Trajectory Optimisation in Learned Multimodal Dynamical Systems via Latent-ODE Collocation

Aidan Scannell
University of Bristol
aidan.scannell@bristol.ac.uk

Carl Henrik Ek
University of Cambridge
che29@cam.ac.uk

Arthur Richards
University of Bristol
arthur.richards@bristol.ac.uk

***Abstract*—This paper presents a two-stage method to perform trajectory optimisation in multimodal dynamical systems with unknown nonlinear stochastic transition dynamics. The method finds trajectories that remain in a preferred dynamics mode where possible and in regions of the transition dynamics model that have been observed and can be predicted confidently. The first stage leverages a Mixture of Gaussian Process Experts method to learn a predictive dynamics model from historical data. Importantly, this model learns a gating function that indicates the probability of being in a particular dynamics mode at a given state location. This gating function acts as a coordinate map for a latent Riemannian manifold on which shortest trajectories are solutions to our trajectory optimisation problem. Based on this intuition, the second stage formulates a geometric cost function, which it then implicitly minimises by projecting the trajectory optimisation onto the second-order geodesic ODE; a classic result of Riemannian geometry. A set of collocation constraints are derived that ensure trajectories are solutions to this ODE, implicitly solving the trajectory optimisation problem.**

I. Introduction

Many physical systems operate under switching dynamics modes due to changing environmental or internal conditions. Examples include: robotic grasping where objects with different properties have to be manipulated, robotic locomotion in environments with varying surface types and the control of aircraft in environments subject to different levels of turbulence. When controlling these systems, it may be preferred to find trajectories that remain in a single dynamics mode. This paper is interested in controlling a DJI Tello quadcopter in an environment with spatially varying turbulence induced by a fan at the side of the room, shown in Fig. 1. It is hard to know the exact transition dynamics due to complex and uncertain interactions between the quadcopter and the fan. The system's transition dynamics resemble a mixture of two modes: a turbulent mode in front of the fan and a non-turbulent mode everywhere else. When planning a trajectory from start state $\mathbf{x}_0$ to desired state $\mathbf{x}_f$ it is preferred to avoid entering the turbulent mode, as it results in poor performance and sometimes even system failure.

Trajectory optimisation comprises a powerful set of techniques for finding open-loop controls of dynamical systems such that an objective function is minimised whilst satisfying a set of constraints. It is commonly used for controlling aircraft, robotic manipulators, and walking robots [1, 7, 25]. One caveat to trajectory optimisation is that it requires a relatively accurate mathematical model of the system. Traditionally, these mathematical models are built using first principles based on physics. However, accurately modelling the underlying transition dynamics can be challenging and lead to the introduction of model errors. For example, both observation and process noise are inherent in many real-world systems and can be hard to model due to both spatial and temporal variations. Incorrectly accounting for this uncertainty can have a detrimental impact on controller performance and is an active area of research in the robust and stochastic optimal control communities [6, 21].

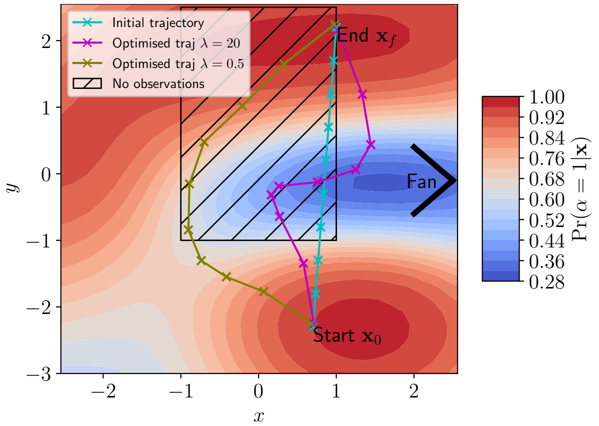

Fig. 1 – This work seeks to velocity control a DJI Tello quadcopter in an indoor environment subject to two modes of operation characterised by process noise (turbulence). A high turbulence mode is induced by placing a desktop fan at the right side of the room. Data from four trajectories following a 2D $\mathbf{x} = (x, y)$ target trajectory captures the variability (process noise) in the dynamics. Our transition dynamics model learns each mode's mixing probability over the domain; the contours show how the probability of dynamics mode 1 ($\alpha = 1$) varies. Our method finds trajectories between $\mathbf{x}_0$ and $\mathbf{x}_f$ that either prioritise remaining in the non-turbulent mode (green) or prioritise avoiding regions of the learned dynamics with high epistemic uncertainty due to lack of training observations (magenta).

The difficulties associated with constructing mathematical models can be overcome by learning from observations [13]. However, learning dynamics models for control introduces other difficulties. For example, it is important to know where

Aidan Scannell is a PhD student at the EPSRC Centre for Doctoral Training in Future Autonomous and Robotic Systems (FARSCOPE) at the Bristol Robotics Laboratory.

the model cannot predict confidently due to a lack of training observations. This concept is known as epistemic uncertainty and is reduced in the limit of infinite data. Probabilistic models have been used to account for epistemic uncertainty and also provide a principled approach to modelling stochasticity i.e. aleatoric uncertainty [4, 20]. For example, [3, 4, 17] use Gaussian processes (GPs) to learn transition dynamics. GPs lend themselves to data-efficient learning through the selection of informative priors, and when used in a Bayesian setting offer well calibrated uncertainty estimates. Methods for learning probabilistic multimodal transition dynamics have also been proposed: [14] use a Mixture of Gaussian Process Experts (MoGPE) method, [16] use deep generative models and [10] use a Bayesian model that learns independent dynamics modes whilst maintaining a probabilistic belief over which mode is responsible for predicting at a given input location.

There has also been work developing control algorithms exploiting learned multimodal transition dynamics [9]. However, our work differs as it seeks to find trajectories that remain in a single dynamics mode whilst avoiding regions of the transition dynamics that cannot be predicted confidently. To the best of our knowledge, there is no previous work addressing such trajectory optimisation in dynamical systems.

