Learning Multimodal Models for Robot Dynamics Online with a Mixture of Gaussian Process Experts

Christopher D. McKinnon and Angela P. Schoellig

***Abstract*— For decades, robots have been essential allies alongside humans in controlled industrial environments like heavy manufacturing facilities. However, without the guidance of a trusted human operator to shepherd a robot safely through a wide range of conditions, they have been barred from the complex, ever changing environments that we live in from day to day. Safe learning control has emerged as a promising way to start bridging algorithms based on first principles to complex real-world scenarios by using data to adapt, and improve performance over time. Safe learning methods rely on a good estimate of the robot dynamics and of the bounds on modelling error in order to be effective. Current methods focus on either a single adaptive model, or a fixed, known set of models for the robot dynamics. This limits them to static or slowly changing environments. This paper presents a method using Gaussian Processes in a Dirichlet Process mixture model to learn an increasing number of non-linear models for the robot dynamics. We show that this approach enables a robot to re-use past experience from an arbitrary number of previously visited operating conditions, and to automatically learn a new model when a new and distinct operating condition is encountered. This approach improves the robustness of existing Gaussian Process-based models to large changes in dynamics that do not have to be specified ahead of time.**

I. INTRODUCTION

At the core of most control algorithms in robotics is a model that captures the relationship between the state, the input, and the dynamics of a robotic system. The model can be used to optimize a reward function and to ensure that the system achieves its goals in a safe and reliable way [1], [2]. If the model for the system is partially unknown, the reward function can incorporate an element to encourage exploration of the system dynamics [3], [4]. This establishes a better mapping between the state, input, and dynamics, such that the controller can later exploit well-known, high-reward actions [3]. An accurate assessment of the risk associated with taking a control action, especially if it has not been taken before, is of key importance during the exploration process [5], [6]. Using this assessment to ensure safety is known as safe learning.

Safe learning methods generally incorporate an approximate initial guess for the system dynamics with some bounds on the modelling error incurred in the approximation [7], [8]. A learning term then refines the initial guess over time using experience data to better approximate the true dynamics. The goal is to guarantee that the system does not violate safety constraints (e.g., limits on the control input or path tracking error) while achieving the control objective (e.g., following a path) and while at the same time improving the model of system and, consequently, its task performance over time. Most learning algorithms learn a single model for system dynamics or use multiple models that are trained ahead of time based on appropriate training data from operating the robot in all relevant conditions. This presents a challenge for robots that are deployed into a wide range of operating conditions which may not all be known ahead of time.

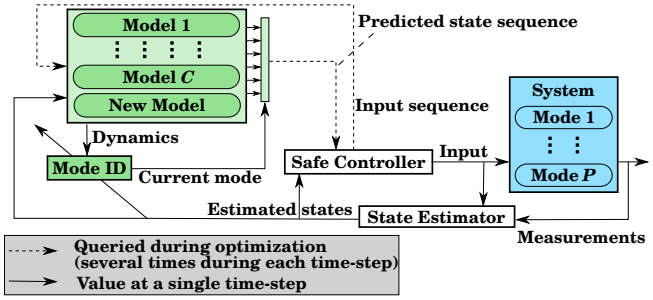

Fig. 1. Block diagram showing the proposed multimodal model learning in closed loop with a safe controller. The robotic system dynamics consist of P distinct modes depending on the operating conditions (blue). Our algorithm (green) learns a multimodal model for the system dynamics, and selects the correct model based on recent measurements at run time. The diagonal arrow indicates that a recent history of data is used. The proposed model learning scheme is designed for a safe controller such as the one presented in [7].

This paper presents a method to model the dynamics of robotic systems where the dynamics may be subject to large changes that depend on discrete latent variables (see Fig. 1). These latent variables reflects a discrete set of operating conditions or physical configurations (dynamic modes) of the robot that change the dynamics. Examples are weather or terrain conditions, or payload configurations. The proposed method uses a Dirichlet Process (DP) to represent the dynamic modes of the system in a way that does not require the number of modes to be specified ahead of time, and Gaussian Process (GP) experts to learn the dynamics of the robot in each mode while making only mild, prior assumptions on the robots dynamics in each mode. The GP experts naturally express the uncertainty in the dynamics in regions of the state-action space depending on whether they have been visited before. The DP allows the model to return to a safe mode when the current set of models does not explain current measurements until a new model is established. The result is a mixture model that learns a new model when the current set of models is insufficient to explain current measurements

The authors are with the Dynamic Systems Lab (www.dynsyslab.org) at the University of Toronto Institute for Aerospace Studies (UTIAS), Canada. Email: chris.mckinnon@mail.utoronto.ca, schoellig@utias.utoronto.ca

This work was supported in part by the Natural Sciences and Engineering Research Council of Canada under the grant RGPIN-2014-04634 and by the Connaught New Researcher Award.

and returns to an existing model when possible. We do not 'forget' previous experiences in different modes when learning new ones, which is a significant advantage over previous methods.

II. RELATED WORK

Learning control has received a great amount of attention in recent years, most notably in the case of single-mode learning control. This is the broad class of learning methods that assumes the true (but initially unknown) mapping between the state, $\mathbf{x}_k$, and the input $\mathbf{u}_k$, and the next state, $\mathbf{x}_{k+1}$, is one-to-one, or at least normally distributed according to some underlying process, $\mathbf{x}_{k+1} \sim \mathcal{N}(f(\mathbf{x}_k, \mathbf{u}_k), \boldsymbol{\sigma})$. Recent developments have contributed safety guarantees [8] and demonstrated impressive results in improved path following [9].

