Noname manuscript No.
(will be inserted by the editor)

Gaussian Process Model Predictive Control of An Unmanned Quadrotor

Gang Cao · Edmund M-K Lai · Fakhrul Alam

Received: date / Accepted: date

Abstract The Model Predictive Control (MPC) trajectory tracking problem of an unmanned quadrotor with input and output constraints is addressed. In this article, the dynamic models of the quadrotor are obtained purely from operational data in the form of probabilistic Gaussian Process (GP) models. This is different from conventional models obtained through Newtonian analysis. A hierarchical control scheme is used to handle the trajectory tracking problem with the translational subsystem in the outer loop and the rotational subsystem in the inner loop. Constrained GP based MPC are formulated separately for both subsystems. The resulting MPC problems are typically nonlinear and non-convex. We derived a GP based local dynamical model that allows these optimization problems to be relaxed to convex ones which can be efficiently solved with a simple active-set algorithm. The performance of the proposed approach is compared with an existing unconstrained Nonlinear Model Predictive Control (NMPC). Simulation results show that the two approaches exibit similar trajectory tracking performance. However, our approach has the advantage of incorporating constraints on the control inputs. In addition, our approach only requires 20% of the computational time for NMPC.

Keywords Quadrotor Trajectory Tracking · Model Predictive Control · Gaussian Process

1 Introduction

The quadrotor helicopter (or quadrotor for short) is an aerial vehicle with vertical take-off and landing capabilities. It has received a lot of interests recently due to

Gang Cao
School of Engineering and Advanced Technology, Massey University, Auckland, New Zealand
E-mail: g.cao@massey.ac.nz

Edmund M-K Lai
Department of Information Technology and Software Engineering, Auckland University of Technology, Auckland, New Zealand

Fakhrul Alam
School of Engineering and Advanced Technology, Massey University, Auckland, New Zealand

its simplicity, maneuverability, and payload capabilities [2,1]. It has been used in various military and civilian tasks [31,12].

Trajectory tracking is one of the basic functions performed by a quadrotor in autonomous flight. Designing a control system to perform this function is challenging because the quadrotor's dynamics are highly nonlinear and are subjected to random external disturbances. Several control approaches have previously been investigated with varying degrees of success. They include linear techniques such as Proportional-Integral-Derivative (PID) and Linear-Quadratic Regulator (LQR) control [4], as well as nonlinear techniques such as sliding mode [29] and backstepping control [21]. More recently, due to the conceptual simplicity, Model Predictive Control (MPC) techniques have been used in [34,2] based on the linearised model and in [1] based on the nonlinear model. Moreover, physical constraints on the system inputs and outputs, which is important for quadrotors, could easily be included as appropriate penalty terms in the cost function that is used to compute the optimal control.

The performance of MPC is highly dependent on how accurately the model describes the dynamics of the system being controlled. Conventionally, dynamical models are derived from first principles through Newton-Euler [45] or Euler-Lagrange based formalisms [4]. Alternatively, empirical input-output data could be collected from a real, working quadrotor. These data could then be used to construct a Fuzzy Model (FM) [18] or an Artificial Neural Network (ANN) model [41, 11]. This data-driven approach has the advantage that unknown dynamics that are not considered by Newtonian analysis could be captured by the empirical observations. However, it is difficult to evaluate the quality of these FM and ANN models. Gaussian Process (GP) modelling is an alternative data-driven technique based on Bayesian theory. Compared to ANN and FM, a major advantage is that the quality of obtained GP model can be directly evaluated by GP variances which are naturally computed during the modelling and prediction processes. GP based technique has recently been used to learn the flight model of Unmanned Aerial Vehicle (UAV) [19,20] and quadrotors [3,8].

The cost functions used in early GP based MPC problems are deterministic even though the GP models are probabilistic [25,24,26,15]. This issue has been addressed recently in [22,6,8] where the expectation of the cost function is used instead, as proposed in [30]. However, these works did not take into consideration any constraint on system inputs and outputs. In addition, a computationally efficient method is required to solve the resulting GP based MPC optimization problem which is usually nonlinear and non-convex.

In this article, a hierarchical control scheme is applied to the trajectory tracking problem of a quadrotor, where a translational subsystem forms the outer loop and a rotational subsystem is in the inner loop [34,2]. Each subsystem is independently modelled by a GP model. We propose a GP based MPC control scheme, referred to as GPMPC, solve the resulting two MPC tracking problems. It tackles the issues mentioned above regarding the objective function and computational efficiency. The performance of GPMPC is evaluated by simulations on two nontrivial trajectories.

2 Quadrotor System Modelling Using GP

The quadrotor can be viewed as a 6 Degree-of-Freedom (DOF) rigid body with generalized coordinates $\mathbf{q} = [x, y, z, \phi, \theta, \psi]^T \in \mathbb{R}^6$, where x, y, z denotes the quadrotor's positions w.r.t. earth-fixed frame (E-frame) and ϕ, θ, ψ represents quadrotor's attitudes w.r.t. body-fixed frame (B-Frame). Motion is controlled by a main thrust U_1 and three torques U_2, U_3 and U_4. Thus it is an underactuated system. Furthermore, the dynamical model of the quadrotor is defined by the state-space function $\ddot{\mathbf{q}} = f_{\mathbf{q}}(\mathbf{q}, \dot{\mathbf{q}}, U_1, U_2, U_3, U_4)$ which is usually nonlinear [34]. In order to simplify the control of the quadrotor, the system is typically decomposed into two subsystems – a translational subsystem and a rotational subsystem. Let the system state of the translational subsystem be $\mathbf{x}^{\xi} = [x, \dot{x}, y, \dot{y}, z, \dot{z}]^T \in \mathbb{R}^6$ and its control be $\mathbf{u}^{\xi} = [U_1, u_x, u_y]^T \in \mathbb{R}^3$. The dynamics of this subsystem can be described by [34]

$$\dot{\mathbf{x}}^{\xi} = f_{\xi}\left(\mathbf{x}^{\xi}, \mathbf{u}^{\xi}\right) + \boldsymbol{\epsilon}^{\xi} \tag{1}$$

where $f_{\xi} : \mathbb{R}^6 \times \mathbb{R}^3 \to \mathbb{R}^6$ is nonlinear and is usually corrupted by white noises $\boldsymbol{\epsilon}^{\xi} \in \mathbb{R}^6$. u_x and u_y are two intermediate controls to actuate the translational subsystem and are given by

$$\begin{aligned} u_x &= \cos\phi \sin\theta \cos\psi + \sin\phi \sin\psi \\ u_y &= \cos\phi \sin\theta \sin\psi - \sin\phi \cos\psi \end{aligned} \tag{2}$$

Similarly, let $\mathbf{x}^{\eta} = [\phi, \dot{\phi}, \theta, \dot{\theta}, \psi, \dot{\psi}]^T \in \mathbb{R}^6$ and $\mathbf{u}^{\eta} = [U_2, U_3, U_4]^T \in \mathbb{R}^3$ be the state and control for the rotational subsystem. Its system equation is given by [34]

$$\dot{\mathbf{x}}^{\eta} = f_{\eta}\left(\mathbf{x}^{\eta}, \mathbf{u}^{\eta}\right) + \boldsymbol{\epsilon}^{\eta} \tag{3}$$

where $f_{\eta} : \mathbb{R}^6 \times \mathbb{R}^3 \to \mathbb{R}^6$ is another nonlinear function and $\boldsymbol{\epsilon}^{\eta} \in \mathbb{R}^6$ represents the white noise.

2.1 GP Modelling

The system equations (1) and (3) of both subsystems can be expressed in the following general form in the discrete-time domain by

$$\mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_k) + \mathbf{w}_k \tag{4}$$

where $\mathbf{x}_k \in \mathbb{R}^n$ denotes an n-dimensional state vector and $\mathbf{u}_k \in \mathbb{R}^m$ represents an m-dimensional input vector at the sampling time k. $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ is a discrete nonlinear function, and $\mathbf{w}_k \in \mathbb{R}^n$ is Gaussian white noise. To learn such an unknown function $f(\cdot)$ using GP modelling techniques, a natural choice for the model inputs and outputs are the state-control tuple $\tilde{\mathbf{x}}_k = (\mathbf{x}_k, \mathbf{u}_k) \in \mathbb{R}^{n+m}$ and the next state $\mathbf{x}_{k+1}$ respectively. However, in practice, the difference $\Delta\mathbf{x}_k = \mathbf{x}_{k+1} - \mathbf{x}_k \in \mathbb{R}^n$ is usually smaller less than the values of $\mathbf{x}_k$. Thus it is more advantageous to use $\Delta\mathbf{x}_k$ as the model output instead [9].

A GP model is completely specified by its mean and covariance function [35]. Assuming that the mean of the model input $\tilde{\mathbf{x}}_k$ is zero, the squared exponential covariance is given by $\mathbf{K}(\tilde{\mathbf{x}}_i, \tilde{\mathbf{x}}_j) = \sigma_s^2 \exp(-\frac{1}{2}(\tilde{\mathbf{x}}_i - \tilde{\mathbf{x}}_j)^T \boldsymbol{\Lambda}(\tilde{\mathbf{x}}_i - \tilde{\mathbf{x}}_j)) + \sigma_n^2$, where i

and j denote two sampling time steps. The parameters σ_s^2, σ_n^2 and the entries of matrix $\boldsymbol{\Lambda} \in \mathbb{R}^{(n+m)\times(n+m)}$ (usually is a diagonal matrix) are referred to as the hyperparameters $\boldsymbol{\theta}$ of a GP model. Given D training inputs $\tilde{\mathbf{X}} = [\tilde{\mathbf{x}}_1, \cdots, \tilde{\mathbf{x}}_D] \in \mathbb{R}^{(n+m)\times D}$ and their corresponding training targets $\mathbf{y} = [\Delta\mathbf{x}_1, \cdots, \Delta\mathbf{x}_D]^T \in \mathbb{R}^{nD}$, the joint distribution between $\mathbf{y}$ and a test target $\Delta\mathbf{x}_k^*$ corresponding to the test input $\tilde{\mathbf{x}}_k^*$ at sampling time k is assumed to follow a Gaussian distribution. That is

$$p\begin{pmatrix}\mathbf{y}\\ \Delta\mathbf{x}_k^*\end{pmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{matrix}\mathbf{K}(\tilde{\mathbf{X}},\tilde{\mathbf{X}}) + \sigma_n\mathbf{I} & \mathbf{K}(\tilde{\mathbf{X}},\tilde{\mathbf{x}}_k^*)\\ \mathbf{K}(\tilde{\mathbf{x}}_k^*,\tilde{\mathbf{X}}) & \mathbf{K}(\tilde{\mathbf{x}}_k^*,\tilde{\mathbf{x}}_k^*)\end{matrix}\right) \tag{5}$$

where $\mathcal{N}(\cdot)$ denotes a multivariate Gaussian distribution and $\mathbf{0} \in \mathbb{R}^{nD}$ is a zero vector. In addition, the posterior distribution over the observations can be obtained by restricting the joint distribution to only contain those targets that agree with the observations. This is achieved by conditioning the joint distribution on the observations, and results in the predictive mean and variance function as follows [35]