Motivated by trajectory optimisation, we adopt (and extend) the well-known MoGPE method with a GP-based gating network [24] to learn a time-invariant transition dynamics model. Our trajectory optimisation formulates a cost function that exploits the geometric structure learned by the GP-based gating network along with its well calibrated uncertainty estimates. We observe that trajectories minimising our cost function are geodesics on a Riemannian manifold parameterised by the desired mode's gating function. With this observation, we exploit a classic result of Riemannian geometry and project the trajectory optimisation onto a continuous-time ODE, whose solutions implicitly minimise our cost function. Solutions to this ODE are trajectories that remain in a single dynamics mode (where possible) and avoid regions of the dynamics that cannot be predicted confidently. We then solve this latent ODE using Hermite-Simpson collocation [11].

The remainder of this paper details our two-stage approach to learning the transition dynamics and performing trajectory optimisation. Section II formally states our problem and Section III details the formulation of the transition dynamics as a probabilistic model. Section III-E details our approach to performing scalable Bayesian inference. Section IV recaps concepts from Riemannian geometry before introducing our geometric cost function. It then details how we implicitly minimise it by projecting the trajectory optimisation onto a latent ODE. Section V gives results of the method tested on a real-world velocity-controlled quadcopter example.

II. Problem Statement

This work is interested in performing trajectory optimisation in multimodal nonlinear systems with unknown transition dynamics. It considers continuous-time, continuous-state, nonlinear stochastic dynamics,

$$\begin{aligned}\dot{\mathbf{x}}(t) &= f(\mathbf{x}(t), \mathbf{u}(t)) + \epsilon(t) \\ &= f^{(k)}(\mathbf{x}(t), \mathbf{u}(t)) + \epsilon^{(k)}(t) \quad \text{if} \quad \alpha(t) = k\end{aligned} \tag{1}$$

with states $\mathbf{x} \in \mathcal{X}$ and controls $\mathbf{u} \in \mathcal{U}$ where $\mathcal{X} = \mathbb{R}^D$ and $\mathcal{U} = \mathbb{R}^F$. One of K dynamics modes $\{f^{(k)}\}_{k=1}^{K}$ and associated noise models $\epsilon^{(k)}(t) \sim \mathcal{N}(0, (\sigma^{(k)})^2)$ are selected by a switching (or gating) variable $\alpha(t) \in \{1, \dots, K\}$.

Trajectory optimisation seeks to find the state and control trajectories $\bar{\mathbf{x}}$, $\bar{\mathbf{u}}$, for times $t \in [t_0, t_f]$ that minimise some cost function g whilst satisfying constraints c and boundary conditions. The trajectory optimisation problem is given by,

$$\begin{aligned}\min_{\mathbf{x}(t), \mathbf{u}(t)} & \int_{t_0}^{t_f} g(\mathbf{x}(t), \mathbf{u}(t)) \mathrm{d}t \quad \forall t \\ \text{s.t. } & \text{Eq. 1} \\ & c(\mathbf{x}(t)) \leq 0 \quad \forall t \\ & \mathbf{x}(t_0) = \mathbf{x}_0, \quad \mathbf{x}(t_f) = \mathbf{x}_f\end{aligned} \tag{2}$$

This work is interested in finding trajectories that,

1) attempt to remain in a preferred dynamics mode k^*,
2) avoid regions of the learned dynamics with high epistemic uncertainty, i.e. that cannot be predicted confidently (due to limited training observations).

The standard approach is to construct a cost function that encodes our two goals. It could take the form,

$$J = \int_{t_0}^{t_f} g_{\text{mode}}(\mathbf{x}(t), \mathbf{u}(t)) + g_{\text{epistemic}}(\mathbf{x}(t), \mathbf{u}(t)) \mathrm{d}t \tag{3}$$

where g_{mode} favours remaining in dynamics mode k^* and $g_{\text{epistemic}}$ favours trajectories avoiding regions of the dynamics with high epistemic uncertainty. We follow this approach and construct a cost function based on probabilistic Riemannian geometry. We then exploit a classic result of Riemannian geometry and project the trajectory optimisation onto an ODE whose solutions implicitly minimise it.

III. Stage One - Model Learning

The first stage of our method learns a probabilistic representation of the transition dynamics using the well known MoGPE model. It is capable of modelling both the aleatoric uncertainty inherent in the system as well as the epistemic uncertainty associated with learning from observations. We assume access to historical data comprising state transitions from E trajectories of length N sampled with a fixed time step $\Delta_t = t_*$. The data set has $T = EN$ elements, and we abuse notation and append the independent trajectories along time to get the data set $\mathcal{D} = \{(\mathbf{x}_{t-1}, \mathbf{u}_{t-1}), \Delta\mathbf{x}_t\}_{t=1}^{T}$.

In this section we first introduce GPs and the sparse GP approximation that is used throughout. We then formulate Eq. 1 as a discrete-time probabilistic transition dynamics model and derive a novel variational lower bound based on sparse GPs which enables scalable Bayesian inference.

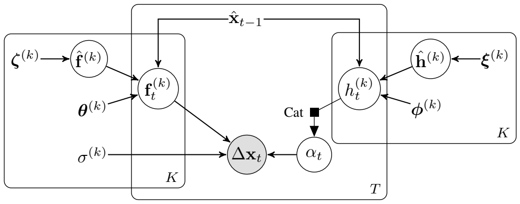

Fig. 2 – Graphical model of the transition dynamics where the state difference $\Delta\mathbf{x}_t$ is generated by pushing the state and control $\hat{\mathbf{x}}_{t-1}$ through the latent process.

