CS 330 Autumn 2020 Final Project Report
Embedding Physics in Meta-Learning for a Dynamical System: Analysis of a Cart-Pole System

Somrita Banerjee [1]

Extended Abstract

For various dynamical systems, the parameters of the system are not known beforehand and may even change online. Effectively controlling such dynamical systems requires that the control algorithm adapt online as the parameters of the system change. Control algorithms have been developed that perform end-to-end reinforcement learning (Nagabandi et al., 2018) but these black-box frameworks are difficult to trust for safety-critical systems. Instead, model-based control is used where the adaptation of control is driven by adaptation of the model e.g. (Harrison et al., 2018a). Model identification with varying parameters is effectively solved using meta-learning by leveraging prior data to create a generalized model of dynamics and training the model explicitly to adapt quickly to an unknown set of parameters using few observations of dynamics.

Conventionally, data-driven meta-learning discards rather than leverages existing nominal physics models, which may not be highly accurate, but can provide a starting point for learning and, importantly, can be used to understand and explain how the otherwise black-box meta-learned model adapts. The focus of this project is to investigate how to best embed nominal physics based models in meta-learning models for dynamical systems.

We evaluate our models on a simple cart-pole system. First, we evaluate meta-learning alone and a linear physics-based model alone as tools for system identification. The meta-learning framework used is ALPaCA (Adaptive Learning for Probabilistic Connectionist Architectures) (Harrison et al., 2018b). Second, we evaluate the benefit of simply adding the linear physics-based model to the meta-learned model. Third, we propose a method to adapt not only the meta-learned parameters, but also the nominal model parameters online, producing parameter estimates that are updated online and a model whose adaptation is interpretable. Fourth, we investigate techniques to make the nominal model features and the meta-learned features independent. Fifth, we evaluate the framework in a situation where the task changes midway through a rollout, i.e. a situation that violates a fundamental assumption of meta-learning, to identify areas of further improvement. The different frameworks chosen are compared to each other along two metrics - their ability to predict dynamics and their ability to estimate the parameters of the system.

First, we compare the ability of the models to predict dynamics. The ALPaCA model alone uses just a few observations online to update the prior belief, reduce prediction uncertainty, and provide posteriors close to ground truth. However, these predictions are susceptible to oscillating errors, which result from overshooting the true dynamics and then using consecutive measurements to correct the belief. Simply adding the nominal model allows the ALPaCA model to learn the residual and helps reduce the oscillations. The best performance is seen when the ALPaCA model is augmented with the nominal model features and the parameters are jointly updated. In this case, the predictions are the smoothest with the lowest average error.

Second, we compare the ability of the models to estimate the physical parameters of the system. Least square regression estimates provide a single parameter value, not a probability distribution, and do not account for time dependence between estimates. When the nominal model is added to ALPaCA, only the meta-learned parameters get updated online which have no physical significance. Therefore, these parameters do not regress to anything meaningful. However, when the adaptive linear model parameters are updated alongside the meta-learned parameters, the former correspond to physical quantities and these estimates converge quickly to the true parameter values.

In this project, we apply meta-learning to identify a dynamical system with unknown parameters as quickly as possible. We also propose techniques to embed known simplified physics models in the meta-learning framework. We show that if the physics models are affine in physically significant parameters, the evolution of those parameters can be used to explain and probe the adaptation of the otherwise black-box dynamics adaptation. Our combined framework can utilize only a few observations online to adapt to different unknown parameters.

[1]Aeronautics and Astronautics Department, Stanford University, Stanford, California, USA. Correspondence to: Somrita Banerjee <somrita@stanford.edu>.

Introduction

For various dynamical systems, the parameters of the system are not known beforehand and may even change online. This is the case for systems with unknown or difficult-to-model dynamics such as the dynamics of a wheel as it travels over rough sandy terrain (Shibly et al., 2005) or the dynamics of a gripper with a novel gecko-inspired adhesive (Estrada et al., 2016). This is also the case for systems where the parameters vary considerably during test time such as predicting the dynamics of a drone in varying wind conditions (O'Connell et al., 2019) or predicting the amount of solar power that can be stored in varying weather conditions (Wang & Koprinska, 2018).