$$m(\tilde{\mathbf{x}}_k^*) = E_f[\Delta\mathbf{x}_k^*] = \mathbf{K}(\tilde{\mathbf{x}}_k^*,\tilde{\mathbf{X}})\mathbf{K}_\sigma^{-1}\mathbf{y} \tag{6a}$$

$$\begin{aligned}\sigma^2(\tilde{\mathbf{x}}_k^*) = Var_f[\Delta\mathbf{x}_k^*] &= \mathbf{K}(\tilde{\mathbf{x}}_k^*,\tilde{\mathbf{x}}_k^*)\\ &- \mathbf{K}(\tilde{\mathbf{x}}_k^*,\tilde{\mathbf{X}})\mathbf{K}_\sigma^{-1}\mathbf{K}(\tilde{\mathbf{X}},\tilde{\mathbf{x}}_k^*)\end{aligned} \tag{6b}$$

where $\mathbf{K}_\sigma = \mathbf{K}(\tilde{\mathbf{X}},\tilde{\mathbf{X}}) + \sigma_n\mathbf{I}$. The state at the next sampling time $k+1$ also follows a Gaussian distribution. That is

$$p(\mathbf{x}_{k+1}) \sim \mathcal{N}(\boldsymbol{\mu}_{k+1}, \boldsymbol{\Sigma}_{k+1}) \tag{7}$$

where

$$\boldsymbol{\mu}_{k+1} = \mathbf{x}_k + m(\tilde{\mathbf{x}}_k^*) \tag{8a}$$

$$\boldsymbol{\Sigma}_{k+1} = \sigma^2(\tilde{\mathbf{x}}_k^*) \tag{8b}$$

Typically, the hyperparameters of the GP model are learned by maximizing the log-likelihood function given by

$$\begin{aligned}\log p(\mathbf{y}|\tilde{\mathbf{X}},\boldsymbol{\theta}) = &- \frac{1}{2}\mathbf{y}^T\mathbf{K}_\sigma^{-1}\mathbf{y} - \frac{1}{2}\log\left|\mathbf{K}_\sigma^{-1}\right|\\ &- \frac{D}{2}\log(2\pi)\end{aligned} \tag{9}$$

This results in a nonlinear non-convex optimization problem that is traditionally solved by using Conjugate Gradient (CG) or Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithms.

2.2 Uncertainty propagation

With the GP model obtained, one-step-ahead predictions can be made by using (6) and (8). When multiple-step predictions are required, the conventional way is to iteratively perform multiple one-step-ahead predictions using the estimated mean values. However, this process does not take into account the uncertainties introduced by each successive prediction. This issue has been shown to be important in time-series predictions [14].

The uncertainty propagation problem can be dealt with by assuming that the joint distribution of the training inputs is uncertain and follows a Gaussian distribution. That is,

$$p(\tilde{\mathbf{x}}_k) = p\big(\mathbf{x}_k, \mathbf{u}_k\big) \sim \mathcal{N}(\tilde{\boldsymbol{\mu}}_k, \tilde{\boldsymbol{\Sigma}}_k) \tag{10}$$

with mean and variance given by

$$\tilde{\boldsymbol{\mu}}_k = [\boldsymbol{\mu}_k, E[\mathbf{u}_k]]^T \tag{11a}$$

$$\tilde{\boldsymbol{\Sigma}}_k = \begin{bmatrix} \boldsymbol{\Sigma}_k & Cov[\mathbf{x}_k, \mathbf{u}_k] \\ Cov[\mathbf{u}_k, \mathbf{x}_k] & Var[\mathbf{u}_k] \end{bmatrix} \tag{11b}$$

where $Cov[\mathbf{x}_k, \mathbf{u}_k] = E[\mathbf{x}_k\mathbf{u}_k] - \boldsymbol{\mu}_k E[\mathbf{u}_k]$. Here, $E[\mathbf{u}_k]$ and $Var[\mathbf{u}_k]$ are the mean and variance of the system controls.

The exact predictive distribution of the training target could then be obtained by integrating over the training input distribution:

$$p(\Delta\mathbf{x}_k^*) = \int p(f(\tilde{\mathbf{x}}_k^*)|\tilde{\mathbf{x}}_k^*)p(\tilde{\mathbf{x}}_k^*)d\tilde{\mathbf{x}}_k^* \tag{12}$$

However, this integral is analytically intractable. Numerical solutions can be obtained using Monte-Carlo simulation techniques. In [5], a moment-matching based approach is proposed to obtain an analytical Gaussian approximation. The mean and variance at an uncertain input can be obtained through the laws of iterated expectations and conditional variances respectively [9]. They are given by

$$m(\tilde{\mathbf{x}}_k^*) = E_{\tilde{\mathbf{x}}_k^*}\Big[E_f\left[\Delta\mathbf{x}_k^*\right]\Big] \tag{13a}$$

$$\sigma^2(\tilde{\mathbf{x}}_k^*) = E_{\tilde{\mathbf{x}}_k^*}\Big[Var_f\left[\Delta\mathbf{x}_k^*\right]\Big] + Var_{\tilde{\mathbf{x}}_k^*}\Big[E_f\left[\Delta\mathbf{x}_k^*\right]\Big] \tag{13b}$$

Equation (8) then becomes

$$\boldsymbol{\mu}_{k+1} = \boldsymbol{\mu}_k + m(\tilde{\mathbf{x}}_k^*) \tag{14a}$$

$$\begin{aligned}\boldsymbol{\Sigma}_{k+1} = &\boldsymbol{\Sigma}_k + \sigma^2(\tilde{\mathbf{x}}_k^*) \\ &+ Cov\big[\mathbf{x}_k, \Delta\mathbf{x}_k\big] + Cov\big[\Delta\mathbf{x}_k, \mathbf{x}_k\big]\end{aligned} \tag{14b}$$

The computational complexity of GP inference using (13) is $\mathcal{O}(D^2n^2(n+m))$ which is quite high. Hence, GP is normally only suitable for problems with limited dimensions (under 12 as suggested by most publications) and limited size of training data. For problems with higher dimensions, sparse GP approaches [33] are often used.

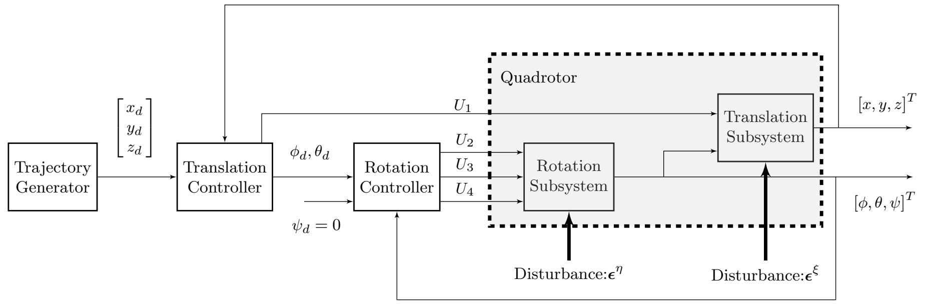


Fig. 1: The Overall Control Scheme for Quadrotor

3 Control Problem Formulation

3.1 MPC Problem for Subsystems

With the quadrotor system decomposed into two subsystems, a hierarchical structure as shown in Figure 1 can be used for the controller [34,2]. In the outer loop, the translational subsystem is controlled to follow a sequence of desired positions $[x_d, y_d, z_d]^T$. The optimal control U_1 and two intermediate controls u_x and u_y are obtained by minimizing the tracking errors. With $\psi_d = 0$, the desired attitudes θ_d and ϕ_d can be obtained using (2). Then, the rotational subsystem's attitudes $[\phi, \theta, \psi]^T$ are tuned to achieve the given target values in the inner loop. By minimizing attitude errors, the optimal controls U_2, U_3 and U_4 can be obtained. Finally, the optimal control inputs U_1, U_2, U_3 and U_4 are applied to the quadrotor.

For a horizon of $H \geq 1$, the discrete MPC trajectory tracking problem in the outer loop is given by

$$\min_{\mathbf{u}^\xi(\cdot)} \sum_{i=1}^{H} \left\{ \left\| \mathbf{x}^\xi_{k+i} - \mathbf{r}^\xi_{k+i} \right\|^2_{\mathbf{Q}^\xi} + \left\| \mathbf{u}^\xi_{k+i-1} \right\|^2_{\mathbf{R}^\xi} \right\} \tag{15a}$$

$$\text{s.t.} \quad \mathbf{x}^\xi_{k+i+1} = f_1(\mathbf{x}^\xi_{k+i}, \mathbf{u}^\xi_{k+i-1}) \tag{15b}$$

$$\mathbf{x}^\xi_{\min} \leq \mathbf{x}^\xi_{k+i} \leq \mathbf{x}^\xi_{\max} \tag{15c}$$

$$\mathbf{u}^\xi_{\min} \leq \mathbf{u}^\xi_{k+i-1} \leq \mathbf{u}^\xi_{\max} \tag{15d}$$

where the $f_1(\cdot)$ represents the GP model of the translational subsystem. $\|\cdot\|_{\mathbf{Q}^\xi}$ and $\|\cdot\|_{\mathbf{R}^\xi}$ denote the two 2-norms weighted by positive definite matrices $\mathbf{Q}^\xi$ and $\mathbf{R}^\xi$ respectively. $\mathbf{x}^\xi_{k+i}$ and $\mathbf{u}^\xi_{k+i-1}$ are the system states and control inputs, and $\mathbf{r}^\xi_{k+i} = [x_{d,k+i}, y_{d,k+i}, z_{d,k+i}]^T$ denotes the desired positions at time $k+i$. In addition, $\mathbf{x}^\xi_{\max} \geq \mathbf{x}^\xi_{\min}$ and $\mathbf{u}^\xi_{\max} \geq \mathbf{u}^\xi_{\max}$ are the upper and lower bounds of the system states and control inputs respectively.

In the same way, for the inner loop, the discrete MPC optimization problem is given by

$$\min_{\mathbf{u}^{\eta}(\cdot)} \sum_{i=1}^{H} \left\{ \left\| \mathbf{x}^{\eta}_{k+i} - \mathbf{r}^{\eta}_{k+i} \right\|^2_{\mathbf{Q}^{\eta}} + \left\| \mathbf{u}^{\eta}_{k+i-1} \right\|^2_{\mathbf{R}^{\eta}} \right\} \tag{16a}$$

$$\text{s.t.} \quad \mathbf{x}^{\eta}_{k+i+1} = f_2(\mathbf{x}^{\eta}_{k+i}, \mathbf{u}^{\eta}_{k+i-1}) \tag{16b}$$

$$\mathbf{x}^{\eta}_{\min} \leq \mathbf{x}^{\eta}_{k+i} \leq \mathbf{x}^{\eta}_{\max} \tag{16c}$$

$$\mathbf{u}^{\eta}_{\min} \leq \mathbf{u}^{\eta}_{k+i-1} \leq \mathbf{u}^{\eta}_{\max} \tag{16d}$$

where $f_2(\cdot)$ represents the GP model of the rotational subsystem.