A. Sparse Gaussian Processes

A GP [18] is a distribution over functions $f : \mathbb{R}^{D_f} \rightarrow \mathbb{R}$ fully defined by a mean function $\mu(\cdot)$ and a covariance function $k(\cdot,\cdot)$. For a given set of inputs from the function's domain $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ the associated function values $\mathbf{f} = \{f(\mathbf{x}_1), \dots, f(\mathbf{x}_N)\}$ are jointly Gaussian,

$$p(\mathbf{f} \mid \mathbf{X}) = \mathcal{N}(\mathbf{f} \mid \boldsymbol{\mu}_{\mathbf{x}}, \mathbf{K}_{\mathbf{xx}}), \tag{4}$$

where $\boldsymbol{\mu}_{\mathbf{x}} = \mu(\mathbf{X})$ is the mean vector, and $\mathbf{K}_{\mathbf{xx}} = k(\mathbf{X}, \mathbf{X})$ is the covariance function evaluated between the inputs $\mathbf{X}$. In this work, the squared exponential covariance function with Automatic Relevance Determination is used for all GPs. The distribution over the function value f_* at a new input $\mathbf{x}_*$ (i.e. to make a prediction) is given by the conditional,

$$p(f_* \mid \mathbf{x}_*, \mathbf{f}, \mathbf{X}) = \mathcal{N}(\mu_* + \mathbf{k}_{*\mathbf{x}}\mathbf{K}_{\mathbf{xx}}^{-1}(\mathbf{f} - \boldsymbol{\mu}_{\mathbf{x}}), k_{**} - \mathbf{k}_{*\mathbf{x}}\mathbf{K}_{\mathbf{xx}}^{-1}\mathbf{k}_{\mathbf{x}*}).$$

This GP conditional is computationally expensive due to conditioning on all of the training data $\mathbf{X}, \mathbf{f}$. Introducing a set of M inducing points from the same GP prior can reduce the computational cost if $M < N$. The inducing inputs are denoted $\boldsymbol{\xi}$ and outputs as $\hat{\mathbf{f}} = f(\boldsymbol{\xi})$. The inducing outputs $\hat{\mathbf{f}}$ are jointly Gaussian with the latent function values $\mathbf{f}$ so the GP predictive distribution can be approximated by conditioning on this smaller set of inducing points,

$$\begin{aligned} &p(f_* \mid \mathbf{x}_*, \mathbf{f}, \mathbf{X}) \approx p(f_* \mid \mathbf{x}_*, \hat{\mathbf{f}}, \boldsymbol{\xi}) \qquad (5) \\ &\quad = \mathcal{N}\left(\mu_* + \mathbf{k}_{*\boldsymbol{\xi}}\mathbf{K}_{\boldsymbol{\xi\xi}}^{-1}(\hat{\mathbf{f}} - \boldsymbol{\mu}_{\boldsymbol{\xi}}), k_{**} - \mathbf{k}_{*\boldsymbol{\xi}}\mathbf{K}_{\boldsymbol{\xi\xi}}^{-1}\mathbf{k}_{\boldsymbol{\xi}*}\right). \end{aligned}$$

The approximation becomes exact when the inducing points $\hat{\mathbf{f}}$ are a sufficient statistic for the latent function values $\mathbf{f}$ [22].

B. Model Definition

This work learns a discrete-time representation of Eq. 1,

$$\Delta\mathbf{x}_t = f^{(k)}(\mathbf{x}_{t-1}, \mathbf{u}_{t-1}; \Delta t = t_*) + \epsilon_{t-1}^{(k)} \quad \text{if} \quad \alpha_t = k, \tag{6}$$

where $\mathbf{x}_t \in \mathbb{R}^D$ and $\mathbf{u}_t \in \mathbb{R}^F$ are the states and controls at time t respectively, and $\alpha_t \in \{1, \dots, K\}$ is a mode indicator variable that indicates one of K dynamics modes at time t.

A time series of observations from time a to time b (inclusive) is denoted by $\mathbf{x}_{a:b}$ (analogously for other variables). A single input is denoted as $\hat{\mathbf{x}}_{t-1} = (\mathbf{x}_{t-1}, \mathbf{u}_{t-1})$, all inputs are denoted as $\hat{\mathbf{x}}_{1:T}$, and the set of all outputs as $\Delta\mathbf{x}_{1:T}$. The d^{th} dimension of the k^{th} mode's latent transition dynamics function $f^{(k)}$, evaluated at $\hat{\mathbf{x}}_{t-1}$, is denoted $f_{t,d}^{(k)} = f_d^{(k)}(\hat{\mathbf{x}}_{t-1})$, for all dimensions as $\mathbf{f}_t^{(k)}$ and at all data points as $\mathbf{f}_{1:T}^{(k)}$.

The model is built upon sparse GP priors on each of the transition dynamics functions $f^{(k)}$ with independent GPs placed on each state dimension d,

$$p\left(\mathbf{f}_{1:T}^{(k)} \mid \hat{\mathbf{x}}_{1:T}, \hat{\mathbf{f}}^{(k)}\right) = \prod_{t=1}^{T}\prod_{d=1}^{D} p\left(f_{t,d}^{(k)} \mid \hat{\mathbf{x}}_{t-1}, \hat{\mathbf{f}}_d^{(k)}\right) \tag{7}$$

where $p\left(f_{t,d}^{(k)} \mid \hat{\mathbf{x}}_{t-1}, \hat{\mathbf{f}}_d^{(k)}\right)$ is a sparse GP conditional (Eq. 5). The M inducing inputs and outputs associated with the d^{th} dimension of the k^{th} mode's latent function $f^{(k)}$ are denoted as $\boldsymbol{\zeta}_d^{(k)}$ and $\hat{\mathbf{f}}_d^{(k)}$ respectively. They are collected as $\boldsymbol{\zeta}^{(k)}$ and $\hat{\mathbf{f}}^{(k)}$ for all output dimensions and as $\boldsymbol{\zeta}$ and $\hat{\mathbf{f}}$ for all modes. For notational conciseness, the dependency on the inducing inputs $\boldsymbol{\zeta}^{(k)}$ is dropped throughout. The process noise in each mode is modelled as,

$$p\left(\Delta\mathbf{x}_t | \mathbf{f}_t^{(k)}\right) = \mathcal{N}\left(\Delta\mathbf{x}_t \mid \mathbf{f}_t^{(k)}, \text{diag}\left[\left(\sigma_1^{(k)}\right)^2, \dots, \left(\sigma_D^{(k)}\right)^2\right]\right),$$

where $\left(\sigma_d^{(k)}\right)^2$ represents the noise variance associated with the d^{th} dimension of the k^{th} mode.