Effectively controlling such dynamical systems often requires that the control algorithm adapt online as the parameters of the system change. Control algorithms have been developed for parameter-varying tasks that perform end-to-end reinforcement learning for control (Nagabandi et al., 2018). However, for safety-critical systems, a black-box end-to-end control framework is difficult to trust and we often lean on model-based control. In the case of model-based control, the adaptation of control is driven by adaptation of the model (e.g. (Harrison et al., 2018a)), making system identification especially crucial for dynamical systems with varying parameters.

In order to quickly identify dynamical systems with varying parameters, the model of the system must quickly adapt online to a new set of parameters. If the model is known (even approximately) in a closed form, e.g. from physics, the model can be adapted online using regression (Andrews, 1974) to fit the parameters that best explain the observed dynamics. On the other hand, if prior information is available in the form of data from dynamical systems with different parameters, a data-driven approach such as meta-learning (Finn et al., 2017) can be used. By treating dynamical systems with different parameters as different tasks, the data-driven model can be trained to adapt quickly to a new task, i.e. adapt quickly online to a system with unknown parameters (Harrison et al., 2018a). Meta-learning is especially suited to identify systems with varying parameters since it can leverage prior data to create a generalized model of dynamics and can be trained explicitly to adapt quickly to an unknown set of parameters using few observations of dynamics.

This work is focused on applying meta-learning to system identification for dynamical systems with varying parameters. Conventionally, data-driven meta-learning discards rather than leverages existing nominal physics models. These existing nominal physics models may not be a perfect or highly accurate representation of the true dynamics, especially for dynamics that are novel — such as the behavior of a new gecko-inspired adhesive — or have complex interaction effects — such as the movement of a wheel on deformable sandy terrain. However, these nominal physics models can provide a starting point for the model to learn from and, importantly, can be used to understand and explain how the otherwise black-box meta-learned model adapts.

The contributions of this project are fivefold. First, we evaluate meta-learning alone and a nominal physics-based model alone as tools for system identification for dynamics with varying parameters. Second, we evaluate the benefit of simply adding a nominal physics-based model to the meta-learned model. Third, we propose a method to adapt not only the meta-learned parameters, but also the nominal model parameters online, producing parameter estimates that are updated online and a model whose adaptation is interpretable. Fourth, we investigate techniques to make the nominal model features and the meta-learned features independent, though these techniques did not work well. Fifth, we evaluate the framework in a situation where the task changes midway through a rollout, i.e. a situation that violates a fundamental assumption of meta-learning, to identify areas of further improvement.

Problem statement

The focus of this project is to investigate how to best embed nominal physics based models in meta-learning models for dynamical systems. In order to evaluate the performance of the various techniques, we consider a simple cart-pole system where a cart moves in one dimension along the x axis and a pole is attached to the cart but free to rotate about its pivot (see Figure 1). As a force (input) is applied to the cart, the state of the system — described by the position of the cart, velocity of the cart, the angle that the pole makes with the vertical, and the angular velocity of the pole — evolves with time. For this cart-pole system, the true dynamics model is known but for the purposes of evaluation, we assume we only have access to a simplified linear dynamics model. This is intended to mimic situations where we know simplified dynamics but not the full dynamics of the system, e.g. we know the Bekker terramechanics model (Bekker, 1969) but not the true complicated and unmodeled dynamics of a wheel traversing rough sandy terrain. The purpose of this project is to investigate how such nominal physics models can be used in conjunction with meta-learning to identify systems with varying parameters quickly online. For the cart-pole system under consideration, the parameters that are unknown (and vary between tasks) are the length of the pole and the mass of the pole. The mass of the cart is assumed to be known.