Multimodal safe control and path planning has also received a growing amount of attention. Applications include safely gliding a parafoil under a variety of wind conditions [10] and planning safe paths among uncertain agents such as pedestrians or automobiles [11]. The assumption and challenge in these cases is that the environment or obstacles in the environment have hidden states that cannot be directly measured. Similar to our method, the algorithms try to infer this hidden state based on available observations and use this for safe planning. An additional feature of our approach is that we attempt to build this model online.

Recent results in single-mode, safe learning control have taken great steps to improve performance while maintaining bounds on modelling error and therefore safety. Approaches by [7], [9], [12]–[14] use GPs as corrective terms for approximate prior models and update them over time as more experience is gathered.

One model that exhibits especially good real-time performance and has been demonstrated in several real-world examples is presented in [7]. This approach continually reconstructs the GP disturbance model based on a fixed number of data points, to ensure the process model can be evaluated in constant time even if new experience is added. Storing the data in first-in-first-out bins of fixed size allows the algorithm to update the data used in the GP in real time. If the mode changes, the model un-learns the existing mode by over-writing all of that data and re-learns the new mode. During this process, it suffers from the same problems related to hyper-parameters as mentioned above including either requiring over-conservative bounds to accommodate multiple modes, or have bounds that are realistic for a single-mode, but are unsafe while the model transitions between modes and is using data from more than one mode. Our method aims to overcome these limitations by learning a separate model for each distinct mode.

In addition to the single-mode, safe learning controllers, multimodal algorithms exist which identify a number of dynamic modes ahead of time using labelled or unlabelled training data and switch to the most likely model during operation [10], [11], [15], [16]. This allows them to maintain persistent knowledge of a robots dynamics across a wide range of operating conditions. Inferring the correct mode from measurements during operation allows them to maintain a high level of performance and robustness even when the mode is not directly observed. The method proposed in [17] for linear systems even infers the number of modes at training time. These approaches do, however, require that the number of modes and/or training data from each mode be available ahead of time, which can be a challenging task in robotics. In contrast, our method does not require the number of modes or training data from each mode to be available ahead of time. Rather, it learns new modes as they arise during operation.

One method that does allow multimodal models to be learned during operation is [18] which learns an infinite mixture of linear experts. This mixture expands as the system experiences new dynamics. However, linear experts can only represent nonlinear dynamics locally and therefore require multiple mode switches over a larger region of the state space even if the true underlying mode does not change. We aim to use GPs for experts which can represent nonlinear dynamics globally and therefore only require a mode switch when the underlying mode changes if the modes are learned correctly. This is an advantage for predictive controllers which rely on predicting the robots dynamics far from the current state.

The proposed work is based on combining GPs and the Dirichlet Process (DP), which is used in Bayesian non-parametric clustering models. GPs have been combined with DPs before to obtain a powerful regression tool [19]. For their GP mixture, the posterior is the distribution corresponding to every possible assignment of data points to experts; therefore the likelihood is a sum over (exponentially many) assignments which must be evaluated by sampling. This was an offline method and not designed to be tractable in real-time for a robotics application. In our approach we assume that points come from only one mode at a time so each point belongs to only one expert. Inferring the mode allows us to assign an experience to a single GP which avoids the computationally expensive sampling. The computational cost of our method scales linearly with the number of experts compared to a single GP.

In light of the current approaches and their limitations, the goal of this paper is to present a method for adapting to multiple dynamic modes with guarantees on safety using a realistic and computationally efficient representation of the system dynamics (including predictive uncertainty). The aim is to design a learning algorithm that, while initially equipped for "everywhere mediocrity", learns the specific set of skills necessary to achieve excellence in the relevant operating conditions.

III. PROBLEM STATEMENT

The goal of this work is to learn a dynamic model from data that can predict the future states for a non-linear, switching dynamic system where the number of modes and dynamics in each mode are not known ahead of time. The algorithm should learn new models when new modes are encountered, and improve existing models when modes

are re-visited. The model should also include a reasonable estimate of model uncertainty that acts as an upper bound on model error at all times.

Further assumptions can be summarized as follows:

- The mapping $(\mathbf{u}_k, \mathbf{x}_k) \rightarrow \mathbf{x}_{k+1}$ is one-to-one for a given mode.
- The mode is constant over a short time horizon.
- The number of modes and the mapping $(\mathbf{u}_k, \mathbf{x}_k) \rightarrow \mathbf{x}_{k+1}$ for each mode is not known ahead of time.

A short time horizon could be similar to the horizon considered for Model Predictive Control (MPC).

The system can be modelled by some nominal dynamics, $\mathbf{f}(\mathbf{x}_k, \mathbf{u}_k)$, with additive, initially unknown dynamics, $\mathbf{g}^c(\mathbf{a}_k)$, that are specific to a mode, c, and depend on features $\mathbf{a}_k$:

$$\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k) + \mathbf{g}^c(\mathbf{a}_k). \tag{1}$$

The unknown dynamics are assumed to be a deterministic function with additive, zero-mean, Gaussian noise,

$$\mathbf{g}^c(\mathbf{a}_k) = \mathbf{g}_0^c(\mathbf{a}_k) + \boldsymbol{\eta}^c, \tag{2}$$

where $\boldsymbol{\eta}^c \sim \mathcal{N}(0, \boldsymbol{\Sigma}_n^c)$, and $\boldsymbol{\Sigma}_n^c$ is the measurement noise covariance.

