Learning-Based Hybrid Model Predictive Control for Systems With Partially Known and Evolving Dynamics

Leroy D'Souza, Yash Vardhan Pant, Sebastian Fischmeister

University of Waterloo, Waterloo, ON, Canada

{l8dsouza, yash.pant, sfischme}@uwaterloo.ca

Abstract

Safe operation of autonomous system in environments where system dynamics are partially unmodelled and potentially time-varying remains an open challenge. We present an approach that works alongside a data-driven predictive algorithm to control a system subject to such dynamics. The proposed approach bootstraps from prior estimates of these dynamics learnt from previously seen data, while also incorporating likelihood estimates from newly obtained information. We demonstrate an experiment where performance improves by a factor of over 80% on a repetitive control task when these unmodelled dynamics terms are iteratively learned over time.

1 Introduction

A dominant subset of data-driven control research involves learning unmodelled (or *residual*) dynamics, uncaptured by low-fidelity first principles models, to obtain improved control performance (D'Souza and Pant 2023; Hewing, Kabzan, and Zeilinger 2020; Calandra et al. 2015). Recent work has also started to address this problem in light of systems subject to *hybrid* residual dynamics. In this paper, we consider a more realistic setting of the problem where, provided with a hybrid *bayesian* residual model, the system must operate (using a Model Predictive Controller (MPC)) in an environment where the active mode of the hybrid model in effect at different points of the workspace is apriori unknown. Moreover, we consider that this distribution can change over time resulting in a "dynamic" environment.

Motivating example (EX-M) Consider an autonomous robot performing a repetitive task in an outdoor environment consisting of different terrains, viz. mud, snow, ice etc., with each terrain having a corresponding residual dynamics mode in the provided hybrid model. With no apriori information about the terrain (equivalently mode) distribution, the mode of the hybrid model to be used at each timestep in the MPC dynamics prediction is unknown, which can lead to poor tracking performance. This necessitates the use of a mapping algorithm to identify the mode distribution from data.

Furthermore, the terrain distribution can shift over time due to natural causes e.g., additional snowfall or due to robot motion e.g., snow potentially melts when repeatedly travelled on causing the terrain to change to wet mud. As a result, the mapping algorithm alluded to previously must be capable of adapting to samples generated by time-varying terrain distributions in a streaming fashion.

Related Work. There has been work that addresses similar problem statements. Lew et al. (2022) uses online bayesian meta-learning to learn true residual dynamics with guarantees via adaptive confidence sets but does not address the problem of hybrid residual dynamics and mode estimation. Nagy et al. (2023) is capable of dealing with unseen hybrid modes by trying to weight existing modes of the hybrid model. However, it does not build a mode map for better information retention on repetitive tasks and cannot deal with varying modes across the MPC horizon. D'Souza and Pant (2023) constructs a hybrid MPC formulation but assumes full information of the ground truth mode distribution.

Contributions. In this work, we propose an adaptive mode-mapping algorithm that: 1) uses a likelihood-prior trade-off scheme, conditioned on data density estimates, to control rate of adaptation depending on previously seen data, and 2) uses of a density-based sample retainment scheme to prevent information forgetting. We then utilize the output of this algorithm in a hybrid MPC controller (D'Souza and Pant 2023) and show that it improves performance iteratively over a repetitive control task when subject to time-varying dynamics terms.

2 Problem Setup

We first define the dynamics for the system under consideration with x_k, u_k denoting the state of the system and the input applied to it at timestep k respectively.

$$x_{k+1} = f(x_k, u_k) + g_k(x_k, u_k) + w_k \tag{1}$$

$f(x_k, u_k)$ denotes the known dynamics in (1). The residual dynamics, g, are *hybrid* with $\mathcal{M}$ number of modes defined as $g_{\text{set}} = \{g_m \mid m \in \{1, \dots, \mathcal{M}\}\}$. For example, $\mathcal{M} = 3$ in $EX - M$ i.e., one mode for each terrain.

We introduce a set of discrete variables, $\delta_{m,k}$, to select between the elements of g_{set} for the prediction in (1) yielding,

$$g_k(x_k, u_k) = \sum_{m=1}^{\mathcal{M}} \delta_{m,k} g_m(x_k, u_k) \tag{2}$$

Let ice correspond to mode 1 in EX-M. For any point in the

This work was supported in part by Magna International and the NSERC Discovery Grant.