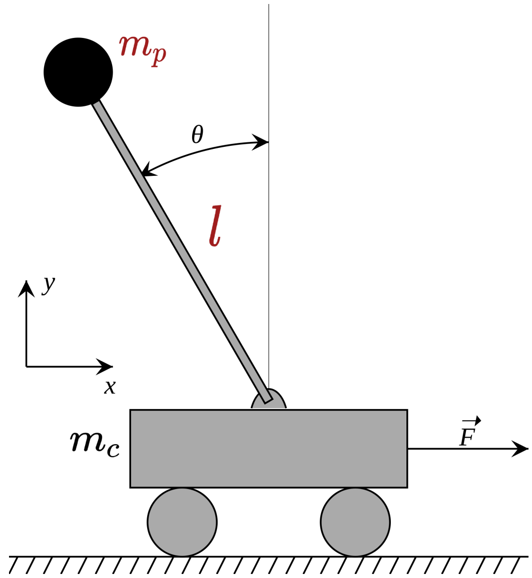

Figure 1. A cart-pole system where the two parameters that are unknown are the mass of the pole and the length of the pole.

Approach

First, we evaluate meta-learning alone and a linear physics-based model alone as tools for system identification of the cart-pole system. Second, we evaluate the benefit of simply adding the linear physics-based model to the meta-learned model. Third, we propose a method to adapt not only the meta-learned parameters, but also the nominal model parameters online, producing parameter estimates that are updated online and a model whose adaptation is interpretable. Fourth, we investigate techniques to make the nominal model features and the meta-learned features independent, though these techniques did not produce any improvements in practice. Fifth, we evaluate the framework in a situation where the task changes midway through a rollout, i.e. a situation that violates a fundamental assumption of meta-learning, to identify areas of further improvement.

Meta-learned model alone

The meta-learning framework considered in this work is ALPaCA (Adaptive Learning for Probabilistic Connectionist Architectures) (Harrison et al., 2018b). ALPaCA learns a neural network model offline that is trained to be able to rapidly incorporate online information to predict posterior distributions. Online adaptation is achieved by performing Bayesian linear regression on the last layer of the network, and thus is also capable of providing confidence intervals for predictions. In other words, ALPaCA learns priors over all the network weights, including the last layer weights, so that it can optimally predict the posterior by conditioning on just a few online observations and updating only the last layer of weights. The training is done to minimize the divergence between the predicted and true posterior, given a small number of online observations.

I chose to use ALPaCA over some other meta-learning framework like MAML (Finn et al., 2017) because: 1. I'm already familiar with its implementation and code base, making it easier to get started on this project and 2. the Bayesian regression formulation makes it easy to reason about prior and posterior distributions over physical parameters (rather than updates using gradient descent for MAML).

In the ALPaCA framework for learning cart-pole dynamics, the input is the current state and force applied on the cart and the output is the next state of the cart-pole system. Using offline data from systems with different values of length of the pole and mass of the pole, the model is trained to generalize to a system with unknown parameters using only a few observations. The ALPaCA model outputs not only a mean prediction for the next state, but also a probability distribution over the next state. The model's prediction over the next state $\boldsymbol{x}_+$ (which consists of position x, angle θ, as well as their time derivatives)

$$\boldsymbol{x}_+ = h(\boldsymbol{x}, \boldsymbol{u}; \theta_{\text{net}}) = \boldsymbol{\Phi}_{\text{net}}(\boldsymbol{x}, \boldsymbol{u}) \;\; \theta_{\text{net}} \tag{1}$$

is a function of the model parameters θ_{net} (which are also the last layer weights) and a matrix of meta-learned model features over the current state $\boldsymbol{x}$ and current action $\boldsymbol{u}$ in the form of the rest of the network weights $\boldsymbol{\Phi}_{\text{net}}(\boldsymbol{x}, \boldsymbol{u})$.