IV. METHODOLOGY

In this section, we present our approach to modelling systems with multimodal dynamics using a combination of GPs and DPs.

A. Dirichlet-Gaussian Process Mixture Model (DPGPMM)

The goal is to learn the unknown dynamics, $g^c(\mathbf{a})$, for each dynamic mode from data, and automatically detect the relevant mode or create a new model if necessary. To do this, we propose is a Dirichlet Process Gaussian Process Mixture Model (DPGPMM). The DP is used to learn the number of dynamic modes and the GP is used to model the error between the dynamics of the real system and the prior model in each mode. There are four key properties of the model that make it ideally suited for safe learning the dynamics of multimodal systems:

1) The ability to handle an increasing number modes (DP);
2) The definition of a 'safe mode' when no data is available or there is a poor fit (DP & GP);
3) The quantitative bounds for the model error (GP); and
4) The possibility to improve the model over time (GP).

B. Gaussian Process (GP) Disturbance Model

We model the disturbance, $\mathbf{g}(\cdot)$, as a GP based on past observations. We drop the $(\cdot)^c$ for notational convenience, as we learn a GP for each mode separately. Since there are many good references on GPs [20], here we provide only a high-level sketch. The learned model depends on previously gathered experiences which are assembled from measurements of the state denoted by $\hat{\mathbf{x}}$ and the $\mathbf{u}$ using (1), so that

$$\hat{\mathbf{g}}(\mathbf{a}_{k-1}) = \hat{\mathbf{x}}_k - \mathbf{f}(\hat{\mathbf{x}}_{k-1}, \mathbf{u}_{k-1}). \tag{3}$$

The resulting pair, $\{\mathbf{a}_{k-1}, \hat{\mathbf{g}}(\mathbf{a}_{k-1})\}$, forms an individual experience. For simplicity, we model each dimension of the disturbance using a separate GP. Below we derive the equations for a single dimension of $\mathbf{g}(\cdot)$ denoted by $g(\cdot)$.

A GP is a distribution over functions given past experiences, $\mathcal{D}_n = \{\hat{g}(\mathbf{a}_i), \mathbf{a}_i\}_{i=1}^n$, and kernel hyper-parameters.

We assume the experiences are noisy observations of the true function $g(\mathbf{a}_k)$; this is, $\hat{g}(\mathbf{a}_k) = g(\mathbf{a}_k) + \eta_\eta$ where $\eta_\eta \sim \mathcal{N}(0, \sigma_\eta^2)$. The posterior distribution is characterized by a mean and variance which can be queried at any point $\mathbf{a}_*$ using

$$\mu_n(\mathbf{a}_*) = \mathbf{k}_n(\mathbf{a}_*)\mathbf{K}_n^{-1}\hat{\mathbf{g}}_n, \tag{4}$$

$$\sigma_n^2(\mathbf{a}_*) = \kappa(\mathbf{a}_*, \mathbf{a}_*) - \mathbf{k}_n(\mathbf{a}_*)\mathbf{K}_n^{-1}\mathbf{k}_n(\mathbf{a}_*)^T, \tag{5}$$

where $\hat{\mathbf{g}}_n = [\hat{g}(\mathbf{a}_1), ..., \hat{g}(\mathbf{a}_n)]^T$ is the vector of observed function values, the covariance matrix $\mathbf{K}_n \in \mathbb{R}^{n \times n}$ has entries $[\mathbf{K}_n(\mathbf{a}_i, \mathbf{a}_j)] = \kappa(\mathbf{a}_i, \mathbf{a}_j) + \sigma_\eta^2 \delta_{ij}$, where δ_{ij} is the Kronecker delta, and the vector $\mathbf{k}_n(\mathbf{a}_*) = [\kappa(\mathbf{a}_*, \mathbf{a}_1), ..., \kappa(\mathbf{a}_*, \mathbf{a}_n)]$ contains the covariances between the new test point $\mathbf{a}_*$ and the observed data points $\mathcal{D}_n$. For this work, we use the squared exponential kernel,

$$\kappa(\mathbf{a}_i, \mathbf{a}_j) = \sigma_f^2 \exp\left(-\frac{1}{2}(\mathbf{a}_i - \mathbf{a}_j)^T \mathbf{L}^{-2}(\mathbf{a}_i - \mathbf{a}_j)\right) \tag{6}$$

because of its success in modelling robot dynamics [2], [6], [7], [9], [13]. The hyper-parameters are the diagonal matrix, $\mathbf{L}$, of length-scales which are inversely related to the importance of each element of $\mathbf{a}$, and the process noise variance, σ_f^2, which is the variance of the prior family of functions represented by $g(\cdot)$.

As training data is added to a particular GP, uncertainty is reduced and the posterior distribution of the GP specializes to a particular family of functions which represents the system dynamics in a particular mode. The DP is then used as a distribution over these families of functions where each family, represented by a GP, is the system dynamics in a particular mode.