Problems (15) and (16) can be rewritten in the following general form:

$$\mathbf{V}^*_k = \min_{\mathbf{u}(\cdot)} \mathcal{J}(\mathbf{x}_k, \mathbf{u}_{k-1}, \mathbf{r}_k) \tag{17a}$$

$$\text{s.t.} \ \mathbf{x}_{k+i|k} = f(\mathbf{x}_{k+i-1|k}, \mathbf{u}_{k+i-1}) \tag{17b}$$

$$\mathbf{x}_{\min} \leq \mathbf{x}_{k+i|k} \leq \mathbf{x}_{\max} \tag{17c}$$

$$\mathbf{u}_{\min} \leq \mathbf{u}_{k+i-1} \leq \mathbf{u}_{\max} \tag{17d}$$

$$i = 1, \cdots, H$$

with the quadratic cost function

$$\begin{aligned} &\mathcal{J}(\mathbf{x}_k, \mathbf{u}_{k-1}, \mathbf{r}_k) \\ &= \sum_{i=1}^{H} \left\{ \left\| \mathbf{x}_{k+i} - \mathbf{r}_{k+i} \right\|^2_{\mathbf{Q}} + \left\| \mathbf{u}_{k+i-1} \right\|^2_{\mathbf{R}} \right\} \end{aligned} \tag{18}$$

It should be noted that the control horizon is assumed to be equal to the prediction horizon H in this paper. In the rest of this article, the cost function $\mathcal{J}(\mathbf{x}_k, \mathbf{u}_{k-1}, \mathbf{r}_k)$ shall be rewritten as $\mathcal{J}(\mathbf{x}_k, \mathbf{u}_{k-1})$ for brevity.

3.2 MPC with GP Models

When the dynamical system is described by a GP model, the original problem (17) becomes a stochastic one [16]. The minimization should be performed over the expected value of $\mathcal{J}(\cdot)$ instead and the constraints are modified as follows.

$$\mathbf{V}^*_k = \min_{\mathbf{u}(\cdot)} E\big[\mathcal{J}(\mathbf{x}_k, \mathbf{u}_{k-1})\big] \tag{19a}$$

$$\text{s.t.} \quad p(\mathbf{x}_{k+1}|\mathbf{x}_k) \sim \mathcal{N}(\boldsymbol{\mu}_{k+1}, \boldsymbol{\Sigma}_{k+1}) \tag{19b}$$

$$\mathbf{u}_{\min} \leq \mathbf{u}_{k+i-1} \leq \mathbf{u}_{\max} \tag{19c}$$

$$p\big\{\mathbf{x}_{k+i|k} \geq \mathbf{x}_{\min}\big\} \geq \eta \tag{19d}$$

$$p\big\{\mathbf{x}_{k+i|k} \leq \mathbf{x}_{\max}\big\} \geq \eta \tag{19e}$$

where η denotes a confidence level. For $\eta = 0.95$, the chance constraints (19d) and (19e) are equivalent to

$$\boldsymbol{\mu}_{k+i} - 2\boldsymbol{\Sigma}_{k+i} \geq \mathbf{x}_{\min} \tag{20a}$$

$$\boldsymbol{\mu}_{k+i} + 2\boldsymbol{\Sigma}_{k+i} \leq \mathbf{x}_{\max} \tag{20b}$$

Given (18),

$$\begin{aligned} & E\big[\mathcal{J}(\mathbf{x}_k, \mathbf{u}_{k-1})\big] \\ & = E\Big[\sum_{i=1}^{H} \Big\{ \big\|\mathbf{x}_{k+i} - \mathbf{r}_{k+i}\big\|_{\mathbf{Q}}^2 + \big\|\mathbf{u}_{k+i-1}\big\|_{\mathbf{R}}^2 \} \Big] \\ & = \sum_{i=1}^{H} E\Big[\big\|\mathbf{x}_{k+i} - \mathbf{r}_{k+i}\big\|_{\mathbf{Q}}^2 + \big\|\mathbf{u}_{k+i-1}\big\|_{\mathbf{R}}^2\Big] \\ & = \sum_{i=1}^{H} \Big\{ E\Big[\big\|\mathbf{x}_{k+i} - \mathbf{r}_{k+i}\big\|_{\mathbf{Q}}^2\Big] + E\Big[\big\|\mathbf{u}_{k+i-1}\big\|_{\mathbf{R}}^2\Big] \Big\} \end{aligned} \tag{21}$$

In practice, the controls are deterministic. Hence, $E\big[\mathbf{u}_k^2\big] = \mathbf{u}_k^2$ and (21) becomes

$$\begin{aligned} & E\big[\mathcal{J}(\mathbf{x}_k, \mathbf{u}_{k-1})\big] \\ & = \sum_{i=1}^{H} \Big\{ E\Big[\big\|\mathbf{x}_{k+i} - \mathbf{r}_{k+i}\big\|_{\mathbf{Q}}^2\Big] + \big\|\mathbf{u}_{k+i-1}\big\|_{\mathbf{R}}^2 \Big\} \\ & = \sum_{i=1}^{H} \Big\{ \big\|\boldsymbol{\mu}_{k+i} - \mathbf{r}_{k+i}\big\|_{\mathbf{Q}}^2 + \big\|\mathbf{u}_{k+i-1}\big\|_{\mathbf{R}}^2 + \mathit{trace}\big(\mathbf{Q}\boldsymbol{\Sigma}_{k+i}\big) \Big\} \\ & = h\left(\boldsymbol{\mu}_k, \mathbf{u}_{k-1}\right) \end{aligned} \tag{22}$$

The elaboration of (22) can be found in Appendix A. With this cost function and the state constraints (20), we are able to relax the original stochastic optimization problem (19) to a deterministic nonlinear one. Furthermore, the resulting deterministic cost function involves the model variance $\boldsymbol{\Sigma}$. This allows model uncertainties to be explicitly included in the computation of optimized controls.

4 Proposed Solution

Solving the constrained MPC optimization problem (22) with state constraints (20) is not simple because it is typically nonlinear and non-convex. Solving non-convex problems due to they are computationally complicated and have multiple local optima. This significantly limits the application of MPC in real world problems. An effective and efficient solution method is therefore very important [36, 43, 38, 39, 37, 40]. A conventional approach is to use derivative-based methods such as Sequential Quadratic Programming (SQP) and interior-point algorithms [10]. When the derivatives of the cost function are unavailable or are too difficult to compute, they could be iteratively approximated by using sampling methods [28, 27]. An alternative solution is to use evolutionary algorithms such as Particle Swarm Optimization (PSO) [44]. A more complete review of solution methods can be found in [10].

In this section, we present our proposed solution which is by local linearization. This allows the original problem to be relaxed into a convex one which can then be solved efficiently by active-set methods.

4.1 GP Based Local Dynamical Model

There are many different ways by which a GP model could be linearised. In [3], a GP based local dynamical model allows standard robust control methods to be used on the partially unknown system directly. Another GP based local dynamical model is proposed in [32] to integrate GP model with dynamic programming. In these two cases, the nonlinear optimization problems considered are *unconstrained.*

In this paper, we propose a different GP based local model. In this local model, $\mathbf{x}_k$ in (4) is replaced by $\mathbf{s}_k = [\boldsymbol{\mu}_k, \mathbf{vec}(\sqrt{\boldsymbol{\Sigma}_k})]^T \in \mathbb{R}^{n+n^2}$. Here, $\mathbf{vec}(\cdot)$ denotes the vectorization of a matrix [1] Hence (4) becomes

$$\mathbf{s}_{k+1} = \mathcal{F}'\left(\mathbf{s}_k, \mathbf{u}_k\right) \tag{23}$$

Linearizing at the operating point $(\mathbf{s}^*_k, \mathbf{u}^*_k)$ where $\mathbf{s}^*_k = [\boldsymbol{\mu}^*_k, \mathbf{vec}(\sqrt{\boldsymbol{\Sigma}^*_k})]^T$, we have

$$\Delta\mathbf{s}_{k+1} = \frac{\partial\mathcal{F}'}{\partial\mathbf{s}_k}\Delta\mathbf{s}_k + \frac{\partial\mathcal{F}'}{\partial\mathbf{u}_k}\Delta\mathbf{u}_k \tag{24}$$

Here, $\Delta\mathbf{s}_k = \mathbf{s}_k - \mathbf{s}^*_k$ and $\Delta\mathbf{u}_k = \mathbf{u}_k - \mathbf{u}^*_k$. The Jacobian matrices are

$$\frac{\partial\mathcal{F}'}{\partial\mathbf{s}_k} = \begin{bmatrix} \frac{\partial\boldsymbol{\mu}_{k+1}}{\partial\boldsymbol{\mu}_k} & \frac{\partial\boldsymbol{\mu}_{k+1}}{\partial\sqrt{\boldsymbol{\Sigma}_k}} \\ \frac{\partial\sqrt{\boldsymbol{\Sigma}_{k+1}}}{\partial\boldsymbol{\mu}_k} & \frac{\partial\sqrt{\boldsymbol{\Sigma}_{k+1}}}{\partial\sqrt{\boldsymbol{\Sigma}_k}} \end{bmatrix} \in \mathbb{R}^{(n+n^2)\times(n+n^2)} \tag{25a}$$

$$\frac{\partial\mathcal{F}'}{\partial\mathbf{u}_k} = \begin{bmatrix} \frac{\partial\boldsymbol{\mu}_{k+1}}{\partial\mathbf{u}_k} \\ \frac{\partial\sqrt{\boldsymbol{\Sigma}_{k+1}}}{\partial\mathbf{u}_k} \end{bmatrix} \in \mathbb{R}^{(n+n^2)\times m} \tag{25b}$$

with the entries given by

$$\frac{\partial\boldsymbol{\mu}_{k+1}}{\partial\sqrt{\boldsymbol{\Sigma}_k}} = \frac{\partial\boldsymbol{\mu}_{k+1}}{\partial\boldsymbol{\Sigma}_k}\frac{\partial\boldsymbol{\Sigma}_k}{\partial\sqrt{\boldsymbol{\Sigma}_k}} \tag{26a}$$

$$\frac{\partial\sqrt{\boldsymbol{\Sigma}_{k+1}}}{\partial\boldsymbol{\mu}_k} = \frac{\partial\sqrt{\boldsymbol{\Sigma}_{k+1}}}{\partial\boldsymbol{\Sigma}_{k+1}}\frac{\partial\boldsymbol{\Sigma}_{k+1}}{\partial\boldsymbol{\mu}_k} \tag{26b}$$

$$\frac{\partial\sqrt{\boldsymbol{\Sigma}_{k+1}}}{\partial\sqrt{\boldsymbol{\Sigma}_k}} = \frac{\partial\sqrt{\boldsymbol{\Sigma}_{k+1}}}{\partial\boldsymbol{\Sigma}_{k+1}}\frac{\partial\boldsymbol{\Sigma}_{k+1}}{\partial\boldsymbol{\Sigma}_k}\frac{\partial\boldsymbol{\Sigma}_k}{\partial\sqrt{\boldsymbol{\Sigma}_k}} \tag{26c}$$

$$\frac{\partial\sqrt{\boldsymbol{\Sigma}_{k+1}}}{\partial\mathbf{u}_k} = \frac{\partial\sqrt{\boldsymbol{\Sigma}_{k+1}}}{\partial\boldsymbol{\Sigma}_{k+1}}\frac{\partial\boldsymbol{\Sigma}_{k+1}}{\partial\mathbf{u}_k} \tag{26d}$$

[1] $\boldsymbol{\Sigma}_k$ is a real symmetric matrix therefore can be diagonalized. The square root of a diagonal matrix can simply be obtained by computing the square roots of diagonal entries.