AAAI 2024 Spring Symposium on User-Aligned Assessment of Adaptive AI Systems. Stanford University, Stanford, CA, USA.

workspace, x_k^{ws}, containing ice, $\delta_{1,k} = 1$ and $\delta_{2,k} = \delta_{3,k} = 0$ with (2) reducing to $g_k(x_k, u_k) = g_1(x_k, u_k)$.
The above description applies similarly to the noise term, w_k, with each mode corresponding to different noise parameters that we assume to describe a zero-mean Gaussian.

For our problem, we assume we are given a hybrid Bayesian approximation, $\hat{g}$, of the true residual dynamics in (1). However, we do not know the dynamic function that generates the above described $\delta_{m,k}$'s. We aim to learn an approximation, $\hat{\delta}(x_k^{ws}, m)$, of this function that maps x_k^{ws} to $\delta_{m,k} \, \forall \, m \in \{1, \ldots, \mathcal{M}\}$ from observed data collected during iterative runs of the system in the environment.

3 Methodology

Our approach uses a neural network (Sitzmann et al. 2020) parametrized by weights θ to model $\hat{\delta}$. To learn $\hat{\delta}$, we must compute the likelihood that a particular mode, m, is active at a point x_k^{ws}. From (1), this can be done using mismatches, d_k, between the predicted state, $\hat{x}_{k+1}$ (obtained using $\hat{g}$), and true measured state, x_{k+1}, collected over the course of a trajectory with $d_k = \hat{x}_{k+1} - x_{k+1}$. That is, we obtain the likelihoods $l_{m,k} \propto P(\delta_{m,k} = 1 \mid \hat{g}, \hat{x}_{k+1} - x_{k+1})$.

Since $\hat{\delta}$ is learnt iteratively, it is used to generate prior predictions, $\phi_{m,k} = \phi_m(x_k^{ws})$, as it is implicitly conditioned on previously seen data via the gradient descent training loop.

The prior and likelihood can then be traded off to ultimately obtain posteriors used as training labels, y_k.

$$y_k \propto ({l_{m,k}}^{\alpha_k}) \phi_m(x_k^{ws}) \tag{3}$$

α_k is the trade-off term derived using the density of previously collected samples which can be interpreted as follows,

- A higher α_k boosts trust in the likelihood compared to the prior and can be used to prevent trusting overconfident but inaccurate prior predictions.
- A lower α_k places more trust in the prior and can be used to implicitly specify the rate at which the posterior y_k should adapt to shifts in the mode distribution over time.

This results in a dataset $D_C = (x_k^{ws}, y_k)$ containing samples from the current trajectory. In order to prevent information forgetting, we keep track of previous data samples, D_P, and retain a subset of those samples, D_R, that fall below a density threshold, c_R for a kernel density estimator (Silverman 2018) computed on D_C.

The weights θ are then updated using Adam (Kingma and Ba 2017) on $D_C \cup D_R$ at the end of every run so that $\hat{\delta}$ better estimates the dynamic ground truth function.

Finally we use a hybrid MPC controller in order to control the system given $\hat{\delta}$ and a reference trajectory, x_k^{ref}, to be tracked (D'Souza and Pant 2023).[1]

4 Results

We demonstrate the proposed algorithm on a 2-D quadrotor system (Yuan et al. 2022) with each residual mode of $\hat{g}$ subject to wind dynamics as a function of velocity along the z-axis. Control performance is measured by Cost $= \sum_{k=0}^{T-1}(||x_k - x_k^{ref}||_Q^2 + ||u_k||_R^2)$ with Q and R denoting penalty matrices. The accuracy metric for $\hat{\delta}$ converts soft label vectors to one-hot (as required by the controller) for comparison against the ground truth.

