mathbf{x}_s) \) itself, i.e. the prior covariance. Then, we get the following predicted distribution from the joint distribution (5) [13]:

\[
\mathcal{P}(\mathbf{d}(\mathbf{x}_s) | \mathcal{D}) \sim \mathcal{N}(\mu(\mathbf{x}_s), \sigma^2(\mathbf{x}_s)),
\]

\[
\begin{align*}
\mu(\mathbf{x}_s) &= \mathbf{k}(\mathbf{x}_s)^T (K + \sigma_n^2 I_N)^{-1} \hat{\mathbf{d}}_D, \\
\sigma^2(\mathbf{x}_s) &= \mathbf{k}(\mathbf{x}_s, \mathbf{x}_s) + \sigma_n^2 - \mathbf{k}(\mathbf{x}_s)^T (K + \sigma_n^2 I_N)^{-1} \mathbf{k}(\mathbf{x}_s).
\end{align*}
\]

Notice here that the predicted mean \( \mu(\mathbf{x}_s) \) and variance \( \sigma^2(\mathbf{x}_s) \) are given in the closed form, which is one of the versatile properties of GPR, that is, the predicted results can be easily utilized for control.

The hyperparameters of the GPR are determined by solving the optimization problem:

\[
\max_{\theta \in \mathbb{R}^{11}} \log \mathcal{P}(\hat{\mathbf{d}}_D | \mathbf{d}_D, \theta).
\]

Here, \( \mathbf{d}_D = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\} \) is the training data, and \( \theta = [\sigma_l I_1 \cdots \sigma_l I_9] \in \mathbb{R}^{11} \). This optimization problem can be numerically solved by standard gradient methods [13].
4. Robust pose tracking control based on Gaussian processes

4.1. Unknown dynamics representation as uncertainties

Considering the probabilistic characteristics of the predicted values by the GPR, we represent the unknown dynamics as the uncertainties with some offsets as follows:

\[
\mathbf{d}(x) = \begin{bmatrix} 0_1 \\
\mu(x) + \beta \Delta \mathbf{\sigma}(x) \end{bmatrix}.
\]  

(6)

Here, \( \mu(x) = [\mu_1(x) \cdots \mu_6(x)]^T \in \mathbb{R}^6 \) and \( \mathbf{\sigma}(x) = [\sigma_1(x) \cdots \sigma_6(x)]^T \in \mathbb{R}^6 \) are the stick vectors of the means and variances predicted by the GPR, respectively. \( \Delta = \text{diag}(\Delta_1, \ldots, \Delta_6) \in \mathbb{R}^{6 \times 6}, \Delta_i \in [-1, 1] \ \forall \ i \in \{1, \ldots, 6\} \) and \( \beta > 0 \) is the parameter to determine the confidence region of the existence of the uncertainties.

The representation (6) means that we deal with the predicted mean \( \mu(x) \) as the deterministic information and the probabilistic characteristic is handled by \( \beta \Delta \mathbf{\sigma}(x) \). In this work, we set \( \beta = 3 \), i.e. the 99.73% confidence region (only in the sense of the given training data).

4.2. Feedback linearization based on predicted means

We first denote the desired pose trajectories by \( p_d(t) = [x_d(t) y_d(t) z_d(t)]^T \in \mathbb{R}^3 \) and \( \mathbf{\xi}_d(t) = [\phi_d(t) \theta_d(t) \psi_d(t)]^T \in \mathbb{R}^3 \). Then, defining the outputs as the tracking errors:

\[
\begin{bmatrix} y_p \\
y_q \end{bmatrix} = \begin{bmatrix} p - p_d \\
\mathbf{\xi} - \mathbf{\xi}_d \end{bmatrix} \in \mathbb{R}^6,
\]  

(7)

we obtain the following output dynamics from (2), (4), (6), and (7):

\[
\begin{bmatrix} \dot{y}_p \\
\dot{y}_q \end{bmatrix} = \begin{bmatrix} \frac{\rho h_p}{m} - g e_3 - \bar{p}_d \\
f_{\Omega} - \mathbf{\dot{\xi}}_d \end{bmatrix} + \mu(x) + \beta \Delta \mathbf{\sigma}(x). 
\]  

(8)

Let us next consider the following input transformation with the new input \( \bar{u} = [\tilde{F}^T \tilde{\Omega}]^T \in \mathbb{R}^6 \):

\[
u = \begin{bmatrix} \frac{m\tilde{F}}{m} \\
\frac{1}{J} \end{bmatrix} \begin{bmatrix} \frac{\bar{F} + g e_3 + \bar{p}_d - \mu_1(x) - \mu_2(x)}{\mu_3(x)} \\
\frac{\mu_4(x)}{\mu_5(x)} - \mathbf{\Omega}_d \end{bmatrix}.
\]  

(9)

Define the new state as \( \mathbf{z} = [y_p^T y_q^T \mathbf{\xi}_d^T]^T \in \mathbb{R}^9 \). Then the variance information of the unknown dynamics can be rewritten by \( \mathbf{\sigma}(x) = \mathbf{\sigma}(t, z) \) because the desired pose trajectories \( p_d \) and \( \mathbf{\xi}_d \) are given by functions of time \( t \).