Since $\dfrac{\partial\sqrt{\boldsymbol{\Sigma}_k}}{\partial\boldsymbol{\Sigma}_k} = \dfrac{1}{2\sqrt{\boldsymbol{\Sigma}_k}}$ and $\dfrac{\partial\sqrt{\boldsymbol{\Sigma}_{k+1}}}{\partial\boldsymbol{\Sigma}_{k+1}} = \dfrac{1}{2\sqrt{\boldsymbol{\Sigma}_{k+1}}}$, they can be expressed as

$$\frac{\partial\boldsymbol{\mu}_{k+1}}{\partial\boldsymbol{\Sigma}_k} = \frac{\partial\boldsymbol{\mu}_{k+1}}{\partial\tilde{\boldsymbol{\mu}}_k}\frac{\partial\tilde{\boldsymbol{\mu}}_k}{\partial\boldsymbol{\Sigma}_k} + \frac{\partial\boldsymbol{\mu}_{k+1}}{\partial\tilde{\boldsymbol{\Sigma}}_k}\frac{\partial\tilde{\boldsymbol{\Sigma}}_k}{\partial\boldsymbol{\Sigma}_k} \tag{27a}$$

$$\frac{\partial\boldsymbol{\Sigma}_{k+1}}{\partial\boldsymbol{\mu}_k} = \frac{\partial\boldsymbol{\Sigma}_{k+1}}{\partial\tilde{\boldsymbol{\mu}}_k}\frac{\partial\tilde{\boldsymbol{\mu}}_k}{\partial\boldsymbol{\mu}_k} + \frac{\partial\boldsymbol{\Sigma}_{k+1}}{\partial\tilde{\boldsymbol{\Sigma}}_k}\frac{\partial\tilde{\boldsymbol{\Sigma}}_k}{\partial\boldsymbol{\mu}_k} \tag{27b}$$

$$\frac{\partial\boldsymbol{\Sigma}_{k+1}}{\partial\boldsymbol{\Sigma}_k} = \frac{\partial\boldsymbol{\Sigma}_{k+1}}{\partial\tilde{\boldsymbol{\mu}}_k}\frac{\partial\tilde{\boldsymbol{\mu}}_k}{\partial\boldsymbol{\Sigma}_k} + \frac{\partial\boldsymbol{\Sigma}_{k+1}}{\partial\tilde{\boldsymbol{\Sigma}}_k}\frac{\partial\tilde{\boldsymbol{\Sigma}}_k}{\partial\boldsymbol{\Sigma}_k} \tag{27c}$$

$$\frac{\partial\boldsymbol{\Sigma}_{k+1}}{\partial\mathbf{u}_k} = \frac{\partial\boldsymbol{\Sigma}_{k+1}}{\partial\tilde{\boldsymbol{\mu}}_k}\frac{\partial\tilde{\boldsymbol{\mu}}_k}{\partial\mathbf{u}_k} + \frac{\partial\boldsymbol{\Sigma}_{k+1}}{\partial\tilde{\boldsymbol{\Sigma}}_k}\frac{\partial\tilde{\boldsymbol{\Sigma}}_k}{\partial\mathbf{u}_k} \tag{27d}$$

$\dfrac{\partial\tilde{\boldsymbol{\mu}}_k}{\partial\boldsymbol{\Sigma}_k}$ and $\dfrac{\partial\tilde{\boldsymbol{\Sigma}}_k}{\partial\boldsymbol{\Sigma}_k}$ can be easily obtained based on (11). Elaborations of $\dfrac{\partial\boldsymbol{\Sigma}_{k+1}}{\partial\tilde{\boldsymbol{\mu}}_k}$ and $\dfrac{\partial\boldsymbol{\Sigma}_{k+1}}{\partial\tilde{\boldsymbol{\Sigma}}_k}$ can be found in [9].

4.2 Problem Reformulation

Based on the local model derived above, define the state variable as

$$\begin{aligned}\mathbf{Z}_{k+1} &= \left[\mathbf{s}_{k+1|k}, \cdots, \mathbf{s}_{k+H|k}\right]^T \in \mathbb{R}^{H(n+n^2)} \\ &= [\boldsymbol{\mu}_{k+1}, \sqrt{\boldsymbol{\Sigma}_{k+1}}, \cdots, \boldsymbol{\mu}_{k+H}, \sqrt{\boldsymbol{\Sigma}_{k+H}}]^T\end{aligned} \tag{28}$$

Also, let

$$\mathbf{U}_k = [\mathbf{u}_k, \cdots, \mathbf{u}_{k+H-1}]^T \in \mathbb{R}^{Hm} \tag{29}$$

$$\mathbf{r}^*_{k+1} = [\mathbf{r}_{k+1}, \mathbf{0}, \cdots, \mathbf{r}_{k+H}, \mathbf{0}]^T \in \mathbb{R}^{H(n+n^2)} \tag{30}$$

Problem (19) then becomes

$$\min_{\mathbf{U}} \left\{ \left\|\mathbf{Z}_{k+1} - \mathbf{r}^*_{k+1}\right\|^2_{\tilde{\mathbf{Q}}} + \|\mathbf{U}_{k+1}\|^2_{\tilde{\mathbf{R}}} \right\} \tag{31a}$$

$$\text{s.t.} \quad \mathbf{I}_{Hn}\mathbf{x}_{\min} \leq \mathbf{M}_z\mathbf{Z}_{k+1} \leq \mathbf{I}_{Hn}\mathbf{x}_{\max} \tag{31b}$$

$$\mathbf{I}_{Hm}\mathbf{u}_{\min} \leq \mathbf{U}_{k+1} \leq \mathbf{I}_{Hm}\mathbf{u}_{\max} \tag{31c}$$

where

$$\begin{aligned}\tilde{\mathbf{Q}} = \operatorname{diag}\{[&\mathbf{Q}, \operatorname{diag}\{\mathbf{vec}(\mathbf{Q})\}, \cdots, \mathbf{Q}, \\ &\operatorname{diag}\{\mathbf{vec}(\mathbf{Q})\}]\} \in \mathbb{R}^{H(n+n^2)\times H(n+n^2)}\end{aligned} \tag{32}$$

$\tilde{\mathbf{R}} = \operatorname{diag}\{[\mathbf{R}, \cdots, \mathbf{R}]\} \in \mathbb{R}^{Hm\times Hm}$, $\mathbf{I}_a \in \mathbb{R}^a$ is the identity vector, and

$$\mathbf{M}_z = \begin{bmatrix} \mathbf{I}_n^T & 2\mathbf{I}_{n^2}^T & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I}_n^T & 2\mathbf{I}_{n^2}^T & \cdots & \mathbf{0} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{I}_n^T & 2\mathbf{I}_{n^2}^T \end{bmatrix} \in \mathbb{R}^{H\times H(n+n^2)} \tag{33}$$

Let $\mathbf{T}_u \in \mathbb{R}^{Hm\times Hm}$ be a lower triangular matrices with unit entries. Then,

$$\mathbf{U}_k = \mathbf{I}_{Hm}\mathbf{u}_{k-1} + \mathbf{T}_u \Delta\mathbf{U}_k. \tag{34}$$

The change in $\mathbf{Z}_{k+1}$ can be expressed as

$$\Delta\mathbf{Z}_{k+1} = \tilde{\mathbf{A}}\Delta\mathbf{s}_k + \tilde{\mathbf{B}}\Delta\mathbf{U}_k \tag{35}$$

based on the local model, with the state and control matrices given by

$$\tilde{\mathbf{A}} = \left[\mathbf{A}, \mathbf{A}^2, \cdots, \mathbf{A}^H\right]^T \in \mathbb{R}^{H(n+n^2)} \tag{36a}$$

$$\tilde{\mathbf{B}} = \begin{bmatrix} \mathbf{B} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{AB} & \mathbf{B} & \cdots & \mathbf{0} \\ \vdots & \vdots & \vdots & \vdots \\ \mathbf{A}^{H-1}\mathbf{B} & \mathbf{A}^{H-2}\mathbf{B} & \cdots & \mathbf{B} \end{bmatrix} \in \mathbb{R}^{H(n+n^2)\times Hm} \tag{36b}$$

where $\mathbf{A}$ and $\mathbf{B}$ are the two Jacobian matrices (25) and (26) respectively. The corresponding state variable $\mathbf{Z}_{k+1}$ is therefore given by

$$\mathbf{Z}_{k+1} = \mathbf{s}_k + \mathbf{T}_z\left(\tilde{\mathbf{A}}\Delta\mathbf{s}_k + \tilde{\mathbf{B}}\Delta\mathbf{U}_k\right) \tag{37}$$

where $\mathbf{T}_z \in \mathbb{R}^{H(n+n^2)\times H(n+n^2)}$ denotes a lower triangular matrix with unit entries.

Based on (34) and (37), problem (31) can be expressed in a more compact form as

$$\min_{\Delta\mathbf{U}} \frac{1}{2}\left\|\Delta\mathbf{U}_k\right\|^2_{\boldsymbol{\Phi}} + \boldsymbol{\psi}^T\Delta\mathbf{U}_k + \mathbf{C} \tag{38a}$$

$$\text{s.t.} \quad \Delta\mathbf{U}_{\min} \leq \begin{bmatrix} \mathbf{T}_u \\ \mathbf{T}_z\tilde{\mathbf{B}} \end{bmatrix} \Delta\mathbf{U}_k \leq \Delta\mathbf{U}_{\max} \tag{38b}$$

where

$$\boldsymbol{\Phi} = \tilde{\mathbf{B}}^T\mathbf{T}_z^T\tilde{\mathbf{Q}}\mathbf{T}_z\tilde{\mathbf{B}} + \mathbf{T}_u^T\tilde{\mathbf{R}}\mathbf{T}_u \in \mathbb{R}^{Hm\times Hm} \tag{39a}$$

$$\boldsymbol{\psi} = 2(\mathbf{s}_k\tilde{\mathbf{Q}}\mathbf{T}_z\tilde{\mathbf{B}} + \Delta\mathbf{s}_k\tilde{\mathbf{A}}^T\tilde{\mathbf{Q}}\tilde{\mathbf{B}} \tag{39b}$$

$$- \mathbf{r}^*_{k+1}\tilde{\mathbf{Q}}\mathbf{T}_z\tilde{\mathbf{B}} + \mathbf{u}_{k-1}\tilde{\mathbf{R}}\mathbf{T}_u) \in \mathbb{R}^{Hm} \tag{39c}$$

$$\mathbf{C} = (\mathbf{s}_k^2 + \mathbf{r}^*_{k+1})\tilde{\mathbf{Q}} + 2\mathbf{s}_k\Delta\mathbf{s}_k\tilde{\mathbf{Q}}\mathbf{T}_z\tilde{\mathbf{A}} \tag{39d}$$

$$+ \mathbf{u}_{k-1}^2\tilde{\mathbf{R}} + \Delta\mathbf{s}_k^2\tilde{\mathbf{A}}^T\tilde{\mathbf{Q}}\tilde{\mathbf{A}} \tag{39e}$$

$$- 2\mathbf{r}^*_{k+1}(\mathbf{s}_k\tilde{\mathbf{Q}} - \Delta\mathbf{s}_k\tilde{\mathbf{Q}}\mathbf{T}_z\tilde{\mathbf{A}}) \tag{39f}$$

$$\Delta\mathbf{U}_{\min} = \begin{bmatrix} \mathbf{I}_{Hm}(\mathbf{u}_{\min} - \mathbf{u}_{k-1}) \\ \mathbf{I}_{H(n+n^2)}(\mathbf{x}_{\min} - \mathbf{s}_k - \mathbf{T}_z\tilde{\mathbf{A}}\Delta\mathbf{s}_k) \end{bmatrix} \tag{39g}$$

$$\Delta\mathbf{U}_{\max} = \begin{bmatrix} \mathbf{I}_{Hm}(\mathbf{u}_{\max} - \mathbf{u}_{k-1}) \\ \mathbf{I}_{H(n+n^2)}(\mathbf{x}_{\max} - \mathbf{s}_k - \mathbf{T}_z\tilde{\mathbf{A}}\Delta\mathbf{s}_k) \end{bmatrix} \tag{39h}$$