C. Dirichlet Process (DP)

For the DPGPMM, the DP acts as a distribution over modes, which assumes the number of modes is infinite. In reality, however, only a small number of modes will actually have data. Suppose there are C modes and let $\mathbf{c} = (1, .., C)$ be the vector of indicator variables for the modes. The conditional probability of each mode c when integrating over all possible modes is then

$$p(c = j|\mathbf{c}, \alpha) = \frac{n_j}{N - 1 + \alpha} \quad \text{for existing modes} \tag{7}$$

$$p(c = C + 1|\mathbf{c}, \alpha) = \frac{\alpha}{N - 1 + \alpha} \quad \text{for new modes} \tag{8}$$

where $j \in \{1, ..., C\}$ and n_j is the number of points in expert j, N is the total number of data points in all the experts, and α is a parameter of the DP called the concentration parameter, which controls the prior probability of new modes [19]. We have used $\mathbf{c}$ as a shorthand to indicate the existing mixture model, which includes the GP associated

with each mode and hence the number of points in that GP.

The important properties of the DP are that modes with more experiences are more likely, and the model always includes an element for a new mode, which is the GP with no data, or the GP prior. The GP prior, $g^*(\mathbf{a}) \sim \mathcal{N}(0, \sigma_f^2)$, has the largest variance (σ_f^2 using (5)), which results in the most conservative bounds on the disturbance and thus acts as a 'safe' mode.

D. Mode Inference

During deployment, the goal is to find the best estimate of the current and future model error given all past experiences. We use a recent history of p experiences, $\mathcal{D}^- = \{\hat{g}(\mathbf{a}_i), \mathbf{a}_i\}_{i=k-p}^{k}$, to infer the mode at the current time-step, k, and use the most likely mode to predict the model error at future time-steps. We assume the mode is constant over short time periods; so all samples in $\mathcal{D}^-$ should come from the same mode. The posterior probability of the jth mode is

$$p(c = j|\mathcal{D}^-, \mathbf{c}) \propto p(\mathcal{D}^-|c = j)p(c = j|\mathbf{c}, \alpha). \quad (9)$$

where $p(c = j|\mathbf{c}, \alpha)$ is the prior probability of mode j calculated using (7) or (8), and $p(\mathcal{D}^-|c = j)$ is the probability of recent experiences under mode $c = j$. The probability of recent measurements under mode j is

$$p(\mathcal{D}^-|c = j) = \prod_{i=(k-p)}^{k} p(\hat{g}(\mathbf{a}_i)|\mu_n^{c_j}(\mathbf{a}_i), \sigma_n^{c_j}(\mathbf{a}_i)), \quad (10)$$

where $\mu_n^{c_j}$ and $\left(\sigma_n^{c_j}\right)^2$ are the mean and variance of the GP representing mode j. A new cluster is created when the probability of $c = C + 1$ is larger than any expert given recent data. Since it is computationally expensive to create a new GP model, which involves inverting an $n_j \times n_j$ matrix, we remain in the safe mode until the model returns to an existing mode. While in the existing mode, the system uses experience gathered while it was in the safe mode to create a new GP model. This approach assumes that the system does not transition between two unknown modes before transitioning back to an existing mode. This was inspired by [21] which is about managing experiences for visual navigation in visually changing outdoor environments.

V. GROUND ROBOT MODEL

We demonstrate our multimodal learning approach in experiment on a ground robot, namely the Clearpath Husky (see Fig. 2). This section outlines the choice of *a-priori* model.

We use a similar a-priori model to [9], which uses a unicycle model, but we include a term for translation along the body y-axis and use first-order dynamics with a time delay for $\dot{x}^b$ and $\dot{\theta}$. The translational velocity expressed in the body frame is $(\dot{x}_k^b, \dot{y}_k^b)$ and the rotational velocity is $\dot{\theta}_k$. The addition of the y-component in velocity is to include a learning term to account primarily for offsets between the pivot point and the origin of the body frame. The proposed model is

$$\begin{bmatrix} x_{k+1} \\ y_{k+1} \\ \theta_{k+1} \end{bmatrix} = \begin{bmatrix} x_k \\ y_k \\ \theta_k \end{bmatrix} + \Delta t \begin{bmatrix} \cos\theta_k & -\sin\theta_k & 0 \\ \sin\theta_k & \cos\theta_k & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{x}_k^b \\ \dot{y}_k^b \\ \dot{\theta}_k \end{bmatrix} \quad (11)$$

where

$$\begin{bmatrix} \dot{x}_{k+1}^b \\ \dot{y}_{k+1}^b \\ \dot{\theta}_{k+1} \end{bmatrix} = \mathbf{h}(c_k, \cdot) \quad (12)$$

where $\mathbf{h}(c_k, \cdot)$ is the true nonlinear process model for mode c at time-step k that we wish to approximate.

Experiments have shown that the translational and rotational dynamics can be reasonably well approximated by a first order system with time-delay, so we use a first order prior model and a learning term to model the dynamics of the real system,

$$\mathbf{h}(c_k, \cdot) \approx \begin{bmatrix} \dot{x}_k^b + \Delta t \left(\frac{1}{T_1}(u_{k-d} - \dot{x}_k^b)\right) \\ 0 \\ \dot{\theta}_k + \Delta t \left(\frac{1}{T_2}(w_{k-d} - \dot{\theta}_k)\right) \end{bmatrix} + \begin{bmatrix} g_u^c(\mathbf{a_k}) \\ g_v^c(\mathbf{a_k}) \\ g_w^c(\mathbf{a_k})) \end{bmatrix} \quad (13)$$

where T_1 and T_2 are the time constants for the translational and rotational dynamics, d is the number of time-steps of the delay, and $\mathbf{a}$ is the disturbance dependency, which will be defined in Sec. VI-B.