Substituting (9) into (8) yields the following uncertain linearized state equation:

\[
\dot{\mathbf{z}} = \begin{bmatrix} 0_3 & I_3 & 0_3 \\
0_3 & 0_3 & 0_3 \\
0_3 & 0_3 & 0_3 \end{bmatrix} \mathbf{z} + \begin{bmatrix} O_3 & 0_3 \\
0_3 & I_3 & 0_3 \\
0_3 & 0_3 & I_3 \end{bmatrix} \begin{bmatrix} \bar{u} \\
\beta \Delta \mathbf{\sigma}(t, z) \end{bmatrix}.
\]  

(10)

Here, \( O_n \in \mathbb{R}^{n \times n} \) is the \( n \times n \) zero matrix.

Remark 4.1: (10) is the linear system with the bounded uncertainties determined by the parameter \( \beta \) and the variance information \( \mathbf{\sigma}(x) \) predicted by the GPR. Here, the predicted mean \( \mu(x) \) is cancelled out by the feedforward input in (9). Therefore, standard robust control methods can be applied as shown in the following section.

4.3. Robust pose tracking control

We finally design a pose tracking controller to guarantee robust stability against the uncertainty \( \beta \Delta \mathbf{\sigma}(t, z) \). Based on the Lyapunov redesign technique [25], we propose the following tracking control law:

\[
\begin{align*}
\bar{u} &= -K_c z + v, \\
v &= \begin{cases} -\|\beta \mathbf{\sigma}(t, z)\|_{2B^T P_z} \frac{B^T P_z}{\|B^T P_z\|} & \text{if } \|\beta \mathbf{\sigma}(t, z)\|_{2B^T P_z} \geq \epsilon \\
-\|\beta \mathbf{\sigma}(t, z)\|_{2B^T P_z}^2 \frac{2B^T P_z}{\epsilon} & \text{otherwise.}
\end{cases}
\end{align*}
\]  

(11)

Here, \( K_c \in \mathbb{R}^{6 \times 9} \) is the feedback gain matrix to stabilize the origin of (10), i.e. designed so that \( A - BK_c \) is Hurwitz. \( P \in \mathbb{R}^{9 \times 9} \) is the solution of the following Lyapunov equation for any positive semidefinite matrix \( Q \in \mathbb{R}^{9 \times 9} \):

\[
(A - BK_c)^T P + P(A - BK_c) = -Q,
\]  

and \( \epsilon > 0 \) is the design parameter to determine the tracking performance.

Then, we have the following theorem.

Theorem 4.1: Given smooth trajectories \( p_d, \tilde{p}_d, \mathbf{\xi}_d \) for the system (4), (6), the control law (9), (11) obtains the uniformly ultimate bound of the tracking error \( z \), i.e. there exist a class \( K\mathcal{L} \) function \( \alpha \) and finite time \( T > 0 \) such that

\[
\|z(t)\| \leq \alpha(\|z(0)\|, t) \quad \forall t < T,
\]  

\[
\|z(t)\| \leq \sqrt{\frac{\lambda_{\max}(P) \epsilon}{2\lambda_{\min}(P) \lambda_{\min}(Q)}} \quad \forall t \geq T.
\]  

Here, \( \lambda_{\max}(\cdot) (\lambda_{\min}(\cdot)) \) is the maximum (minimum) eigenvalue of a symmetric matrix.

Proof: See the proof of Theorem 14.3 in [25].
Remark 4.2: Theorem 4.1 seems to indicate that any small ultimate bound can be obtained by making the parameter $\epsilon$ small, but actually, we have to take care of high gain feedback in such a case. We also note that the statement of Theorem 4.1 holds only in the sense of the given training data.

5. Experimental results

5.1. Experimental environment

The utility of the proposed approach is demonstrated via experiments with the self-developed hexarotor UAV testbed shown in Figure 3. The testbed mainly consists of the flight controller “Pixhawk,” motors “Cobra CM-2204/28 Motor,” propellers “HQprop 5 × 3 Carbon Composite,” and frames made of the aluminium alloy “A2017.” The physical parameters are summarized in Table 1.