During offline training, the initializations for the model features $\boldsymbol{\Phi}_{\text{net}}(\boldsymbol{x}, \boldsymbol{u})$ and model parameters θ_{net} are learned. During online task-specific adaptation, only the model parameters θ_{net} (last layer weights) and their associated covariances are adapted.

Nominal model alone

We develop a simplified nominal model that is linear in parameters in order to compare it to the meta-learned model. This comparison should show whether a meta-learned model is significantly better than a simplified physics model. The equations of motion for a cart-pole system are well-known as

$$\begin{aligned}\ddot{x} &= \frac{F + m_p g \sin\theta\cos\theta + m_p l \dot{\theta}^2 \sin\theta}{m_c + m_p \sin^2\theta} \\ \ddot{\theta} &= \frac{-F\cos\theta - (m_p + m_c) g \sin\theta - m_p l \dot{\theta}^2 \sin\theta\cos\theta}{l(m_c + m_p \sin^2\theta)},\end{aligned} \tag{2}$$

where x is the position of the cart, F is the force applied on the cart, θ is the angle that the pole makes with the vertical, m_c is the mass of the cart, m_p is the mass of the pole, g is the acceleration due to gravity, and l is the length of the pole. We assume that the mass of the cart m_c and gravity g are known. In order to parameterize the equations of motion as linear in the unknown parameters — mass of the pole m_p and length of the pole l — we assume that $m_p << m_c$, therefore we approximate $m_c + m_p \sin^2\theta$ as m_c, and develop the formulation

$$\begin{aligned} \ddot{x} &= \begin{bmatrix} \frac{g\sin\theta\cos\theta}{m_c} & \frac{\dot{\theta}^2\sin\theta}{m_c} \end{bmatrix} \begin{bmatrix} m_p \\ m_p l \end{bmatrix} + \frac{F}{m_c} \\ \ddot{\theta} &= \begin{bmatrix} \frac{-g\sin\theta - l_{\text{est}}\dot{\theta}^2\sin\theta\cos\theta}{l_{\text{est}}m_c} & 0 \end{bmatrix} \begin{bmatrix} m_p \\ m_p l \end{bmatrix} + \\ &\quad \frac{-F\cos\theta - m_c g\sin\theta}{l_{\text{est}}m_c}. \end{aligned} \tag{3}$$

In order to use these linear equations, an estimate of the length l_{est} is used that is not updated online.

This simplified linear model can be used in two ways. If given the current state, the force applied, and a noisy estimate of the parameters m_p and l, the model can be used to (noisily) predict the next state. Alternatively, given data from transitions over a few time steps, the model can be used to estimate the parameters m_p and l using linear regression of the form

$$\begin{bmatrix} \left[\ddot{x} - \frac{F}{m_c}\right]_1 \\ \left[\ddot{x} - \frac{F}{m_c}\right]_2 \\ \vdots \end{bmatrix} = \begin{bmatrix} \left[\frac{g\sin\theta\cos\theta}{m_c}\right]_1 & \left[\frac{\dot{\theta}^2\sin\theta}{m_c}\right]_1 \\ \left[\frac{g\sin\theta\cos\theta}{m_c}\right]_2 & \left[\frac{\dot{\theta}^2\sin\theta}{m_c}\right]_2 \\ \vdots & \vdots \end{bmatrix} \begin{bmatrix} m_p \\ m_p l \end{bmatrix}$$
$$\text{i.e.} \quad \boldsymbol{b} = \boldsymbol{A} \begin{bmatrix} m_p \\ m_p l \end{bmatrix}. \tag{4}$$