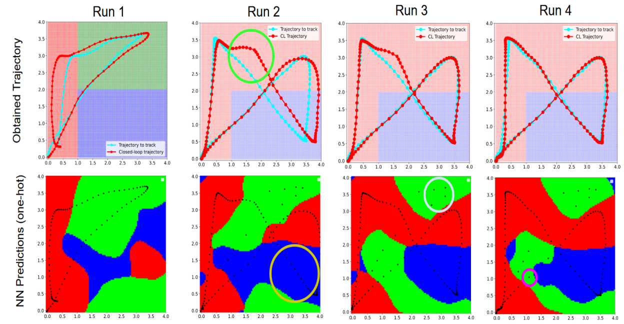

Figure 1: A visualization of predictions from $\hat{\delta}$ over successive runs of a repetitive task. (Top) Ground truth mode distribution + reference, true trajectories. (Bottom) NN predictions after each run's training step. Lack of prior data density in the blue region allows quick convergence of the posterior to obtained likelihood samples (circled in gold) as compared to cautious adaptation to previously green mode changing to red. Change in ground truth mode distribution affects performance (circled in green). Some measured residuals lead to inaccurate label generation (circled in pink). Samples retained in D_R from initial trajectory are circled in grey.

Run #	Avg. Cost (Variance)	Improvement	NN Acc. (%)
2	242.04 (104.62)	-	58.22
3	84.50 (22.74)	65.1%	63.56
4	**42.69** (4.86)	**82.4%**	**67.06**

Table 1: Cost and mapping prediction accuracy (computed on 10K samples) trends over successive iterations of a repetitive task (over 30 simulation runs). Improvement is when compared to avg. cost in Run 2. Penalty weight for workspace variables was 20 to enforce tracking accuracy as compared to 1 for other variables.

As shown in Figure 1, Run 1 executes a trajectory for initial sample collection and hence is not included in Table 1. At deployment time for the repetitive task, the environment has changed as seen in the ground truth mode distribution from Run 2 onwards. Over iterative runs, the posteriors converge to the shifted ground truth and the tracking accuracy visibly improves by Run 4 as further seen by the cost metric in Table 1. The mapping accuracy does not increase significantly with runs as the samples only occupy a small portion of the environment. However, the accuracy in a region *around the repetitive task* is sufficiently high to generate significant improvements in control performance.

5 Conclusions and Future Directions

The proposed algorithm demonstrates improved performance over successive iterations of a repetitive task for a system subject to time-varying dynamics. Future work would involve a) providing safety guarantees under high mode uncertainty, b) being able to handle for previously unseen modes in the environment, c) biasing cost to prioritize collecting residuals that better identify the active mode.

[1] A summary of the entire algorithm with visualizations of component outputs can be found at Mode-mapping HGPMPC Results

References

Calandra, R.; Seyfarth, A.; Peters, J.; and Deisenroth, M. 2015. Bayesian Optimization for Learning Gaits under Uncertainty. *Annals of Mathematics and Artificial Intelligence (AMAI)*, 76.

D'Souza, L.; and Pant, Y. V. 2023. Stochastic Hybrid Model Predictive Control using Gaussian Processes for Systems with Piecewise Residual Dynamics. In *2023 American Control Conference (ACC)*.

Hewing, L.; Kabzan, J.; and Zeilinger, M. N. 2020. Cautious Model Predictive Control Using Gaussian Process Regression. *IEEE Transactions on Control Systems Technology*, 28(6): 2736–2743.

Kingma, D. P.; and Ba, J. 2017. Adam: A Method for Stochastic Optimization. arXiv:1412.6980.

Lew, T.; Sharma, A.; Harrison, J.; Bylard, A.; and Pavone, M. 2022. Safe Active Dynamics Learning and Control: A Sequential Exploration-Exploitation Framework. arXiv:2008.11700.

Nagy, T.; Amine, A.; Nghiem, T. X.; Rosolia, U.; Zang, Z.; and Mangharam, R. 2023. Ensemble Gaussian Processes for Adaptive Autonomous Driving on Multi-friction Surfaces. *arXiv preprint arXiv:2303.13694*.

Silverman, B. W. 2018. *Density estimation for statistics and data analysis*. Routledge.

Sitzmann, V.; Martel, J. N. P.; Bergman, A. W.; Lindell, D. B.; and Wetzstein, G. 2020. Implicit Neural Representations with Periodic Activation Functions. arXiv:2006.09661.

Yuan, Z.; Hall, A. W.; Zhou, S.; Brunke, L.; Greeff, M.; Panerati, J.; and Schoellig, A. P. 2022. safe-control-gym: a Unified Benchmark Suite for Safe Learning-based Control and Reinforcement Learning in Robotics. arXiv:2109.06325.