VI. EXPERIMENTS

A. Experimental Setup

Experiments were conducted on a 50 kg Clearpath Husky skid-steer ground robot shown in Fig. 2. Weights and tyre pressure were varied to produce five dynamic modes while the vehicle was driven manually on a polished concrete surface. The *no mass configuration* was with no mass and high tyre pressure. The *centred configuration* was with a 25 lb weight on the front bumper and a 35 lb weight on the rear bumper. The *offset configuration* was tested with a 35 lb weight on the rear bumper and a 25 lb weight on an arm extended beyond the body (depicted in Fig. 2). The centred and offset configurations were tested with *high and low tyre pressure*. Four trials were conducted in each configuration, where the vehicle was driven manually for a total of 1347 s. The motion of the vehicle was captured using a Vicon motion capture system at 200 Hz, and commands were sent to the vehicle at 10 Hz. Below, we focus on the rotational dynamics since the added mass did not have any significant effect on translation.

B. Experiences

The learned model depends on observations of prediction error, $\hat{g}(\mathbf{a}_k)$, from (3), gathered during previous trials. Experiences are calculated using the a-priori model (12), (13), and measurements of the translational and angular velocity in the body frame, $(\dot{x}_k^b, \dot{y}_k^b, \dot{\theta}_k)$. We down-sample measurements from the motion capture system taking the most recent pose

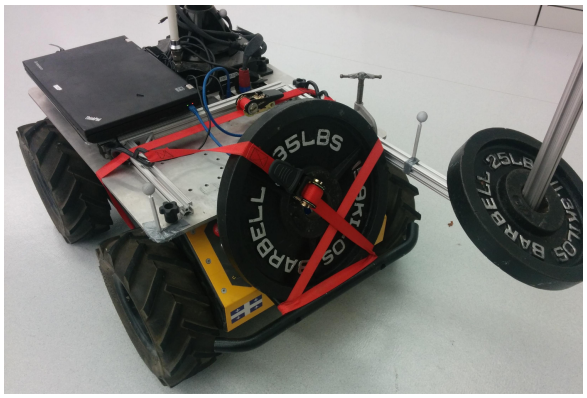

Fig. 2. The Clearpath Husky robot with additional weights used for experiments. The configuration shown changes the mass and pivot point of the vehicle which drastically changes its rotational dynamics. Different loading configurations result in different dynamics. Our approach detects which dynamic mode is currently active and learns a new model when the existing library of dynamic models is insufficient to explain current measurements.

measurement for each time-step where a new command was applied.

Choosing the correct dependence for the disturbance is an important and challenging design task. In our experiments, $\mathbf{a} = (\dot{x}_k^b, \dot{y}_k^b, \dot{\theta}_k^b, w_k, w_{k-1}, w_{k-2})$. This choice for $\mathbf{a}$ assumes that all disturbances act in the body frame. This assumption was made based on the intuition that disturbances come from varied interaction between the wheels and the surface; since the wheels are fixed in the body frame, the unmodelled dynamics should be as well. Including several inputs was motivated by an obvious time delay in the vehicles behaviour of two to three time-steps. Using this approach, the model can learn systematic time delays in the dynamics in the range from one to three time-steps.

C. Tuning Parameters

Hyper-parameters for the GP were trained using data from the configuration with no mass and fixed thereafter. For this work, we used version 1.0.9 of the GPy package [22]. The maximum number of points in a GP was fixed to 1000. The concentration parameter of the DP, α, was set to 1. This means that with no other indicative measurements, an existing mode is far more likely than a new mode. The prior function variance, $\sigma_f^2 = 0.25^2$, was chosen as an upper bound on the expected disturbances. The noise variance, $\sigma_\eta^2 = 0.05^2$, was chosen as the upper bound on measurement noise, and the kernel length-scales, with diagonal elements $diag(\mathbf{L}) = (1.48, 685, 0.18, 800, 0.69, 0.64)$ were optimized using training data collected with no mass on the Husky. The large length-scales for $\dot{y}_k^b$ and w_k indicates that the GP has learned these elements are not important.

D. Mode Inference Given Fixed Models

First, we demonstrate the mode inference given a fixed number of models trained using known modes. Four runs of about 90 s were conducted in each mode and one was used for training. The confusion matrix [23] in Table I shows that classification errors occur primarily between the modes in the first three columns, where the center of mass of the Husky is roughly over the center of the wheels. This suggests that these modes have similar dynamics which matches our observations and means that experience from one of these modes may be relevant to another. In addition, a classification error in these cases would be of little consequence since the predictions made using any of these models will be similar.

	centred	centred, low	no mass	offset	offset, low	safe
centred	**0.49**	**0.06**	**0.34**	0.02	0.03	0.06
centred, low	**0.21**	**0.43**	**0.27**	0.00	0.01	0.08
no mass	**0.42**	**0.20**	**0.26**	0.01	0.01	0.10
offset	0.01	0.01	0.01	**0.88**	0.02	0.07
offset, low	0.00	0.02	0.04	0.08	**0.78**	0.08

TABLE I

THE CONFUSION MATRIX FOR MODE CLASSIFICATION BASED ON MODELS TRAINED USING THE TRUE LABELS. COLOURED BOXES INDICATE MODES WITH SIMILAR DYNAMICS.