The linear model only outputs a mean prediction over the next state $\boldsymbol{x}_+$

$$\boldsymbol{x}_+ = g(\boldsymbol{x}, \boldsymbol{u}; \theta_{\text{nom}}) = \boldsymbol{\Phi}_{\text{nom}}(\boldsymbol{x}, \boldsymbol{u})\ \theta_{\text{nom}} + \boldsymbol{\phi}_{\text{nom}}(\boldsymbol{x}, \boldsymbol{u}) \tag{5}$$

which is a function of the model parameters m_p and l lumped together in θ_{nom}, as well as a matrix $\boldsymbol{\Phi}_{\text{nom}}(\boldsymbol{x}, \boldsymbol{u})$, and a vector $\boldsymbol{\phi}_{\text{nom}}(\boldsymbol{x}, \boldsymbol{u})$ which are both known functions of the current state $\boldsymbol{x}$ and current action $\boldsymbol{u}$ as described in equation 3.

Adding the nominal model to the meta-learned model

The simplest way of leveraging the physics-based nominal model is to simply add it to the meta-learned model, which should provide a good starting point for learning the residual or deviation of the true dynamics from the nominal model dynamics. In this case, the next state $\boldsymbol{x}_+$

$$\boldsymbol{x}_+ = \boldsymbol{\Phi}_{\text{net}}(\boldsymbol{x}, \boldsymbol{u})\ \theta_{\text{net}} + \boldsymbol{\Phi}_{\text{nom}}(\boldsymbol{x}, \boldsymbol{u})\ \theta_{\text{nom}} + \boldsymbol{\phi}_{\text{nom}}(\boldsymbol{x}, \boldsymbol{u}) \tag{6}$$

is simply an addition of the meta-learned model and the nominal model. The nominal model features $\boldsymbol{\Phi}_{\text{nom}}(\boldsymbol{x}, \boldsymbol{u})$ and $\boldsymbol{\phi}_{\text{nom}}(\boldsymbol{x}, \boldsymbol{u})$ are known functions, and a guess must be made for the nominal model parameters to use for θ_{nom}. During training, initializations are learned for the meta-learned model features $\boldsymbol{\Phi}_{\text{net}}(\boldsymbol{x}, \boldsymbol{u})$ and parameters θ_{net}. During online task-specific adaptation, only the meta-learned model parameters θ_{net} and their associated covariances are adapted.

Fundamentally, this is equivalent to making an initial guess for the physical parameters m_p and l, creating a linear nominal model with those parameter values, and then using a meta-learned model to adapt to the difference between the linear model and the true dynamics (that will likely have parameters different from the initial guess). It is expected that since the guess of the nominal model provides a starting point for learning, the training of the meta-learned model here should be faster than the training of the meta-learned model alone.

Adaptive nominal model parameters with the meta-learned model

The adaptation in the meta-learned models mentioned previously have all been black-box adaptation. There was no way to monitor or estimate the parameters m_p and l corresponding to physical quantities. However, now we take advantage of the fact that the nominal model is linear in the parameters θ_{nom} to stack these nominal model parameters with the meta-learned parameters θ_{net} and adapt both online. In this formulation, the next state $\boldsymbol{x}_+$

$$\boldsymbol{x}_+ = \begin{bmatrix} \boldsymbol{\Phi}_{\text{nom}}(\boldsymbol{x}, \boldsymbol{u}) & \boldsymbol{\Phi}_{\text{net}}(\boldsymbol{x}, \boldsymbol{u}) \end{bmatrix} \begin{bmatrix} \theta_{\text{nom}} \\ \theta_{\text{net}} \end{bmatrix} + \boldsymbol{\phi}_{\text{nom}}(\boldsymbol{x}, \boldsymbol{u}) \tag{7}$$

is predicted using the known nominal model features $\boldsymbol{\Phi}_{\text{nom}}(\boldsymbol{x}, \boldsymbol{u})$ and $\boldsymbol{\phi}_{\text{nom}}(\boldsymbol{x}, \boldsymbol{u})$, the meta-learned features determined during training $\boldsymbol{\Phi}_{\text{net}}(\boldsymbol{x}, \boldsymbol{u})$, as well as the parameters for both the nominal model θ_{nom} and for the meta-learned model θ_{net} that are both updated online for an unknown task.