C. Gating Network

The gating network governs how the dynamics switch between modes. This work is interested in spatially varying modes so formulates an input dependent Categorical distribution over the mode indicator variable α_t,

$$P\left(\alpha_t \mid \mathbf{h}_t\right) = \prod_{k=1}^{K}\left(\Pr(\alpha_t = k \mid \mathbf{h}_t)\right)^{[\alpha_t = k]} = \text{softmax}(\mathbf{h}_t), \tag{8}$$

where $[\alpha_t = k]$ denotes the Iverson bracket. The probabilities of this Categorical distribution $\Pr(\alpha_t = k \mid \mathbf{h}_t)$ are obtained by evaluating K latent gating functions $\{h^{(k)}\}_{k=1}^{K}$ and normalising their output. Each gating function evaluated at $\hat{\mathbf{x}}_{t-1}$ is denoted as $h_t^{(k)} = h^{(k)}(\hat{\mathbf{x}}_{t-1})$ and at all observations $h_{1:T}^{(k)}$. The set of all gating functions evaluated at $\hat{\mathbf{x}}_{t-1}$ is denoted as $\mathbf{h}_t$ and at all observations as $\mathbf{h}_{1:T}$. Each gating function $h^{(k)}$ describes how its corresponding mode's mixing probability varies over the input space.

This work is interested in finding trajectories that can avoid areas of the transition dynamics model that cannot be predicted confidently. Placing GP priors on each gating function provides a principled approach to modelling the epistemic uncertainty associated with each gating function. The gating function's posterior covariance is a quantitative value that can be exploited by the trajectory optimisation.

Each gating function's inducing inputs are denoted $\boldsymbol{\xi}^{(k)}$ and outputs as $\hat{\mathbf{h}}^{(k)}$. For all gating functions, they are collected as $\boldsymbol{\xi}$ and $\hat{\mathbf{h}}$ respectively. The probability that the t^{th} observation is generated by mode k given the inducing inputs is obtained by marginalising the set of gating functions $\mathbf{h}_t$,

$$\Pr(\alpha_t = k \mid \hat{\mathbf{x}}_{t-1}, \hat{\mathbf{h}}) = \int \text{softmax}_k(\mathbf{h}_t) p(\mathbf{h}_t \mid \hat{\mathbf{x}}_{t-1}, \hat{\mathbf{h}}) \text{d}\mathbf{h}_t,$$

where $p(\mathbf{h}_t \mid \hat{\mathbf{x}}_{t-1}, \hat{\mathbf{h}}) = \prod_{k=1}^{K} p\left(h_t^{(k)} \mid \hat{\mathbf{x}}_{t-1}, \hat{\mathbf{h}}^{(k)}\right)$ is the K independent sparse GP priors on the gating functions.

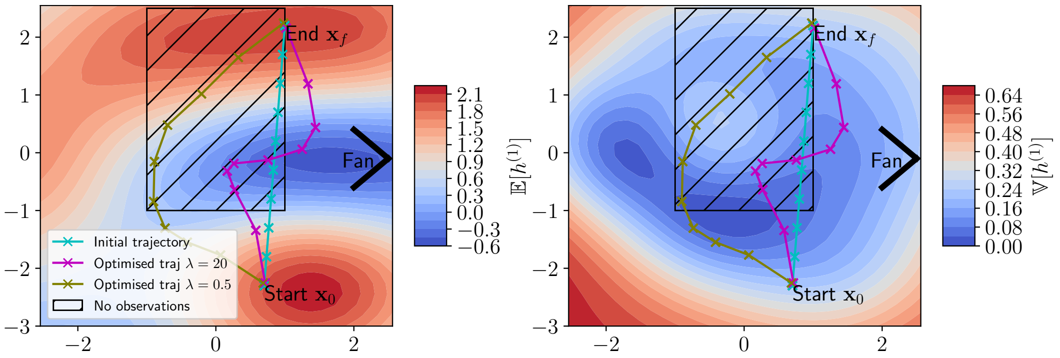

Fig. 3 – Contour plots showing the GP posterior mean (left) and variance (right) over the gating function associated with dynamics mode 1 after training on a subset of the quadcopter data set. The initial and optimised trajectories are overlayed to show the influence of the GP's mean and variance on the trajectory optimisation with different λ settings.

D. Generative Model

This model makes single-step probabilistic predictions, where the predictive distribution over the state difference $\Delta\mathbf{x}_t$ is given by a mixture of Gaussians. This provides the model flexibility to model multimodal transition dynamics f as mixtures of K modes. With this formulation, the marginal likelihood can be rewritten as,

$$p(\Delta\mathbf{x}_{1:T}|\hat{\mathbf{x}}_{1:T}) = \prod_{t=1}^{T}\sum_{k=1}^{K}\Bigg(\underbrace{\left\langle \Pr\left(\alpha_t = k|\hat{\mathbf{x}}_{t-1},\hat{\mathbf{h}}\right)\right\rangle_{p(\hat{\mathbf{h}}|\boldsymbol{\xi})}}_{\text{Mixing Probability}} \underbrace{\left\langle p\left(\Delta\mathbf{x}_t|\hat{\mathbf{x}}_{t-1},\hat{\mathbf{f}}^{(k)}\right)\right\rangle_{p\left(\hat{\mathbf{f}}^{(k)}|\boldsymbol{\zeta}^{(k)}\right)}}_{\text{Dynamics mode } k}\Bigg), \qquad (9)$$

where $\langle\cdot\rangle_{p(x)}$ denotes an expectation under $p(x)$. Fig. 2 shows the graphical model where the K latent gating functions $h^{(k)}$ are evaluated and normalised to obtain the mixing probabilities $\Pr(\alpha_t = k \mid \hat{\mathbf{x}}_{t-1})$. The mode indicator variable α_t is then sampled from a Categorical distribution governed by these probabilities. The indicated mode's latent function $f^{(k)}$ and process noise $\sigma^{(k)}$ are then evaluated to generate the state difference $\Delta\mathbf{x}_t$.