E. Model Learning Example

Figure 3 demonstrates how the algorithm learns new models safely and efficiently when confronted with novel configurations. The model was initialized with data from the *no mass* configuration; that is, initially it has no experience related to the *offset* configuration. The mode identification used the most recent 2 s of data to identify the current mode. When it started moving at 9 s and encountered measurements from the *offset* configuration, it reverted to the safe mode. During this time, the learning term, $g_w^c(\mathbf{a_k})$, is approximately a zero-mean Gaussian with a large variance, which keeps the measurements within 3σ (shaded in red) of the mean (red line). At 18 s, it switches to the learned mode and constantly detects the new mode except when the vehicle is stationary again at 45 s. At this point, all modes predict that vehicle remains stationary so the DP is dominant and the mode with the largest number of points, the initial model trained for no mass, is selected. The model remains in this configuration until 141 s, since the *no mass* configuration is very similar to the *centred* configuration. At 141 s when the Husky is commanded to rotate again, the model switches back to the previously learned model for *offset*, leveraging previous experiences. This demonstrates how the proposed approach safely learns a new model and makes efficient use of experiences by constantly improving existing models.

F. Model Prediction Performance

The method presented in this paper is aimed at improving performance and safety of safe learning controllers such as the one presented in [7]. This safe controller relies on an accurate prediction of the mean state and modelling error bounds over a short time horizon given the current state and a series of inputs, see Fig. 1. In our experiments, the Husky ground robot was manually driven. This gives us a series of states and inputs. To measure the accuracy of the prediction of the mean state by our multimodal model, we use Root Mean Square Error (RMSE) between the prediction made using our model given a state and a series of subsequent

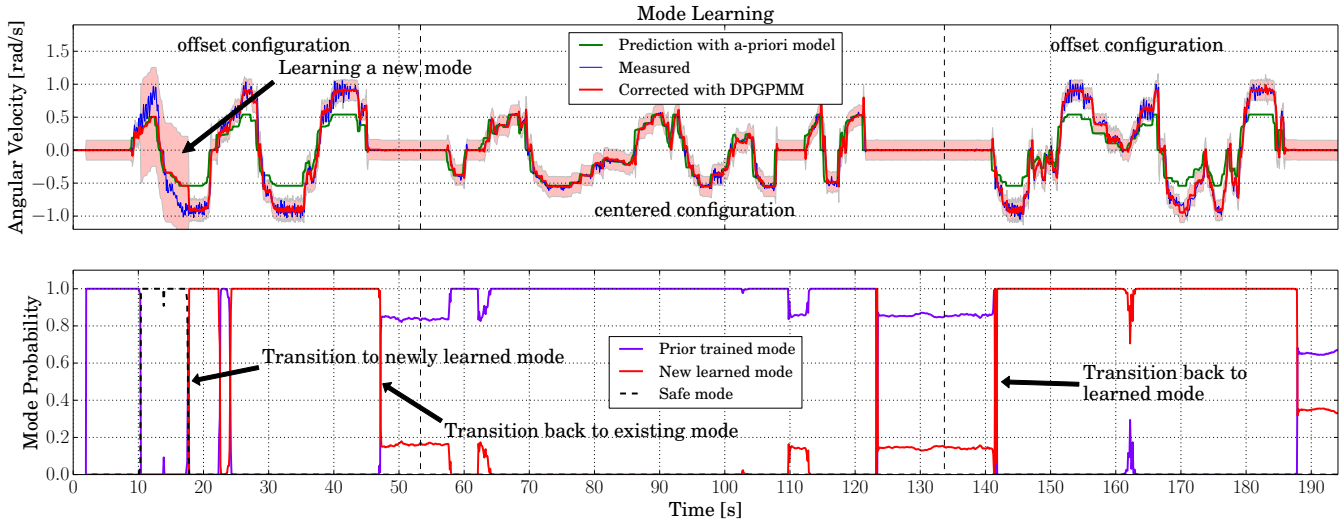

Fig. 3. During this experiment, the Husky was deployed in three different configurations. First, the offset configuration (0-53 s), then the centred configuration (53-134 s), then the offset configuration again (134 s-194 s). The DPGPMM was initialized with a GP pre-trained using data in the no mass configuration. From 10-18 s, the model cannot explain experiences from the offset configuration, so it is in safe mode, which results in a large uncertainty. At 18 s, the model makes a classification error and switches to its initial mode which triggers the creation of a new learned mode. Immediately after, it switches to the new mode and continues improving this mode until 47 s where it switches back to the prior mode. At this time, the vehicle is stationary, so both models explain the motion well but the initial mode contains more points so it switches to this mode. At 141 s, the DPGPMM switches back to the newly learned mode since it has already learned a model for this configuration, so does not go into safe mode first.

inputs, and the measured state at these subsequent time-steps. To measure the accuracy of the error bound prediction, we will use the RMS Z-score (RMSZ) of the prediction at the future time-steps.

The RMSZ for a short window of p time-steps is defined as

$$RMSZ_k = \sqrt{\frac{1}{p}\sum_{i=k+1}^{k+p} \frac{(\dot{\theta}_i^{true} - \mu_{\dot{\theta}}^{c_k}(\mathbf{a}_i))^2}{\sigma_{\dot{\theta}}^{c_k}(\mathbf{a}_i)^2}} \qquad (14)$$

where $\mu_{\dot{\theta}}^{c_k}(\mathbf{a}_i)$ and $\sigma_{\dot{\theta}}^{c_k}(\mathbf{a}_i)$ are the mean and standard deviation of the GP of mode c_k evaluated at $\mathbf{a}_i$. The true measurements of angular velocity, $\dot{\theta}_i^{true}$, are used for comparison. The most likely model at time-step k, c_k, is chosen based on the p most recent points using (9). Values smaller than one indicate the estimate is overly conservative, values close to one indicate a perfect estimate, and values larger than one indicate the method is over-confident.