Since $\tilde{\mathbf{Q}}, \tilde{\mathbf{R}}, \mathbf{T}_z$ and $\mathbf{T}_u$ are positive definite, $\boldsymbol{\Phi}$ is also positive definite. Hence (38) is a constrained Quadratic Programming (QP) problem and is strictly convex. The solution will therefore be unique and satisfies the Karush-Kahn-Tucker (KKT) conditions.

4.3 Optimization Using Active-Set

The optimization problem (38) can be solved by an active-set method [13]. It iteratively seeks an active (or working) set of constraints and solve an equality constrained QP problem until the optimal solution is found. The advantage of this method is that accurate solutions can still be obtained even when they are ill-conditioned or degenerated. In addition, it is conceptually simple and easy to implement. A warm-start technique could also be used to accelerate the optimization process substantially.

Let $\mathbf{G} = [\mathbf{T}_u, \mathbf{T}_z\tilde{\mathbf{B}}]^T$, the constraint (38b) can be written as

$$\begin{bmatrix} \mathbf{G} \\ -\mathbf{G} \end{bmatrix} \Delta\mathbf{U} \leq \begin{bmatrix} \Delta\mathbf{U}_{\max} \\ -\Delta\mathbf{U}_{\min} \end{bmatrix} \tag{40}$$

Ignoring the constant term $\mathbf{C}$, problem (38) becomes

$$\min_{\Delta\mathbf{U}} \frac{1}{2} \|\Delta\mathbf{U}_k\|^2_{\boldsymbol{\Phi}} + \boldsymbol{\psi}^T \Delta\mathbf{U}_k \tag{41a}$$

$$\text{s.t.} \quad \tilde{\mathbf{G}}\Delta\mathbf{U}_k \leq \tilde{\Delta}_{\mathbf{U}} \tag{41b}$$

where $\tilde{\mathbf{G}} = [\mathbf{G}, -\mathbf{G}]^T \in \mathbb{R}^{2H(m+n+n^2)\times Hm}$ and $\tilde{\Delta}_{\mathbf{U}} = [\Delta\mathbf{U}_{\max}, -\Delta\mathbf{U}_{\min}]^T \in \mathbb{R}^{2H(m+n+n^2)}$.

Let $\boldsymbol{\Pi}_{\Delta\mathbf{U}}$ be the set of feasible points, and $\mathcal{I} = \{1, \cdots, 2H(m+n+n^2)\}$ be the constraint index set. For a feasible point $\Delta\mathbf{U}^*_k \in \boldsymbol{\Pi}_{\Delta\mathbf{U}}$, the index set for the active set of constraints is defined as

$$\mathcal{A}(\Delta\mathbf{U}^*_k) = \{i \subseteq \mathcal{I} | \tilde{\mathbf{G}}_i \Delta\mathbf{U}^*_k = \tilde{\Delta}_{\mathbf{U},i}\} \tag{42}$$

where $\tilde{\mathbf{G}}_i$ is the i^{th} row of $\tilde{\mathbf{G}}$ and $\tilde{\Delta}_{\mathbf{U},i}$ is the i^{th} row of the $\tilde{\Delta}_{\mathbf{U}}$. The inactive set is therefore given by

$$\begin{aligned} \mathcal{B}(\Delta\mathbf{U}^*_k) &= \mathcal{I} \setminus \mathcal{A}(\Delta\mathbf{U}^*_k) \\ &= \{i \subseteq \mathcal{I} | \tilde{\mathbf{G}}_i \Delta\mathbf{U}^*_k < \tilde{\Delta}_{\mathbf{U},i}\} \end{aligned} \tag{43}$$

Given any iteration j, the working set $\mathcal{W}^j_k$ contains all the equality constraints plus the inequality constraints in the active set. The following QP problem subject to the equality constraints w.r.t. $\mathcal{W}^j_k$ is considered given the feasible points $\Delta\mathbf{U}^j_k \in \boldsymbol{\Pi}_{\Delta\mathbf{U}}$:

$$\min_{\boldsymbol{\delta}^j} \frac{1}{2} \left\|\Delta\mathbf{U}^j_k + \boldsymbol{\delta}^j\right\|^2_{\boldsymbol{\Phi}} + \boldsymbol{\psi}^T(\Delta\mathbf{U}^j_k + \boldsymbol{\delta}^j) \tag{44a}$$

$$= \min_{\boldsymbol{\delta}^j} \frac{1}{2} \left\|\boldsymbol{\delta}^j\right\|^2_{\boldsymbol{\Phi}} + (\boldsymbol{\psi} + \boldsymbol{\Phi}\Delta\mathbf{U}^j_k)^T \boldsymbol{\delta}^j \tag{44b}$$

$$+ \underbrace{\frac{1}{2} \left\|\Delta\mathbf{U}^j_k\right\|^2_{\boldsymbol{\Phi}} + \boldsymbol{\psi}^T \Delta\mathbf{U}^j_k}_{\text{constant}}$$

$$\text{s.t.} \quad \tilde{\mathbf{G}}_i(\Delta\mathbf{U}^j_k + \boldsymbol{\delta}^j) = \tilde{\Delta}_{\mathbf{U},i}, i \in \mathcal{W}^j_k \tag{44c}$$

This problem can be simplified by ignoring the constant term to:

$$\min_{\boldsymbol{\delta}^j} \frac{1}{2} \left\| \boldsymbol{\delta}^j \right\|_{\boldsymbol{\Phi}}^2 + (\boldsymbol{\psi} + \boldsymbol{\Phi} \Delta \mathbf{U}_k^j)^T \boldsymbol{\delta}^j \tag{45a}$$

$$= \min_{\boldsymbol{\delta}^j} \frac{1}{2} \boldsymbol{\delta}^{j^T} \boldsymbol{\Phi} \boldsymbol{\delta}^j + (\boldsymbol{\psi} + \boldsymbol{\Phi} \Delta \mathbf{U}_k^j)^T \boldsymbol{\delta}^j \tag{45b}$$

$$\text{s.t.} \quad \tilde{\mathbf{G}}_i \boldsymbol{\delta}^j = \tilde{\Delta}_{\mathbf{U},i} - \tilde{\mathbf{G}}_i \Delta \mathbf{U}_k^j, i \in \mathcal{W}_k^j \tag{45c}$$

By applying the KKT conditions to problem (45), we can obtain the following linear equations:

$$\underbrace{\begin{bmatrix} \boldsymbol{\Phi} & \tilde{\mathbf{G}}_{\mathcal{A}}^T \\ \tilde{\mathbf{G}}_{\mathcal{A}} & \mathbf{0} \end{bmatrix}}_{\text{Lagrangian Matrix}} \begin{bmatrix} \boldsymbol{\delta}^j \\ \boldsymbol{\lambda}_k^* \end{bmatrix} = \begin{bmatrix} -\boldsymbol{\psi} - \boldsymbol{\Phi} \Delta \mathbf{U}_k^j \\ \tilde{\Delta}_{\mathbf{U},\mathcal{A}} - \tilde{\mathbf{G}}_{\mathcal{A}} \Delta \mathbf{U}_k^j \end{bmatrix} \tag{46}$$

where $\boldsymbol{\lambda}_k^* \in \mathbb{R}^{2H(m+n+n^2)}$ denotes the vector of Lagrangian multipliers, $\tilde{\mathbf{G}}_{\mathcal{A}} \subseteq \tilde{\mathbf{G}}$ and $\tilde{\Delta}_{\mathbf{U},\mathcal{A}} \subset \tilde{\Delta}_{\mathbf{U}}$ are the weighting matrix and the upper bounds of the constraints w.r.t. $\mathcal{W}_k^j$. Let the inverse of Lagrangian matrix be denoted by

$$\begin{bmatrix} \boldsymbol{\Phi} & \tilde{\mathbf{G}}_{\mathcal{A}}^T \\ \tilde{\mathbf{G}}_{\mathcal{A}} & \mathbf{0} \end{bmatrix}^{-1} = \begin{bmatrix} \mathbf{L}_1 & \mathbf{L}_2^T \\ \mathbf{L}_2 & \mathbf{L}_3 \end{bmatrix} \tag{47}$$

If this inverse exists, then the solution is given by

$$\boldsymbol{\delta}^j = -\mathbf{L}_1(\boldsymbol{\psi} + \boldsymbol{\Phi} \Delta \mathbf{U}_k^j) + \mathbf{L}_2^T (\tilde{\Delta}_{\mathbf{U},\mathcal{A}} - \tilde{\mathbf{G}}_{\mathcal{A}} \Delta \mathbf{U}_k^j) \tag{48a}$$

$$\boldsymbol{\lambda}_k^* = -\mathbf{L}_2(\boldsymbol{\psi} + \boldsymbol{\Phi} \Delta \mathbf{U}_k^j) + \mathbf{L}_3 (\tilde{\Delta}_{\mathbf{U},\mathcal{A}} - \tilde{\mathbf{G}}_{\mathcal{A}} \Delta \mathbf{U}_k^j) \tag{48b}$$

where

$$\mathbf{L}_1 = \boldsymbol{\Phi}^{-1} - \boldsymbol{\Phi}^{-1} \tilde{\mathbf{G}}_{\mathcal{A}}^T (\tilde{\mathbf{G}}_{\mathcal{A}} \boldsymbol{\Phi}^{-1} \tilde{\mathbf{G}}_{\mathcal{A}}^T)^{-1} \tilde{\mathbf{G}}_{\mathcal{A}} \boldsymbol{\Phi}^{-1} \tag{49a}$$

$$\mathbf{L}_2 = \boldsymbol{\Phi}^{-1} \tilde{\mathbf{G}}_{\mathcal{A}}^T (\tilde{\mathbf{G}}_{\mathcal{A}} \boldsymbol{\Phi}^{-1} \tag{49b}$$

$$\mathbf{L}_3 = -(\tilde{\mathbf{G}}_{\mathcal{A}} \boldsymbol{\Phi}^{-1} \tilde{\mathbf{G}}_{\mathcal{A}}^T)^{-1} \tag{49c}$$

If $\boldsymbol{\delta}^j \neq 0$, then the set of feasible points $\Delta \mathbf{U}_k^j$ fails to minimize problem (41). In this case, the next set of feasible point is computed for the next iteration by $\Delta \mathbf{U}_k^{j+1} = \Delta \mathbf{U}_k^j + \kappa^j \boldsymbol{\delta}^j$ with step size

$$\kappa^j = \min \left\{ 1, \min_{i \in \mathcal{B}(\Delta \mathbf{U}_k^j)} \frac{\tilde{\Delta}_{\mathbf{U},i} - \tilde{\mathbf{G}}_i \Delta \mathbf{U}_k^j}{\tilde{\mathbf{G}}_i \boldsymbol{\delta}^j} \right\} \tag{50}$$

If $\kappa^j < 1$, the inequality constraint with index $q = \underset{i \in \mathcal{B}(\Delta \mathbf{U}_k^j)}{\operatorname{argmin}} \frac{\tilde{\Delta}_{\mathbf{U},i} - \tilde{\mathbf{G}}_i \Delta \mathbf{U}_k^j}{\tilde{\mathbf{G}}_i \boldsymbol{\delta}^j}$ should be “activated”, giving the working set $\mathcal{W}_k^{j+1} = \mathcal{W}_k^j \cup q$. Otherwise, we have $\mathcal{W}_k^{j+1} = \mathcal{W}_k^j$.