The experimental environment is shown in Figure 4. The states of the hexarotor UAV are obtained from the measurement by the motion capture system with the motion capture cameras “OptiTrack Prime13” and numerical differentiation techniques, where the velocity information is obtained by applying the following linear predictor as a noise reduction filter:

$$\dot{p}[k] = 0.9\dot{p}[k] - 0.1[p[k] - p[k - 1]] / 0.01.$$  

Here, $k \in \mathbb{Z}$ is the step number to show each sampling period of the experiments. The control input is calculated by the PC with the CPU “Intel®Core™i7-7700K @4.20GHz” and RAM 64.0 GB, and sent to the vehicle via the wireless communication device “DFT, D8R-II PLUS.” The sampling time of the state measurement is 10 ms and that of sending the control input command is 23 ms.

The prediction of the unknown dynamics $d(x)$ via the GPR is conducted by using 200 data points obtained by the preliminary experiments. Here, the 200 data points are extracted from 1500 data points by $k$-means clustering [26]. The hyperparameter $\theta$ is optimized by GPML Matlab Code [27].

5.2. Experiment 1

We first demonstrate the effectiveness of the proposed approach via standard flight to mainly consider the modelling error of the nominal model and the aerodynamical interference between the tilted rotors. For the desired trajectories: $[p_d(t)^T \xi_d(t)^T]^T = [-0.5 \cos(0.1 \pi t) 0.5 \sin(0.1 \pi t) 0.5 0 0 0]^T$ [m, rad], the control law (9), (11) with

$$K_c = \begin{bmatrix} 2.5I_3 & 3.5I_3 & O_3 \\ 3.5I_3 & O_3 & 4.6I_3 \end{bmatrix}, \quad Q = I_9, \quad \epsilon = 0.1 \quad (12)$$

is applied to the hexarotor UAV testbed. As the comparison, we also apply the nominal controller: (9) without $\mu(x)$ and $\dot{u} = -K_c z$ with $K_c$ in (12). This $K_c$ is empirically tuned to obtain good tracking performance for the nominal controller. In the preliminary experiments to collect data for the GPR learning, we modify the desired trajectory to cancel out the offset error yielded by the nominal controller and moreover apply small deviations to obtain the data around the true desired trajectory. In addition, we utilize the desired trajectories replacing $z_d(t) = 0.5$ with $z_d(t) = 0.3$ and $z_d(t) = 0.4$ so that we confirm the utility of the present scheme regardless of the ground effects from the experimental field.

The experimental results are depicted in Figures 5 and 6. Figure 5 shows the time responses of the pose by the nominal controller and Figure 6 illustrates those by the proposed control law. We see the steady position tracking errors from Figure 5(a), and Figure 5(b) shows that the attitude errors are oscillating within around 1.5 deg. On the other hand, we see from Figure 6(a) that the position tracking is much improved, and Figure 6(b) shows the attitude errors are attenuated within 1.0 deg. These results provide the utility of the proposed approach.

5.3. Experiment 2

Let us next demonstrate the effectiveness of the proposed method via landing flight to also consider the ground effects from the experimental field. For the desired trajectories: $[p_d(t)^T \xi_d(t)^T]^T = [0 0 0.1(8 - t) 0 0 0]^T$ [m, rad], the same controllers as in Experiment 1 are applied to the hexarotor UAV testbed. In
Figure 4. Experimental environment.

Figure 5. Results of Experiment 1: Nominal controller: (a) position in $\Sigma_w$ and (b) attitude in $\Sigma_w$.

The preliminary experiments to collect data for the GPR learning, we also modify the desired trajectory to cancel out the offset error yielded by the nominal controller and apply small deviations $x_d(t) = \pm 0.1$ and $y_d(t) = \pm 0.1$ to obtain the data around the true desired trajectory. In this experiment, legs with the height 0.1 m are attached to the vehicle so that $z = 0.1$ means landing of the UAV.

Figure 6. Results of Experiment 1: Proposed controller: (a) position in $\Sigma_w$ and (b) attitude in $\Sigma_w$.

The experimental results are depicted in Figures 7 and 8. Figure 7(a) shows that there exist steady position tracking errors and the hexarotor UAV cannot land even at $t = 8 \text{[s]}$ when the desired height becomes 0 m (less than the landing height 0.1 m). We also see from Figure 7(b) that when the vehicle approaches to the ground, especially in the region $z \leq 0.25$, the attitude errors are much oscillating and the maximum size of the
errors is around 3.0 deg. These poor results imply that the ground effects are not small enough to be neglected. On the other hand, we see from Figure 8(a) that the position tracking is much improved and the hexarotor UAV successfully lands on the field. Figure 8(b) also shows that the attitude errors are attenuated within ±1.8 deg even when the vehicle is near the ground. These results show the utility of the proposed approach.