Figure 4 shows a comparison of the RMS error over 16 trials using three different approaches. First, we use a single GP initialized with data from the no mass configuration. Second, we train a supervised mixture of GPs using perfectly labelled data from all configurations (one for no mass, one for centred, one for offset, and one for offset with low tyre pressure) and manually label which mode is active at run time. This model also acts as a measure of the limit of accuracy of the GP with fixed hyper-parameters and perfect mode inference. Third, the proposed DPGPMM is initialized with one GP trained using data from the no mass configuration and learns the remaining modes during the experiment. The algorithm for updating GPs to maintain a constant size is based on continually choosing a random Subset Of Data (SOD) associated with the model. This was chosen for simplicity and good performance relative to other SOD methods [20].

The results in Fig. 4 show how the proposed method quickly approaches the performance of the mixture model trained with perfectly labelled data, and retains this performance regardless of mode switches. The GP mixture trained using labelled data represents a baseline for 'good' performance using a GP model with fixed hyper-parameters.

The single GP can learn to approximate one set of dynamics after a long period of time, but must un-learn and re-learn dynamics each time it encounters a new mode resulting in consistently higher RMSE and RMSZ.

Using a supervised GP mixture model in practice is difficult because it requires perfect training data to be available ahead of time. This is not always possible. Moreover, such a model does not adapt to slight changes between modes (e.g., caused by tyre pressure) while the proposed method does.

VII. DISCUSSION

Choosing a good a-priori model, (11)-(13) with $g_*^c(a_k) = 0$, and hyper-parameters are essential for the success of any method. For the purpose of this paper, a good prior model should result in model errors, $\hat{g}(\mathbf{a})$ (3), that can be well approximated by a GP. For the squared exponential kernel, this means the error should be smooth, which is why we chose a first-order prior model as opposed to an instantaneous prior as in [9]. The length-scale hyper-parameters then dictate how smooth the function is expected to be. Since we assumed that training data would not be available from all modes, we chose hyper-parameters based on a configuration with no mass on the Husky. Since the Husky rotates at almost

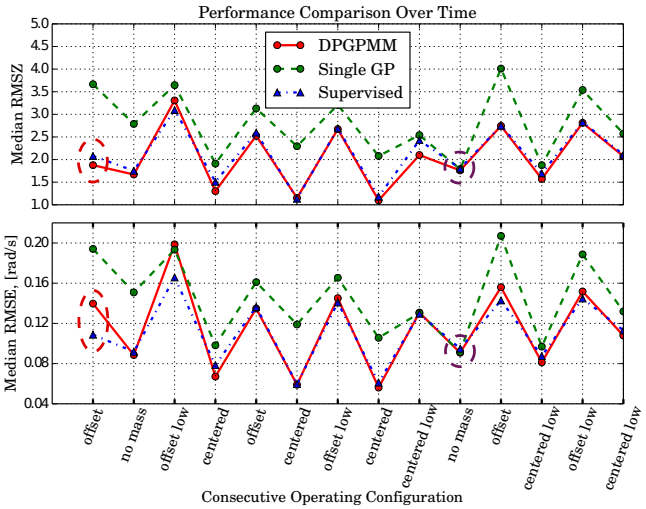

Fig. 4. This plot shows the median of the RMS error and weighted RMS error for an entire runof consecutively changing configurations. For comparison, we show the proposed method (red), a single GP that is initialized with data from the no mass configuration (green), and an ideal mixture model with static GPs trained on labelled training data for each mode (blue). The red circle shows where the proposed method was initially uncertain about the dynamics of the robot in the offset configuration which resulted in higher uncertainty which reduces the RMSZ despite having a higher RMSE. This is a good indication of safe performance. For runs 8-10, the Husky was in either a centred or no mass configuration. After three repeated runs with the same configuration, the single GP adapts to the dynamics in this mode. On subsequent runs, however, it has unlearned the dynamics in the offset configuration which results in dramatically increased RMSE and RMSZ while the proposed method remains close to the ideal model.

twice the rate in an offset configuration, these length scales were no longer as accurate. This results in higher error while the vehicle is in the offset configuration. This could be addressed by separately optimizing the hyper-parameters for each model when enough data is gathered; however, that was beyond the scope of this work.

VIII. CONCLUSIONS

This paper has presented a method using Gaussian Processes as experts in a Dirichlet Process mixture model to learn an increasing number of non-linear models for robot dynamics that are affected by different, discrete operating conditions. We have demonstrated in experiment how this approach stores and re-uses past experience from a robot's deployment in an arbitrary number of previous operating conditions, and automatically learns a new model when a new and distinct operating condition is encountered. The proposed method demonstrates significant improvements over single GP models, and approaches the performance of a supervised method. We hope the reader will find this an interesting option for safe control of multimodal systems where the dynamics in each mode cannot be specified ahead of time.

REFERENCES

[1] J. Kober, J. Bagnell, and J. Peters. Reinforcement Learning in Robotics: A Survey. *Intl. Journal of Robotics Research (IJRR)*, 32(11):1238–1274, 2013.