Alternatively, if the solution gives $\boldsymbol{\delta}^j = 0$, then the current feasible points $\Delta \mathbf{U}_k^j$ could be the optimal solution. This can be verified by checking the Lagrangian multiplier $\lambda_k^* = \min_{i \in \mathcal{W}_k^j \cap \mathcal{I}} \boldsymbol{\lambda}_{k,i}^*$. If $\lambda_k^* \geq 0$, the optimal solution of the (41) at sampling

```
Initialization
    the feasible point $\Delta\mathbf{U}_k^0 \in \boldsymbol{\Pi}_{\Delta\mathbf{U}}$;
    the working set $\mathcal{W}^0 = \mathcal{A}(\Delta\mathbf{U}_k^0)$;
for $j = 0, 1, 2, \cdots$ do
    Compute the $\boldsymbol{\delta}^j$ and $\boldsymbol{\lambda}_k^*$ by solving the linear equations (46);
    if $\boldsymbol{\delta}^j = 0$ then
        $\lambda_k^* = \min_{i \in \mathcal{W}_k^j \cap \mathcal{I}} \boldsymbol{\lambda}_{k,i}^*$,
        $p = \operatorname{argmin}_{i \in \mathcal{W}_k^j \cap \mathcal{I}} \boldsymbol{\lambda}_{k,i}^*$
        if $\lambda_k^* \geq 0$ then
            $\Delta\mathbf{U}_k^* = \Delta\mathbf{U}_k^j$;
            Stop.
        else
            $\mathcal{W}_k^{j+1} = \mathcal{W}_k^j \setminus p$;
            $\Delta\mathbf{U}_k^{j+1} = \Delta\mathbf{U}_k^j$;
        end
    else
        Compute the step length $\kappa^j$ by (50),
        $q = \operatorname{argmin}_{i \in \mathcal{B}(\Delta\mathbf{U}_k^j)} \frac{\tilde{\Delta}_{\mathbf{U},i} - \tilde{\mathbf{G}}_i \Delta\mathbf{U}_k^j}{\tilde{\mathbf{G}}_i \boldsymbol{\delta}^j}$
        if $\kappa^j < 1$ then
            $\Delta\mathbf{U}_k^{j+1} = \Delta\mathbf{U}_k^j + \kappa^j \boldsymbol{\delta}^j$;
            $\mathcal{A}(\Delta\mathbf{U}_k^{j+1}) = \mathcal{A}(\Delta\mathbf{U}_k^j) \cup q$;
        else
            $\Delta\mathbf{U}_k^{j+1} = \Delta\mathbf{U}_k^j + \boldsymbol{\delta}^j$;
            $\mathcal{A}(\Delta\mathbf{U}_k^{j+1}) = \mathcal{A}(\Delta\mathbf{U}_k^j)$;
        end
    end
end
```

Algorithm 1: Active-set method for solving the resulting convex optimization problem

time k is found. Otherwise, this inequality constraint indexed by $p = \operatorname{argmin}_{i \in \mathcal{W}_k^j \cap \mathcal{I}} \boldsymbol{\lambda}_{k,i}^*$ should be removed from the current working set, giving us $\mathcal{W}_k^{j+1} = \mathcal{W}_k^j \setminus p$. Algorithm 1 summarizes the active-set algorithm used in the GPMPC.

4.4 Implementation Issues

The key to solving equation (46) is the inverse of the Lagrangian matrix. However, $\tilde{\mathbf{G}}_{\mathcal{A}}$ is not always full ranked. Thus the Lagrangian matrix is not always invertible. This problem can be solved by decomposing $\tilde{\mathbf{G}}_{\mathcal{A}}$ using QR factorization, giving us $\mathbf{G}_{\mathcal{A}}^T = \mathcal{Q}\,[\mathcal{R}\ \ \mathbf{0}]^T$ where $\mathcal{R} \in \mathbb{R}^{m_1 \times m_1}$ is an upper triangular matrix with $m_1 = \operatorname{rank}(\tilde{\mathbf{G}}_{\mathcal{A}})$. $\mathcal{Q} \in \mathbb{R}^{Hm \times Hm}$ is an orthogonal matrix that can be further decomposed to $\mathcal{Q} = [\mathcal{Q}_1\ \mathcal{Q}_2]$ where $\mathcal{Q}_1 \in \mathbb{R}^{Hm \times m_1}$ and $\mathcal{Q}_2 \in \mathbb{R}^{Hm \times (Hm - m_1)}$. Thus, $\mathbf{G}_{\mathcal{A}}^T = \mathcal{Q} = \mathcal{Q}_1\mathcal{R}$ and

$$\mathbf{L}_1 = \mathcal{Q}_2(\mathcal{Q}_2^T \boldsymbol{\Phi} \mathcal{Q}_2)^{-1} \mathcal{Q}_2^T \tag{51a}$$

$$\mathbf{L}_2 = \mathcal{Q}_1 {\mathcal{R}^{-1}}^T - \mathbf{L}_1 \boldsymbol{\Phi} \mathcal{Q}_1 {\mathcal{R}^{-1}}^T \tag{51b}$$

$$\mathbf{L}_3 = \mathcal{R}^{-1} \mathcal{Q}_1^T \boldsymbol{\Phi} \mathbf{L}_2 \tag{51c}$$

The second issue relates to using the appropriate warm-start technique to improve the convergence rate of the active-set method. For GPMPC, since the changes in the state between two successive sampling instants are usually quite small, we can simply use the previous value $\Delta\mathbf{U}_k^*$ at sampling time k as the starting point $\Delta\mathbf{U}_{k+1}^0$ for the next sampling time $k+1$. This warm-start technique is usually employed in MPC optimizations because of its proven effectiveness [42].

5 Simulation Results

The performance of GPMPC for quadrotor trajectory tracking is evaluated by computer simulations. The parameters of translational and rotational subsystems in the numerical quadrotor system are the same as those used in [2]. All simulations are independently repeated 50 times on a computer with a 3.40GHz Intel® Core$^{\mathrm{TM}}$ 2 Duo CPU with 16 GB RAM, using Matlab® version 8.1. The simulation results presented below are the average values from 50 independent trials.

Two non-trivial trajectories are used. They are referred to as “Elliptical” and “Lorenz” trajectories and are shown as red dotted lines in Figure 4(a) and 4(b)) respectively. The quadrotor subsystems are subject to external Gaussian white noise with zero mean and unit variance. The constraints on the control inputs of the translational subsystem are $0 \leq U_1(k) \leq 100, -0.2 \leq u_x(k) \leq 0.2, -0.2 \leq u_y(k) \leq 0.2$ for the “Elliptical” trajectory, and they are $-45 \leq u_1(k) \leq 0, -2 \leq u_x(k) \leq 2, -2 \leq u_y(k) \leq 2$ for the “Lorenz” trajectory. For the rotational subsystem, all observations are scaled to the range $[0.1, 0.9]$. The inputs are scaled accordingly. This is necessary because the numerical range of the original data is very large. For example, the unscaled angle ϕ lies in the range $[-1.57, 1.57]$ while input U_4 lies in the range $[-3.2, 6.2] \times 10^{-8}$. Using the scaled data leads to much improved training results.

To generate observations for GP modelling, the trajectory tracking tasks are first performed by using the Nonlinear Model Predictive Control (NMPC) strategy proposed in [17] but without input constraints. For each trajectory, 189 observations which consist of inputs, states and outputs are collected for use in GP model training. The initial state and initial control input are zero. The weighting matrices $\mathbf{Q}$ and $\mathbf{R}$ are identity matrices. Sampling frequency f_s is 1 Hz.

5.1 Modelling Results

The first set of results show how well the GP models are trained with different sizes of training data. The full set of 189 data are used for testing. As given in Table 1 and 2, the obtained GP models capture the training data well as the training MSE values are small. The prediction accuracies reflected by the test MSE values show a sudden drop when sufficiently large training sizes are used. The computational time required for training averages from approximately 1.12 seconds for a size of 10 to 4.01 seconds for a size of 189.

Table 1: Prediction training and test MSE values of the GP models in the "Elliptical" tracking problems. ("Trans" denotes the translational subsystem and "Rotate" represents the rotational subsystem.)

	Size	Training	Test	Average Var
"Trans"	10	2.2485e-4	1.4806	**1.0231**
	50	4.1787e-6	1.2531	**0.1074**
	100	3.0511e-7	1.6733e-6	**0.0057**
	189	1.0132e-7	1.0132e-7	**2.4843e-4**
"Rotate"	10	2.7443e-6	2.3232e-4	**2.1150e-4**
	50	3.0020e-8	1.0502e-6	**1.0853e-4**
	100	2.8578e-9	7.5105e-8	**1.0620e-4**
	189	1.0457e-9	1.0457e-9	**1.0590e-4**

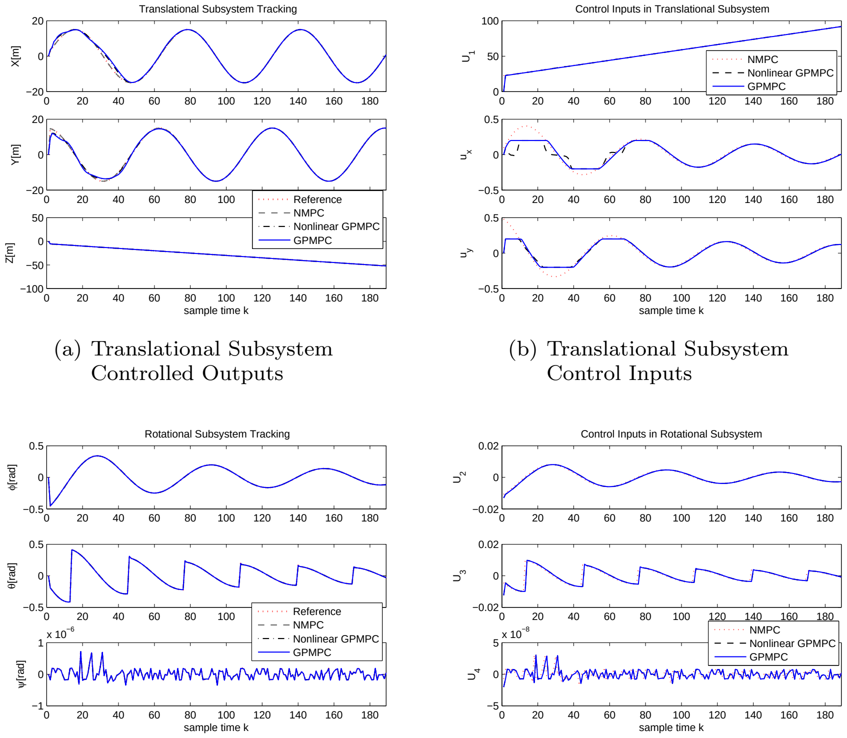
Table 2: Prediction training and test MSE values of the GP models in the "Lorenz" tracking problems. ("Trans" denotes the translational subsystem and "Rotat" represents the rotational subsystem.)