E. Inference

The marginal likelihood in Eq. 9 is based upon sparse GP priors which become exact when $\boldsymbol{\xi} = \boldsymbol{\zeta} = \hat{\mathbf{x}}_{1:T}$. We derive a variational approximation based on sparse approximations that provides scalability via stochastic gradient-based optimisation. Following the approach by [22], the probability space has been augmented with a set of M inducing points for each GP. Instead of collapsing these inducing points, they are explicitly represented as a variational distribution, as seen in [8]. A mean-field approximation is used for each GP's inducing variable distribution,

$$q\left(\hat{\mathbf{f}},\hat{\mathbf{h}}\right) = \prod_{k=1}^{K}\left(\mathcal{N}\left(\hat{\mathbf{h}}^{(k)} \mid \mathbf{m}_h^{(k)}, \mathbf{S}_h^{(k)}\right)\right. \qquad (10)$$
$$\left.\prod_{d=1}^{D}\mathcal{N}\left(\hat{\mathbf{f}}_d^{(k)} \mid \mathbf{m}_{f,d}^{(k)}, \mathbf{S}_{f,d}^{(k)}\right)\right).$$

This variational distribution and Jensen's inequality are used to lower bound the log marginal likelihood $\log p(\Delta\mathbf{x}_{1:T} \mid \hat{\mathbf{x}}_{1:T})$,

$$\mathcal{L} = \sum_{t=1}^{T}\left\langle \log\sum_{k=1}^{K}\Pr(\alpha_t = k \mid \hat{\mathbf{x}}_{t-1},\hat{\mathbf{h}})p(\Delta\mathbf{x}_t \mid \hat{\mathbf{x}}_{t-1},\hat{\mathbf{f}}^{(k)})\right\rangle_{q(\hat{\mathbf{f}},\hat{\mathbf{h}})}$$
$$-\sum_{k=1}^{K}\mathrm{KL}\left(q\left(\hat{\mathbf{f}}^{(k)}\right) \mid\mid p\left(\hat{\mathbf{f}}^{(k)} \mid \boldsymbol{\zeta}^{(k)}\right)\right)$$
$$-\sum_{k=1}^{K}\mathrm{KL}\left(q\left(\hat{\mathbf{h}}^{(k)}\right) \mid\mid p\left(\hat{\mathbf{h}}^{(k)} \mid \boldsymbol{\xi}^{(k)}\right)\right). \qquad (11)$$

Importantly, taking samples for single data points is straightforward and can be implemented efficiently. The variational expectation is not analytically tractable due to the variational distributions, so it is approximated by drawing single samples from $q(\hat{\mathbf{f}})$ and $q(\hat{\mathbf{h}})$. The inducing inputs and kernel hyperparameters are optimised alongside the variational parameters.

F. Results

The model is trained on data collected from the velocity-controlled quadcopter experiment. The controls were kept constant during data collection to reduce the dynamics model to $\Delta\mathbf{x}_t = f(\mathbf{x}_{t-1})$. The trajectory optimisation then exploits differential flatness [19] to recover the velocity controls. The model was trained with $K = 2$ dynamics modes, and a subset of the observations were withheld during training to test the model's ability to model epistemic uncertainty. Fig. 1 shows mode 1's mixing probability over the domain which has clearly learned two dynamics modes characterised by process noise. Fig. 3 shows the predictive mean (left) and variance (right) of the gating function associated with dynamics mode 1 ($h^{(1)}$). The mean is high where the model believes mode 1 is responsible for predicting, low where it believes another mode is responsible, and zero where it is uncertain. The variance (right) has also clearly captured information regrading the epistemic uncertainty, i.e. where there are no observations.

IV. Stage Two - Trajectory Optimisation

In stage one, our method learns a probabilistic representation of the transition dynamics using a MoGPE model.

Importantly, the GP-based gating network infers important information regarding how the transition dynamics switch between modes over the input space. The length of a trajectory $\bar{\mathbf{x}}$ from $\mathbf{x}_0$ to $\mathbf{x}_f$ on the manifold parameterised by the desired mode's gating function (Fig. 3 left) increases when it passes over the contours - analogous to climbing a hill. Given appropriate scaling, shortest trajectories on the manifold will be those that attempt to follow the contours and remain in a single dynamics mode.

In stage two, we exploit this intuition and formulate a geometric cost function to solve the trajectory optimisation in Eq. 2, i.e. find trajectories from $\mathbf{x}_0$ to $\mathbf{x}_f$ that minimise the cost in Eq. 3. We then exploit a classic result of Riemannian geometry that allows us to project the trajectory optimisation onto a latent ODE whose solutions implicitly minimise our geometric cost. This section now introduces our cost function, recaps concepts of Riemannian geometry and then details how this latent ODE is solved using direct collocation.

A. A Geometric Cost Function

The g_{mode} term in Eq. 3 can be expressed as finding shortest trajectories $\bar{\mathbf{x}}$ on the manifold parameterised by the desired mode's gating function. We can measure lengths on the manifold by mapping the trajectory $\bar{\mathbf{x}}$ through the desired mode's gating function $h^{(k)}$,

$$\text{Length}\left(h^{(k)}(\bar{\mathbf{x}})\right) = \int_{t_0}^{t_f} \left\| \dot{h}^{(k)}(\mathbf{x}(t)) \right\| \mathrm{d}t = \int_{t_0}^{t_f} \| \mathbf{J}_{\mathbf{x}_t} \dot{\mathbf{x}}(t) \| \, \mathrm{d}t,$$
$$\mathbf{J}_{\mathbf{x}_t} = \frac{\partial h^{(k)}}{\partial \mathbf{x}(t)} \in \mathbb{R}^{1 \times D} \tag{12}$$

This implies that the length of a trajectory on the manifold can be calculated in the input space using a locally defined norm,

$$\| \mathbf{J}_{\mathbf{x}_t} \dot{\mathbf{x}}(t) \| = \sqrt{\dot{\mathbf{x}}(t) \mathbf{J}_{\mathbf{x}_t}^T \mathbf{J}_{\mathbf{x}_t} \dot{\mathbf{x}}(t)} = \sqrt{\dot{\mathbf{x}}(t) \mathbf{G}_{\mathbf{x}_t} \dot{\mathbf{x}}(t)} \tag{13}$$

where $\mathbf{G}_{\mathbf{x}_t}$ is a symmetric positive definite matrix, akin to a local Mahalanobis distance measure. This gives rise to the definition of a Riemannian metric, a smoothly changing inner product structure [2],

***Definition** 4.1 (Riemannian Metric):* A Riemannian metric $\mathbf{G} : \mathcal{X} \to \mathbb{R}^{D \times D}$ is a smooth function that assigns a symmetric positive definite matrix to any point in $\mathcal{X}$.