Since the current UAV testbed is designed aiming only at adding the ability to horizontally move without tilting its body, it cannot significantly change its roll/pitch angle. Therefore, one of the future directions of this work is to conduct experimental verification to evaluate the robustness against large body tilting.

6. Conclusion

This paper presented a robust pose tracking control method of a fully-actuated hexarotor UAV based on GPR for learning of its unknown dynamics. In the proposed scheme, the means predicted by the GPR are regarded as the deterministic values and explicitly cancelled out by feedforward control inputs to improve the tracking performance. On the other hand, the predicted variances are considered as the bounds of the model uncertainties with high probability. Then, we proposed a robust control law based on the Lyapunov redesign technique for the uncertain system. The utility of the proposed method was demonstrated via two kinds of experiments with a self-developed hexarotor UAV testbed. The first experiment was the standard flight to show the effectiveness of the proposed approach for modelling errors and aerodynamical interference between the tilted rotors. The second experiment was the landing flight to show the utility also for ground effects.

Although the present experimental results show the utility of incorporating GPR, the proposed method works only for repeating process as in the current experimental settings since the learning is conducted offline with preliminary experimental results. Therefore, the main future work is to extend the present approach to incorporate online GPR as in [28, 29], where the main issues are to guarantee the stability of the UAV during the learning process and computational cost for the online implementation.
Disclosure statement

No potential conflict of interest was reported by the author(s).

Funding

This work is supported in part by JSPS KAKENHI under 21K04113 and in part by JSPS KAKENHI under 21H01348.

Notes on contributors

Tatsuya Ibuki received the B.Eng., M.Eng., and Ph.D.Eng. degrees from Tokyo Institute of Technology in 2008, 2010, and 2013, respectively. He was a research fellow with the Japan Society for the Promotion of Science from 2012 to 2013, an assistant professor with the Department of Systems and Control Engineering, Tokyo Institute of Technology from 2013 to 2020, and a visiting scholar with the School of Electrical and Computer Engineering, Georgia Institute of Technology in 2019. Since 2020, he has been an associate professor with the Department of Electronics and Bioinformatics, Meiji University. His research interests include cooperative control of robotic networks, fusion of control theory and machine learning, and vision-based estimation and control.

Hiroto Yoshioka received the B.Eng. and M.Eng degrees in Control Engineering in 2017 and 2019, respectively. He is currently engaged in software development using Robot Operation System (ROS) at eSOL Co., Ltd.

Mitsuji Sampei received the B.Eng., M.Eng., and Dr.Eng. degrees in Control Engineering from Tokyo Institute of Technology in 1983, 1985, and 1987, respectively. From 1987 to 1993, he was a research associate and then an associate professor in Chiba University. Since 1993, he has been with Tokyo Institute of Technology where he is currently a professor in the Department of Systems and Control Engineering. His current research interests are in the field of nonlinear system theory, including underactuated systems, nonholonomic systems, safety-critical control, robotics, and H-infinity control theory.

ORCID

Tatsuya Ibuki  http://orcid.org/0000-0003-1525-1024
Mitsuji Sampei  http://orcid.org/0000-0002-3478-8399