	Size	Training	Test	Average Var
"Trans"	10	4.0309e-4	2.6872	**4.7156**
	50	1.1986e-4	1.1820	**1.1696**
	100	6.5945e-6	0.0122	**0.0105**
	189	3.0415e-6	3.0415e-6	**1.0870e-4**
"Rotate"	10	1.0511e-5	0.0044	**3.1641e-4**
	50	9.4195e-7	4.2896e-5	**1.0686e-4**
	100	3.9616e-8	2.7571e-6	**1.0607e-4**
	189	9.2117e-9	9.2117e-9	**1.0566e-4**

5.2 Control Results

The GP models used in the control tasks are trained with all 189 observations because this ensures that the best quality models are obtained. The performance of using proposed GPMPC scheme is compared with using an exiting GP based MPC algorithm (referred to as "nonlinear GPMPC" or NMPC below) proposed in [23]. Even though our optimization problem (19) with cost function (22) is more complicated than the one considered in [23], they are essentially similar. In addition, we choose $H = 1$ as the prediction horizon. Theoretically, larger values of H is necessary to guarantee the stability of MPC controllers. However, since solving the nonlinear GPMPC problem with larger values of H effectively is an open problem, we restrict H to be 1 in order to make proper comparisons. Our previous work in [7] has demonstrated that the proposed GPMPC can efficiently be used with a longer horizon.

The control results for the two trajectories are shown in Figures 2 and 3. They show that NMPC has the best tracking control performance. However, it should be noted that NMPC does not place any constraints on the control inputs. In general, the proposed GPMPC is able to closely follow the desired position and attitude values with constrained control inputs. The overall trajectory tracking



(a) Translational Subsystem Controlled Outputs

(b) Translational Subsystem Control Inputs

(c) Rotational Subsystem Controlled Outputs

(d) Rotational Subsystem Control Inputs

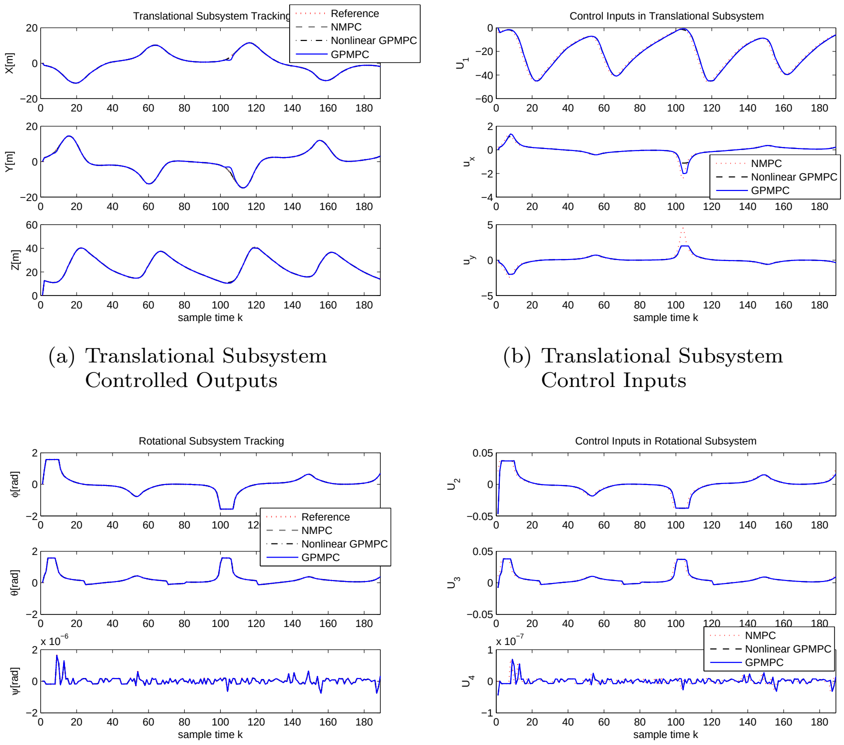
Fig. 2: Controlled outputs and control inputs by using the proposed GPMPC for the both two subsystems in the “Elliptical” trajectory

results through using the GPMPC based hierarchical control scheme are depicted graphically in Figures 4(a) and 4(b).

Even with $H = 1$, the nonlinear GPMPC requires 220 seconds and 272 seconds to compute all 189 control inputs for the “Elliptical” and “Lorenz” trajectory respectively. This is in contrast to the proposed GPMPC algorithm which only takes 60 seconds and 56 seconds. This demonstrates that the proposed GPMPC is computationally much more efficient than nonlinear GPMPC.

6 Conclusions

A new MPC algorithm is proposed for the quadrotor trajectory tracking problem where the quadrotor models are trained from empirical data using GP techniques. hierarchical control scheme based on a computationally efficient GP based for the quadrotor trajectory tracking problem. Models of the translational and rotational subsystems are learnt from collected data using GP modelling techniques rather than by traditional Newtonian analysis. The proposed GPMPC is able to computationally solve the resulting MPC tracking problems which are originally



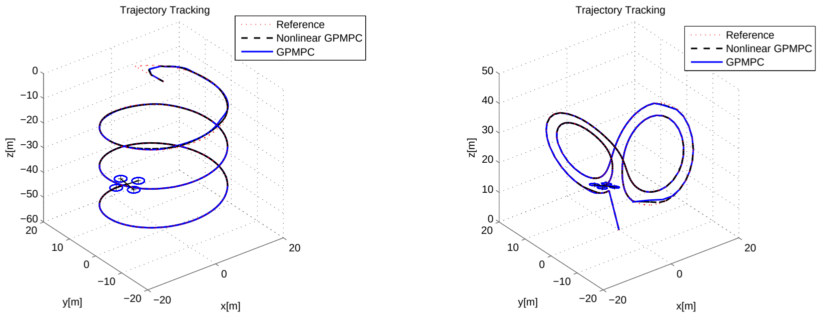
(a) Translational Subsystem Controlled Outputs

(b) Translational Subsystem Control Inputs

(c) Rotational Subsystem Controlled Outputs

(d) Rotational Subsystem Control Inputs

Fig. 3: Controlled outputs and control inputs by using the proposed GPMPC for the both two subsystems in the "Lorenz" trajectory



(a) "Elliptical"

(b) "Lorenz"

Fig. 4: The results of tracking "Elliptical" and "Lorenz" trajectories using the proposed control scheme

non-convex but can be reformulated as convex ones by using a linearisation of GP models. The numerical simulation results show that the proposed control scheme is able to closely track non-trivial trajectories. Its tracking performance is similar to using an NMPC method even though GPMPC has input constraints while no input constraints are placed on NMPC. In addition, compared to using an existing nonlinear GPMPC, the proposed GPMPC based control scheme has the advantage of solving the MPC problem much efficiently.

Appendix A

First rewrite (22) as follows:

$$\begin{aligned}
&E[\mathcal{J}(\mathbf{x}_k, \mathbf{u}_{k-1})] \\
&= E\Big[\sum_{i=1}^{H} \Big\{\big\|\mathbf{x}_{k+i} - \mathbf{r}_{k+i}\big\|_{\mathbf{Q}}^2 + \big\|\mathbf{u}_{k+i-1}\big\|_{\mathbf{R}}^2\Big\}\Big] \\
&= \sum_{i=1}^{H} \Bigg\{ \underbrace{E\Big[(\mathbf{x}_{k+i} - \mathbf{r}_{k+i})^T \mathbf{Q} (\mathbf{x}_{k+i} - \mathbf{r}_{k+i})\Big]}_{\text{probabilistic term}} \\
&\qquad + \underbrace{\mathbf{u}_{k+i-1}^T \mathbf{R} \mathbf{u}_{k+i-1}}_{\text{determinisitc term}} \Bigg\}
\end{aligned} \tag{52}$$

Let Q_{ab} be the entries of $\mathbf{Q}$ thus $Q_{ab} = [\mathbf{Q}]_{ab}$ and ε_{ab} as the entries of $\boldsymbol{\Sigma}_k$ thus $\varepsilon_{ab} = [\boldsymbol{\Sigma}_k]_{ab}$, the “probabilistic term” can be further derived to

$$\begin{aligned}
&E\Big[(\mathbf{x}_{k+i} - \mathbf{r}_{k+i})^T \mathbf{Q} (\mathbf{x}_{k+i} - \mathbf{r}_{k+i})\Big] \\
&= E\Big[\sum_{a=1}^{N}\sum_{b=1}^{N} Q_{ab} (\mathbf{x}_{k+i,a} - \mathbf{r}_{k+i,a})(\mathbf{x}_{k+i,b} - \mathbf{r}_{k+i,b})\Big] \\
&= \sum_{a=1}^{N}\sum_{b=1}^{N} Q_{ab} E\Big[(\mathbf{x}_{k+i,a} - \mathbf{r}_{k+i,a})(\mathbf{x}_{k+i,b} - \mathbf{r}_{k+i,b})\Big] \\
&= \sum_{a=1}^{N}\sum_{b=1}^{N} Q_{ab} \Big\{ E\big[\mathbf{x}_{k+i,a} - \mathbf{r}_{k+i,a}\big] E\big[\mathbf{x}_{k+i,b} - \mathbf{r}_{k+i,b}\big] \\
&\quad \underbrace{+ \operatorname{Cov}\Big((\mathbf{x}_{k+i,a} - \mathbf{r}_{k+i,a}), (\mathbf{x}_{k+i,b} - \mathbf{r}_{k+i,b})\Big)}_{\varepsilon_{ab}} \Big\} \\
&= \sum_{a=1}^{N}\sum_{b=1}^{N} Q_{ab} \Big\{ (\boldsymbol{\mu}_{k+i,a} - \mathbf{r}_{k+i,a})(\boldsymbol{\mu}_{k+i,a} - \mathbf{r}_{k+i,a}) + \varepsilon_{ab} \Big\}
\end{aligned} \tag{53}$$

where

$$\begin{aligned}
&\sum_{a=1}^{N}\sum_{b=1}^{N} Q_{ab} (\boldsymbol{\mu}_{k+i,a} - \mathbf{r}_{k+i,a})(\boldsymbol{\mu}_{k+i,a} - \mathbf{r}_{k+i,a}) \\
&= (\boldsymbol{\mu}_{k+i} - \mathbf{r}_{k+i})^T \mathbf{Q} (\boldsymbol{\mu}_{k+i} - \mathbf{r}_{k+i})
\end{aligned} \tag{54}$$

and

$$\sum_{a=1}^{N}\sum_{b=1}^{N} Q_{ab}\varepsilon_{ab} = \mathit{trace}(\mathbf{Q}\boldsymbol{\Sigma}_{k+i}) \tag{55}$$