Given this definition, we formulate a geometric cost function,

$$J_{\text{geo}} = \min \text{Length}(h^{(k)}(\bar{\mathbf{x}})) = \min \int_{t_0}^{t_f} \| \dot{\mathbf{x}}(t) \|_{\mathbf{G}(\mathbf{x}(t))} \, \mathrm{d}t \tag{14}$$

where $\mathbf{G}(\mathbf{x}(t))$ represents the Riemannian metric tensor associated with the desired mode's gating function. Eq. 14 has encoded the g_{mode} term from Eq. 3 but not the $g_{\text{epistemic}}$ term. We address this by observing that our manifold is stochastic (because it is parameterised by a stochastic function) and extending our metric for probabilistic geometries.

Following [23] we use a metric tensor that captures the variance in the manifold by means of a probability distribution. If a GP is differentiable, then its derivative is also a GP (because the differential operator is linear). Thus, the predictive distribution over the Jacobian given a new input $\mathbf{J}_* = \mathbf{J}(\mathbf{x}_*)$ can be obtained by conditioning on the inducing variables $\hat{\mathbf{h}}^{(k)}$ and then marginalising them with the variational distribution $q\left(\hat{\mathbf{h}}^{(k)}\right)$ [12],

$$p\left(\mathbf{J}_* | \mathbf{x}_*, \boldsymbol{\xi}^{(k)}\right) = \int q\left(\hat{\mathbf{h}}^{(k)}\right) p\left(\mathbf{J}_* \mid \mathbf{x}_*, \hat{\mathbf{h}}^{(k)}, \boldsymbol{\xi}^{(k)}\right) \mathrm{d}\hat{\mathbf{h}}^{(k)}$$
$$= \mathcal{N}\left(\mathbf{J}_* \mid \boldsymbol{\mu}_J, \boldsymbol{\Sigma}_J\right). \tag{15}$$

This induces a non-central Wishart distribution over the metric tensor $\mathbf{G}$,

$$\mathbf{G} = \mathcal{W}_D\left(p, \boldsymbol{\Sigma}_J, \mathbb{E}\left[\mathbf{J}^T\right] \mathbb{E}[\mathbf{J}]\right), \tag{16}$$

where p is the number of degrees of freedom (always one in our case). The expected metric tensor is given by,

$$\mathbb{E}[\mathbf{G}] = \mathbb{E}[\mathbf{J}^T] \, \mathbb{E}[\mathbf{J}] + \boldsymbol{\Sigma}_J. \tag{17}$$

This expected metric tensor includes a covariance term $\boldsymbol{\Sigma}_J$ which implies that lengths on the manifold increase in areas of high covariance. This is a desirable behaviour because it encourages trajectories minimising Eq. 14 to avoid regions of the learned dynamics with high epistemic uncertainty, encoding the $g_{\text{epistemic}}$ cost term in Eq. 3.

B. Implicit Trajectory Optimisation

Trajectories minimising Eq. 14 are length minimising curves on the Riemannian manifold endowed with the metric $\mathbf{G}$, and are known as geodesics. A classic result of differential geometry [2] is that solutions to Eq. 14 (i.e. geodesic trajectories) must satisfy the second-order ODE,

$$\ddot{\mathbf{x}}(t) = f_G(t, \dot{\mathbf{x}}, \mathbf{x})$$
$$= -\frac{1}{2} \mathbf{G}^{-1}(\mathbf{x}(t)) \left[\frac{\partial \, \text{vec}[\mathbf{G}(\mathbf{x}(t))]}{\partial \mathbf{x}(t)} \right]^T (\dot{\mathbf{x}}(t) \otimes \dot{\mathbf{x}}(t)), \tag{18}$$

where $\text{vec}[\mathbf{G}(\mathbf{x}(t)])$ stacks the columns of $\mathbf{G}(\mathbf{x}(t))$ and $\otimes$ denotes the Kronecker product. Thus, projecting our trajectory optimisation onto Eq. 18 with the expected metric from Eq. 17 is equivalent to solving the trajectory optimisation in Eq. 2 with the cost function in Eq. 3.

However, since neither $\dot{\mathbf{x}}(t_0)$ nor $\dot{\mathbf{x}}(t_f)$ are known, it cannot be solved with simple forward or backward integration. Instead, the problem is transcribed using differential flatness [15, 19]. A set of outputs $\mathbf{z}(t)$ are defined such that the states $\mathbf{x}(t)$ and controls $\mathbf{u}(t)$ can be expressed in terms of the flat output $\mathbf{z}(t)$ and a finite number of its derivatives,

$$\mathbf{x}(t) = A(\mathbf{z}(t), \dot{\mathbf{z}}(t), \ldots) \tag{19}$$
$$\mathbf{u}(t) = B(\mathbf{z}(t), \dot{\mathbf{z}}(t), \ldots). \tag{20}$$

In the velocity-controlled quadcopter example, the flat output is the state $\mathbf{z}(t) = \mathbf{x}(t)$ and the control is simply the state derivative $\mathbf{u}(t) = \dot{\mathbf{z}}(t)$. The original trajectory optimisation problem can then be converted to finding $\mathbf{z}(t)$ and $\dot{\mathbf{z}}(t)$ for $t \in [t_0, t_f]$ subject to the boundary conditions and the dynamics,

$$\ddot{\mathbf{x}}(\mathbf{z}(t), \dot{\mathbf{z}}(t)) = f_G(t, \dot{\mathbf{x}}(\mathbf{z}(t), \dot{\mathbf{z}}(t)), \mathbf{x}(\mathbf{z}(t), \dot{\mathbf{z}}(t))). \tag{21}$$

Collocation methods are used to transcribe continuous-time trajectory optimisation problems into nonlinear programs, i.e. constrained parameter optimisation [5, 11]. The expected

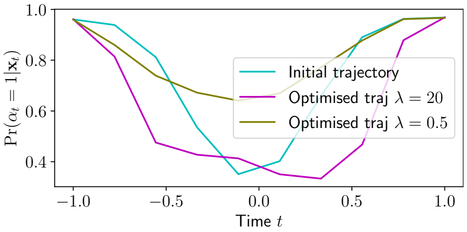

Fig. 4 – Comparision of the intial and optimised trajectories' performance at staying in the desired mode. The plot shows mode 1's mixing probability over the trajectories for two settings of λ.