[2] F. Berkenkamp and A. P. Schoellig. Safe and Robust Learning Control with Gaussian Processes. In *Proc. of the European Control Conference (ECC)*, pages 2501–2506, 2015.

[3] T. Moldovan, S. Levine, M. Jordan, and P. Abbeel. Optimism-Driven Exploration for Nonlinear Systems. In *Proc. of the Intl. Conf. on Robotics and Automation (ICRA)*, pages 3239–3246, 2015.

[4] C. Xie, S. Patil, T. Moldovan, S. Levine, and P. Abbeel. Model-based Reinforcement Learning with Parametrized Physical Models and Optimism-Driven Exploration. In *Proc. of the Intl. Conf. on Robotics and Automation (ICRA)*, pages 504–511, 2016.

[5] F. Berkenkamp, A. P. Schoellig, and A. Krause. Safe Controller Optimization for Quadrotors with Gaussian Processes. In *Proc. of the Intl. Conference on Intelligent Robots and Systems (IROS)*, pages 491–496, 2016.

[6] C. Ostafew, A. P. Schoellig, and T. Barfoot. Conservative to Confident: Treating Uncertainty Robustly Within Learning-Based Control. In *Proc of the Intl. Conf. on Robotics and Automation (ICRA)*, pages 421–427, 2015.

[7] C. Ostafew, A. P. Schoellig, and T. Barfoot. Robust Constrained Learning-based NMPC Enabling Reliable Mobile Robot Path Tracking. *Intl. Journal of Robotics Research (IJRR)*, 35(13):1547–1563, 2016.

[8] A. Aswani, H. Gonzalez, S. Sastry, and C. Tomlin. Provably Safe and Robust Learning-based Model Predictive Control. *Automatica*, 49(5):1216–1226, 2013.

[9] C. Ostafew, A. P. Schoellig, and T. Barfoot. Learning-based Nonlinear Model Predictive Control to Improve Vision-Based Mobile Robot Path-tracking in Challenging Outdoor Environments. In *Proc. of the Intl. Conf. on Robotics and Automation (ICRA)*, pages 4029–4036, 2014.

[10] B. Luders, I. Sugel, and J. How. Robust Trajectory Planning for Autonomous Parafoils Under Wind Uncertainty. In *Proc. of the AIAA Conference on Guidance, Navigation and Control*, 2013.

[11] G. Aoude, B. Luders, J. Joseph, N. Roy, and J. How. Probabilistically Safe Motion Planning to Avoid Dynamic Obstacles with Uncertain Motion Patterns. *Autonomous Robots*, 35(1):51–76, 2013.

[12] J. Gillula and C. Tomlin. Reducing Conservativeness in Safety Guarantees by Learning Disturbances Online: Iterated Guaranteed Safe Online Learning. In *Proc. of Robotics: Science and Systems (RSS)*, pages 81–88, 2012.

[13] P. Bouffard, A. Aswani, and C. Tomlin. Learning-based Model Predictive Control on a Quadrotor: Onboard Implementation and Experimental Results. In *Proc. of the Intl. Conference on Robotics and Automation (ICRA)*, pages 279–284, 2012.

[14] J. Mahler, S. Krishnan, M. Laskey, S. Sen, A. Murali, B. Kehoe, S. Patil, J. Wang, M. Franklin, P. Abbeel, et al. Learning Accurate Kinematic Control of Cable-driven Surgical Robots using Data Cleaning and Gaussian Process Regression. In *Proc. of the Intl. Conference on Automation Science and Engineering (CASE)*, pages 532–539, 2014.

[15] K. Jo, K. Chu, and M. Sunwoo. Interacting Multiple Model Filter-based Sensor Fusion of GPS with In-vehicle Sensors for Real-time Vehicle Positioning. *Transactions on Intelligent Transportation Systems*, 13(1):329–343, 2012.

[16] R. Calandra, S. Ivaldi, M. Deisenroth, E. Rueckert, and J. Peters. Learning Inverse Dynamics Models with Contacts. In *Intl. Conf. on Robotics and Automation (ICRA)*, pages 3186–3191, 2015.

[17] E. Fox, E. Sudderth, M. Jordan, and A. Willsky. Nonparametric Bayesian Learning of Switching Linear Dynamical Systems. In *Proc. of Advances in Neural Information Processing Systems (NIPS)*, pages 457–464, 2009.

[18] L. Jamone, B. Damas, and J. Santos-Victor. Incremental Learning of Context-dependent Dynamic Internal Models for Robot Control. In *Intl. Symp. on Intelligent Control (ISIC)*, pages 1336–1341, 2014.

[19] C. Rasmussen and Z. Ghahramani. Infinite Mixtures of Gaussian Process Experts. In *Proc. of Advances in Neural Information Processing Systems (NIPS)*, volume 2, pages 881–888, 2002.

[20] C. Rasmussen and C. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006.

[21] C. Linegar, W. Churchill, and P. Newman. Work Smart, Not Hard: Recalling Relevant Experiences for Vast-Scale but Time-Constrained Localisation. In *Proc. of the Intl. Conference on Robotics and Automation (ICRA)*, pages 4137–4143, 2015.

[22] GPy. GPy: A Gaussian Process Framework in Python. `http://github.com/SheffieldML/GPy`, since 2012.

[23] K. Murphy. *Machine Learning: A Probabilistic Perspective*. MIT Press, 2012.