Therefore, (52) can be obtained by

$$\begin{aligned} & E[\mathcal{J}(\mathbf{x}_k, \mathbf{u}_{k-1})] \\ & = \sum_{i=1}^{H} \Big\{ (\boldsymbol{\mu}_{k+i} - \mathbf{r}_{k+i})^T \mathbf{Q} (\boldsymbol{\mu}_{k+i} - \mathbf{r}_{k+i}) + \mathbf{u}_{k+i-1}^T \mathbf{R} \mathbf{u}_{k+i-1} \\ & \quad + \mathit{trace}(\mathbf{Q}\boldsymbol{\Sigma}_{k+i}) \Big\} \\ & = \sum_{i=1}^{H} \Big\{ \left\| \boldsymbol{\mu}_{k+i} - \mathbf{r}_{k+i} \right\|_{\mathbf{Q}}^2 + \left\| \mathbf{u}_{k+i-1} \right\|_{\mathbf{R}}^2 + \mathit{trace}\big(\mathbf{Q}\boldsymbol{\Sigma}_{k+i}\big) \Big\} \end{aligned} \tag{56}$$

References

1. Abdolhosseini, M., Zhang, Y., Rabbath, C.A.: An efficient model predictive control scheme for an unmanned quadrotor helicopter. Journal of Intelligent & Robotic Systems **70**(1-4), 27–38 (2013)
2. Alexis, K., Nikolakopoulos, G., Tzes, A.: Switching model predictive attitude control for a quadrotor helicopter subject to atmospheric disturbances. Control Engineering Practice **19**(10), 1195–1207 (2011)
3. Berkenkamp, F., Schoellig, A.P.: Learning-based robust control: Guaranteeing stability while improving performance. In: IEEE/RSJ Proceedings of International Conference on Intelligent Robots and Systems (IROS) (2014)
4. Bouabdallah, S., Noth, A., Siegwart, R.: PID vs LQ control techniques applied to an indoor micro quadrotor. In: IEEE/RSJ Proceedings of International Conference on Intelligent Robots and Systems (IROS), vol. 3, pp. 2451–2456. IEEE (2004)
5. Candela, J.Q., Girard, A., Larsen, J., Rasmussen, C.E.: Propagation of uncertainty in bayesian kernel models-application to multiple-step ahead forecasting. In: IEEE Proceedings of International Conference on Acoustics, Speech, and Signal Processing (ICASSP), vol. 2, pp. II–701. IEEE (2003)
6. Cao, G., Lai, E.M.K., Alam, F.: Gaussian process based model predictive control for linear time varying systems. In: International Workshop on Advanced Motion Control (AMC Workshop). IEEE (2016)
7. Cao, G., Lai, E.M.K., Alam, F.: Gaussian process model predictive control of unknown nonlinear systems. IET Control Theory & Applications (2016). URL `https://arxiv.org/abs/1612.01211`. Accepted for publication
8. Cao, G., Lai, E.M.K., Alam, F.: Gaussian process model predictive control of Unmanned Quadrotors. In: International Conference on Control, Automation and Robotics (ICCAR). IEEE (2016)
9. Deisenroth, M.P.: Efficient reinforcement learning using Gaussian processes. Ph.D. thesis, Karlsruhe Institute of Technology (2010)
10. Diehl, M., Ferreau, H.J., Haverbeke, N.: Efficient numerical methods for nonlinear MPC and moving horizon estimation. In: International Workshop on assessment and future directions on Nonlinear Model Predictive Control, pp. 391–417. Springer, Pavia, Italy (2008)
11. Dierks, T., Jagannathan, S.: Output feedback control of a quadrotor UAV using neural networks. IEEE Transactions on Neural Networks **21**(1), 50–66 (2010)
12. Doherty, P., Rudol, P.: A UAV search and rescue scenario with human body detection and geolocalization. In: Advances in Artificial Intelligence, pp. 1–13. Springer (2007)
13. Fletcher, R.: Practical methods of optimization, second edn. Wiley-Interscience Publication (1987)

14. Girard, A., Rasmussen, C.E., Candela, J.Q., Murray-Smith, R.: Gaussian process priors with uncertain input – Application to multiple-step ahead time series forecasting. In: Advances in Neural Information Processing Systems (NIPS), pp. 545–552. MIT (2003)
15. Grancharova, A., Johansen, T.A., Tøndel, P.: Computational aspects of approximate explicit nonlinear model predictive control. In: Proceedings of the International Workshop on Assessment and Future Directions of Nonlinear Model Predictive Control, pp. 181–192. Springer (2007)
16. Grancharova, A., Kocijan, J., Johansen, T.A.: Explicit stochastic predictive control of combustion plants based on Gaussian process models. Automatica **44**(6), 1621–1631 (2008)
17. Grüne, L., Pannek, J.: Nonlinear model predictive control–Theory and Algorithms. Springer-Verlag, London, U.K (2011)
18. Han, F., Feng, G., Wang, Y., Zhou, F.: Fuzzy modeling and control for a nonlinear quadrotor under network environment. In: IEEE 4th Annual International Conference on Cyber Technology in Automation Control, and Intelligent Systems (CYBER), pp. 395–400. IEEE (2014)
19. Hemakumara, P., Sukkarieh, S.: Non-parametric UAV system identification with dependent Gaussian processes. In: IEEE Proceedings of International Conference on Robotics and Automation (ICRA), pp. 4435–4441. IEEE (2011)
20. Hemakumara, P., Sukkarieh, S.: UAV parameter estimation with multi-output local and global Gaussian process approximations. In: IEEE Proceedings of International Conference on Robotics and Automation (ICRA), pp. 5402–5408. IEEE (2013)
21. Huang, M., Xian, B., Diao, C., Yang, K., Feng, Y.: Adaptive tracking control of underactuated quadrotor unmanned aerial vehicles via backstepping. In: American Control Conference, pp. 2076–2081. IEEE (2010)
22. Klenske, E.D., Zeilinger, M.N., Scholkopf, B., Hennig, P.: Gaussian process-based predictive control for periodic error correction. IEEE Transactions on Control Systems Technology (2015)
23. Kocijan, J., Murray-Smith, R.: Nonlinear predictive control with a Gaussian process model. In: In R. Murray-Smith and R. Shorten (eds.), Switching and Learning in Feedback Systems, pp. 185–200. Springer, Heidelberger, Berlin, Germany (2005)
24. Kocijan, J., Murray-Smith, R., Rasmussen, C.E., Girard, A.: Gaussian process model based predictive control. In: American Control Conference, vol. 3, pp. 2214–2219. IEEE (2004)
25. Kocijan, J., Murray-Smith, R., Rasmussen, C.E., Likar, B.: Predictive control with Gaussian process models. In: Proceedings of IEEE Region 8 EUROCON 2003:Computer As A Tool, vol. A, pp. 352–356. IEEE, Ljubljana (2003)
26. Likar, B., Kocijan, J.: Predictive control of a gas–liquid separation plant based on a Gaussian process model. Computers & Chemical Engineering **31**(3), 142–152 (2007)
27. Liuzzi, G., Lucidi, S., Sciandrone, M.: Sequential penalty derivative-free methods for nonlinear constrained optimization. SIAM Journal on Optimization **20**(5), 2614–2635 (2010)
28. Lucidi, S., Sciandrone, M., Tseng, P.: Objective-derivative-free methods for constrained optimization. Mathematical Programming **92**(1), 37–59 (2002)
29. Madani, T., Benallegue, A.: Sliding mode observer and backstepping control for a quadrotor unmanned aerial vehicles. In: American Control Conference, pp. 5887–5892. IEEE (2007)
30. Mesbah, A.: Stochastic model predictive control: An overview and perspectives for future research. IEEE Control Systems Magazine, Accepted (2016)
31. Metni, N., Hamel, T.: A UAV for bridge inspection: Visual servoing control law with orientation limits. Automation in construction **17**(1), 3–10 (2007)
32. Pan, Y., Theodorou, E.: Probabilistic differential dynamic programming. In: Advances in Neural Information Processing Systems (NIPS), pp. 1907–1915 (2014)
33. Quiñonero-Candela, J., Rasmussen, C.E.: A unifying view of sparse approximate Gaussian process regression. Journal of Machine Learning Research **6**, 1939–1959 (2005)
34. Raffo, G.V., Ortega, M.G., Rubio, F.R.: An integral predictive/nonlinear H∞ control structure for a quadrotor helicopter. Automatica **46**(1), 29–39 (2010)
35. Rasmussen, C., Williams, C.: Gaussian Processes for Machine Learning. MIT Press, Cambridge, MA, USA (2006)
36. Valipour, M.: Application of new mass transfer formulae for computation of evapotranspiration. Journal of Applied Water Engineering and Research **2**(1), 33–46 (2014)
37. Valipour, M.: How much meteorological information is necessary to achieve reliable accuracy for rainfall estimations ?. Agriculture **6**(4), 53 (2016)

38. Valipour, M.: Optimization of neural networks for precipitation analysis in a humid region to detect drought and wet year alarms. Meteorological Applications **23**(1), 91–100 (2016)
39. Valipour, M.: Variations of land use and irrigation for next decades under different scenarios. IRRIGA:Brazilian Journal of Irrigation and Drainage **1**(01), 262–288 (2016)
40. Valipour, M., Sefidkouhi, M.A.G., Raeini, M., et al.: Selecting the best model to estimate potential evapotranspiration with respect to climate change and magnitudes of extreme events. Agricultural Water Management **180**, 50–60 (2017)
41. Voos, H.: Nonlinear and neural network-based control of a small four-rotor aerial robot. In: 2007 IEEE/ASME international conference on Advanced intelligent mechatronics, pp. 1–6. IEEE (2007)
42. Wang, Y., Boyd, S.: Fast model predictive control using online optimization. IEEE Transactions on Control Systems Technology **18**(2), 267–278 (2010)
43. Yannopoulos, S.I., Lyberatos, G., Theodossiou, N., Li, W., Valipour, M., Tamburrino, A., Angelakis, A.N.: Evolution of water lifting devices (pumps) over the centuries worldwide. Water **7**(9), 5031–5060 (2015)
44. Yiqing, L., Xigang, Y., Yongjian, L.: An improved PSO algorithm for solving non-convex NLP/MINLP problems with equality constraints. Computers & chemical engineering **31**(3), 153–162 (2007)
45. Zuo, Z.: Trajectory tracking control design with command-filtered compensation for a quadrotor. IET Control Theory & Applications **4**(11), 2343–2355 (2010)