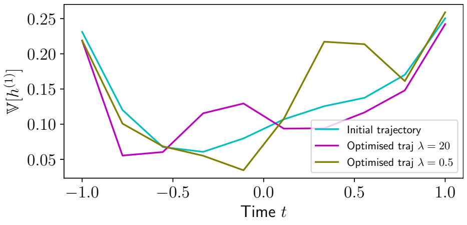

Fig. 5 – Comparison of the initial and optimised trajectories' at avoiding regions of high epistemic uncertainty. It shows the posterior variance associated with the desired mode's gating function.

metric in Eq. 17 is substituted into Eq. 21 and solved via direct collocation. This work implements a simple Hermite-Simpson collocation method that enforces the state derivative interpolated by the polynomials to equal the geodesic ODE f_G at the midpoints between a set of I collocation points $\{\mathbf{z}_i\}_{i=1}^{I}$. This is achieved through the collocation defects,

$$\Delta_{i+\frac{1}{2}} = \ddot{\mathbf{z}}_{i+\frac{1}{2}} - f_G(t_{i+\frac{1}{2}}, \dot{\mathbf{z}}_{i+\frac{1}{2}}, \mathbf{z}_{i+\frac{1}{2}}) \tag{22}$$

where $\ddot{\mathbf{z}}_{i+\frac{1}{2}}$, $\dot{\mathbf{z}}_{i+\frac{1}{2}}$, $\mathbf{z}_{i+\frac{1}{2}}$ are obtained by interpolating between $\mathbf{z}_i$ and $\mathbf{z}_{i+1}$. Eq. 22 defines a set of constraints ensuring trajectories are solutions to the geodesic ODE f_G. The nonlinear program that this work solves is given by,

$$\min_{\mathbf{z}(t), \dot{\mathbf{z}}(t)} \int_{t_0}^{t_f} 1 \mathrm{d}t \tag{23}$$

$$\text{s.t. Eq. 22,} \tag{24}$$

$$\mathbf{x}\left(\mathbf{z}(t_0), \dot{\mathbf{z}}(t_0)\right) = \mathbf{x}_0, \tag{25}$$

$$\mathbf{x}(\mathbf{z}(t_f), \dot{\mathbf{z}}(t_f)) = \mathbf{x}_f \tag{26}$$

This is solved using Sequential Least Squares Programming (SLSQP) in SciPy.

V. Quadcopter Experiments

The trajectory optimisation is tested on the quadcopter control problem shown in Fig. 1. It projects the optimisation onto the desired mode's learned gating function, shown in Fig. 3. It seeks to find trajectories between $\mathbf{x}_0$ and $\mathbf{x}_f$ that 1) remain in the non-turbulent mode and 2) avoid regions of the learned transition dynamics with high epistemic uncertainty. To aid with user control the metric tensor in Eq. 17 is modified with a weighting parameter λ that enables the relevance of the covariance term to be adjusted,

$$\tilde{\mathbf{G}} = \mathbb{E}[\mathbf{J}^T]\,\mathbb{E}[\mathbf{J}] + \lambda \mathbf{\Sigma}_J. \tag{27}$$

TABLE I – Comparison of performance with different settings of λ. The performance measures are summed over collocation points.

Trajectory	Mixing Probability $\sum_{i=1}^{I} \Pr(\alpha_i = 1 \mid \mathbf{z}_i)$	Epistemic Uncertainty $\sum_{i=1}^{I} \mathbb{V}[h^{(1)}(\mathbf{z}_i)]$
Initial	7.480	1.345
Optimised $\lambda = 20$	6.091	$\mathbf{1.274}$
Optimised $\lambda = 0.5$	$\mathbf{8.118}$	1.437

Setting λ to be small should find trajectories that prioritise staying in the desired mode, whereas selecting a large λ should find trajectories that prioritise avoiding regions of the dynamics with high epistemic uncertainty. The trajectory optimisation is tested with two λ settings.

The initial (cyan) trajectory in Fig. 3 is initialised with 10 collocation points indicated by the crosses. The trajectories are compared via mode 1's mixing probability and its gating function's GP variance over the trajectories. We want to maximise the probability of being in our desired mode whilst minimising the amount of variance (due to epistemic uncertainty). The results are shown in Table I. It is clear from Fig. 3 (left) and Fig. 4 that for $\lambda = 0.5$, trajectories favour remaining in dynamics mode 1 at the cost of entering regions of the learned dynamics with high epistemic uncertainty (shown in Fig. 5). For $\lambda = 20$, the trajectory has tried to remain in dynamics mode 1 at the start of the trajectory but then hits the area of high epistemic uncertainty and favours avoiding this region over remaining in dynamics mode 1.

Although not tested, we believe that our approach is theoretically sound and can easily be extended to more than two dynamics modes. However, it is interesting to consider if this is even necessary given our goals. For example, in the quadcopter experiment, we intentionally instantiated the transition dynamics model with two dynamics modes, although in reality there could be more. We engineered our desired dynamics mode to have a noise variance that we deemed operatable. We then used the other dynamics mode to explain away all of the un-operatable modes. We think that in most scenarios a similar approach could be followed.