References

[1] Das J, Cross G, Qu C, et al. Devices, systems, and methods for automated monitoring enabling precision agriculture. 2015 IEEE International Conference on Automation Science. IEEE; 2015. p. 462–469.
[2] Vazquez-Nicolas J, Zamora E, Gonzalez-Hernandez I, et al. Towards automatic inspection: crack recognition based on quadrotor UAV-taken images. 2018 International Conference on Unmanned Aircraft System. IEEE; 2018. p. 654–659.
[3] Liang X, Fang Y, Sun N, et al. Nonlinear hierarchical control for unmanned quadrotor transportation systems. IEEE Trans Ind Electr. 2018 Apr;65(4):3395–3405.
[4] Kim S, Choi S, Kim HJ. Aerial manipulation using a quadrotor with a two DOF robotic arm. 2013 IEEE/RSJ International Conference Intelligent Robots Systems. IEEE; 2013. p. 4990–4995.
[5] Micheletto G, Ryll M, Franchi A. Fundamental actuation properties of multicopters: force-moment decoupling and fail-safe robustness. IEEE Trans Robot. 2018 Mar;34(3):702–715.
[6] Rajappa S, Ryll M, Blihff HH, et al. Modeling, control and design optimization for a fully-actuated hexrotor aerial vehicle with tilted propeller. 2015 IEEE International Conference Robot Automation. IEEE; 2015. p. 4006–4013.
[7] Giribet JL, Sanchez-Pena RS, Ghersin AS. Analysis and design of a tilted rotor hexacopter for fault tolerance. IEEE Trans Aeros Electron Syst. 2016 Aug;52(4):1555–1567.
[8] Tadokoro Y, Ibuki T, Sampei M. Classification and structural evaluation of fully-actuated hexrotor UAV’s. 2018 American Control Conference. IEEE; 2018. p. 1945–1950.
[9] Tadokoro Y, Ibuki T, Sampei M. Maneuverability analysis of a fully-actuated hexrotor UAV considering tilt angles and arrangement of rotors. IFAC-PapersOnLine. 2017;50(1):8981–8986.
[10] Lee D, Ryan T, Kim HJ. Autonomous landing of a VTOL UAV on a moving platform using image-based visual servoing. 2012 IEEE International Conference Robot Automation. IEEE; 2012. p. 971–976.
[11] Hu B, Lu L, Mishra S. Fast, safe and precise landing of a quadrotor on an oscillating platform. 2015 American Control Conference. IEEE; 2015. p. 3836–3841.
[12] Lee D, Awan A, Kim S, et al. Adaptive control for a VTOL UAV operating near a wall. AIAA Guid Navig Control Conference 2012. AIAA; 2012. no. 4835.
[13] Rasmussen CE, Williams CKI. Gaussian processes for machine learning. London: The MIT Press; 2006.
[14] Berkenkamp F, Schoellig AP, Krause A. Safe controller optimization for quadrotors with Gaussian processes. 2016 IEEE International Conference Robot Automation. IEEE; 2016. p. 491–496.
[15] Wang L, Theodoron EA, Egerstedt M. Safe learning of quadrotor dynamics using barrier certificates. 2018 IEEE International Conference Robot Automation. IEEE; 2018. p. 2460–2465.
[16] Khan M, Ibuki T, Chatterjee A. Safety uncertainty in control barrier functions using Gaussian processes. 2021 IEEE International Conference Robot Automation. IEEE; 2021. p. 6003–6009.
[17] Berkenkamp F, Schoellig AP. Safe and robust learning control with Gaussian processes. 2015 European Control Conferences. IEEE; 2015. p. 2496–2501.
[18] Akametalu AK, Fisac JF, Gillula JH, et al. Reachability-based safe learning with Gaussian processes. 53rd IEEE Conference Decis Control. IEEE; 2014. p. 1424–1431.
[19] Ostafew CJ, Schoellig AP, Barfoot TD. Conservative to confident: treating uncertainty robustly within learning-based control. 2015 IEEE International Conference Robot Automation. IEEE; 2015. p. 421–427.
[20] Deisenroth MP, Fox D, Rasmussen CE. Gaussian processes for data-efficient learning in robotics and control. IEEE Trans Patt Anal Mach Int. 2015 Feb;37(2):408–423.
[21] Garcia Carrillo LR, Dzul Lopez AE, Lozano R, et al. Quad rotorcraft control: vision-based hovering and navigation. London: Springer-Verlag; 2013.
[22] Hehn M, D’Andrea R. A flying inverted pendulum. 2011 IEEE International Conference Robot Automation. IEEE; 2011. p. 763–770.
[23] Endo M, Ibuki T, Sampei M. Collision-free formation control for quadrotor networks based on distributed
quadratic programs. 2019 American Control Conference. IEEE; 2019. p. 3335–3340.

[24] Murray RM, Li Z, Sastry SS. A mathematical introduction to robotic manipulation. Boca Raton (FL): CRC Press; 1994.

[25] Khalil HK. Nonlinear systems. 3rd ed. Upper Saddle River (NJ): Prentice Hall; 2002.

[26] Bishop CM. Pattern recognition and machine learning. New York (NY): Springer-Verlag; 2006.

[27] Rasmussen CE, Nickisch H. Documentation for GPML Matlab code version 4.2. http://www.gaussianprocess.org/gpml/code/matlab/doc/ (last accessed 2022-09-13).

[28] Huber MF. Recursive Gaussian process: on-line regression and learning. Patt Recogn Lett. 2014;45:85–91.

[29] Seo K, Yamakita M. Nonlinear time-varying system identification with recursive Gaussian process. 2017 American Control Conference. IEEE; 2017. p. 825–830.