VI. Conclusion

This paper presents our novel two-stage method for performing trajectory optimisation in unknown multimodal dynamical systems. The first stage learns a probabilistic transition dynamics model using a MoGPE method with a GP-based gating network. The trajectory optimisation is then projected onto a probabilistic Riemannian manifold parameterised by the gating network. The method is evaluated on a real-world quadcopter example that shows the transition dynamics model can successfully learn a factorised representation of the underlying dynamics modes. Given a start and end state, the trajectory optimisation can be tuned to find trajectories that either prioritise 1) remaining in a desired dynamics mode or 2) avoiding regions of high epistemic uncertainty. In future work it would be interesting to explore active learning techniques that can avoid entering the undesired dynamics modes during data collection.

References

[1] John T. Betts. "Survey of Numerical Methods for Trajectory Optimization". In: *Journal of Guidance, Control, and Dynamics* 21.2 (Mar. 1, 1998), pp. 193–207.

[2] Manfredo do Carmo. *Riemannian Geometry*. Mathematics: Theory & Applications. Birkhäuser Basel, 1992.

[3] M. Cutler and J. P. How. "Efficient Reinforcement Learning for Robots Using Informative Simulated Priors". In: IEEE International Conference on Robotics and Automation. IEEE, May 2015, pp. 2605–2612.

[4] Marc Deisenroth and Carl Rasmussen. "PILCO: A Model-Based and Data-Efficient Approach to Policy Search." In: *International Conference on Machine Learning*. Vol. 28. Jan. 1, 2011, pp. 465–472.

[5] F. Fahroo and I. M. Ross. "Direct Trajectory Optimization by a Chebyshev Pseudospectral Method". In: *Proceedings of the 2000 American Control Conference*. American Control Conference. Vol. 6. June 2000, pp. 3860–3864.

[6] Randy Freeman and Petar V. Kokotovic. *Robust Nonlinear Control Design: State-Space and Lyapunov Techniques*. Modern Birkhäuser Classics. Birkhäuser Basel, 1996.

[7] Divya Garg et al. "A Unified Framework for the Numerical Solution of Optimal Control Problems Using Pseudospectral Methods". In: *Automatica* 46.11 (Nov. 1, 2010), pp. 1843–1851.

[8] James Hensman, Nicolo Fusi, and Neil D Lawrence. "Gaussian Processes for Big Data". In: *Proceedings of the 29th Conference on Uncertainty in Artificial Intelligence*. Uncertainty in Artificial Intelligence. Vol. 29. 2013, pp. 282–290.

[9] Randa Herzallah and David Lowe. "PMAC: Probabilistic Multimodality Adaptive Control". In: *International Journal of Control* 93.7 (July 2, 2020), pp. 1637–1650.

[10] Markus Kaiser, Clemens Otte, Thomas A. Runkler, and Carl Henrik Ek. "Bayesian Decomposition of Multi-Modal Dynamical Systems for Reinforcement Learning". In: *Neurocomputing* 416 (Nov. 27, 2020), pp. 352–359.

[11] Matthew Kelly. "An Introduction to Trajectory Optimization: How to Do Your Own Direct Collocation". In: *SIAM Review* 59.4 (Jan. 2017), pp. 849–904.

[12] Felix Leibfried, Vincent Dutordoir, S. T. John, and Nicolas Durrande. *A Tutorial on Sparse Gaussian Processes and Variational Inference*. Feb. 2, 2021. arXiv: 2012.13962 `[cs, stat]`. URL: http://arxiv.org/abs/2012.13962.

[13] Lennart Ljung. *System Identification: Theory for the User*. 2nd ed. Prentice Hall Information and System Sciences Series. Pearson, 1999. 640 pp.

[14] C. D. McKinnon and A. P. Schoellig. "Learning Multimodal Models for Robot Dynamics Online with a Mixture of Gaussian Process Experts". In: *IEEE International Conference on Robotics and Automation*. IEEE, May 2017, pp. 322–328.

[15] M. B. Milam, K. Mushambi, and R. M. Murray. "A New Computational Approach to Real-Time Trajectory Generation for Constrained Mechanical Systems". In: *Proceedings of the 39th IEEE Conference on Decision and Control*. IEEE Conference on Decision and Control. Vol. 1. IEEE, Dec. 2000, pp. 845–851.

[16] Thomas M. Moerland, Joost Broekens, and Catholijn M. Jonker. *Learning Multimodal Transition Dynamics for Model-Based Reinforcement Learning*. Aug. 8, 2017. arXiv: 1705.00470.

[17] Yunpeng Pan and Evangelos Theodorou. "Probabilistic Differential Dynamic Programming". In: *Advances in Neural Information Processing Systems*. Vol. 27. 2014, pp. 1907–1915.

[18] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. Adaptive Computation and Machine Learning. Cambridge, Mass: MIT Press, 2006. 248 pp.

[19] I. M. Ross and F. Fahroo. "Pseudospectral Methods for Optimal Motion Planning of Differentially Flat Systems". In: *IEEE Transactions on Automatic Control* 49.8 (Aug. 2004), pp. 1410–1413.

[20] J. Schneider. "Exploiting Model Uncertainty Estimates for Safe Dynamic Control Learning". In: *Advances in Neural Information Processing Systems*. Vol. 9. 1996, pp. 1047–1053.

[21] Robert F. Stengel. *Stochastic Optimal Control: Theory and Application*. John Wiley & Sons, Inc., 1986. 638 pp.

[22] Michalis Titsias. "Variational Learning of Inducing Variables in Sparse Gaussian Processes". In: *Artificial Intelligence and Statistics*. Artificial Intelligence and Statistics. PMLR, Apr. 15, 2009, pp. 567–574.

[23] Alessandra Tosi, Søren Hauberg, Alfredo Vellido, and Neil D Lawrence. "Metrics for Probabilistic Geometries". In: *Proceedings of the 30th Conference*. Uncertainty in Artificial Intelligence. 2014, pp. 800–808.

[24] Volker Tresp. "Mixtures of Gaussian Processes". In: *Advances in Neural Information Processing Systems*. Vol. 13. 2000, pp. 654–660.

[25] Oskar Von Stryk and Roland Bulirsch. "Direct and Indirect Methods for Trajectory Optimization". In: *Annals of Operations Research* 37 (Dec. 1, 1992), pp. 357–373.