This formulation allows us to explain the adaptation of the model's predictions of the next state $\boldsymbol{x}_+$ by monitoring the adaptation of the model's estimate of the nominal model parameters θ_{nom} which have a physical significance. For example, if the model's estimate of the length of the pole l is negative or otherwise outside a reasonable range, we know that the model cannot currently be trusted to correctly predict the dynamics.

Enforcing orthogonality of features

I expected that the previous adaptive approach could lead to a situation where one of the meta-learned features overlaps heavily with a nominal model feature, e.g. the nominal model feature corresponding to m_p. In this case, the estimate of the parameter m_p would be diluted because some of the physical effect of m_p would be attributed to a different meta-learned parameter. In order to avoid such a situation, I tried to encourage orthogonality of the features by adding a loss to the training of ALPaCA. This loss is in the form of a regularization term

$$\sum_{i=1}^{n_x} \left\| I - \mathbf{\Phi}_i \mathbf{\Phi}_i^T \right\|_F^2 \tag{8}$$

where $\mathbf{\Phi}_i$ is the ith row of the $\mathbf{\Phi}$ matrix, which in turn consists of both the nominal model features $\mathbf{\Phi}_{\text{nom}}(\boldsymbol{x}, \boldsymbol{u})$ and the meta-learned features $\mathbf{\Phi}_{\text{net}}(\boldsymbol{x}, \boldsymbol{u})$. As discussed in the next section, this loss ended up not having any significant impact on the results.

Changing task parameters in the same rollout

At the core of meta-learning is the idea that parameters vary across different tasks, but within a single task, the parameters remain constant. For example, when working with the Omniglot dataset, we train the meta-learning model to adapt to Bengali, Kannada, etc. but the underlying assumption is that at test time, each task only corresponds to one language and alphabets don't change at runtime.

With this in mind, we evaluate how our framework performs if this assumption is violated, i.e. if the task parameters are changed midway through a rollout. We tested the combined framework of a linear adaptive model with the meta-learned model on tasks where the parameters change midway, e.g. the mass of the pole and length of the pole change halfway through the rollout. This is not a realistic scenario for the toy cart-pole example but could be a realistic concern when applying meta-learning to systems where the parameters vary continuously or without clear segmentation, such as the motion of a wheel over deformable terrain where the terrain properties vary continuously and sometimes suddenly. As discussed in the next section, the combined framework performed surprisingly well even when parameters changed in the middle of a task.

Results

The different frameworks chosen — meta-learned model alone, nominal model alone, nominal model added to the meta-learned model, and adaptive nominal model parameters with the meta-learned model — are compared to each other along two metrics. The first metric is their ability to predict dynamics, i.e. the next state. The second metric is their ability to estimate the parameters of mass of the pole m_p and length of the pole l. For the purposes of illustration, two cart-pole systems are considered with different parameter values for mass of pole m_p and length of pole l. The same control input is applied to both systems and the resulting state evolution is different, as shown in Figure 2.

Predicting dynamics

As shown in Figure 2, using just ALPaCA alone produces dynamics predictions that track the ground truth quite well. The priors for ALPaCA are the same in both cases and they utilize just a few observations online to provide the correct (and different) posteriors for the two cases. For the first three seconds, the model predictions have high uncertainty which quickly tapers down. It is worth noting that a large portion of this test case includes values of state that are outside the range used for training. The parameter values for m_p and l are also chosen to be at least partially outside the range used for training.

To better analyze the behavior of the various model frameworks, we switch from looking at the dynamics prediction itself to looking at the dynamics prediction error, as shown in Figure 3. The meta-learning model ALPaCA alone (mean depicted with an orange line and 90% confidence interval shown shaded) predicts dynamics with low error but it is susceptible to oscillating errors, which result from overshooting the true dynamics and then using consecutive measurements to correct its belief. Just simply adding the nominal model allows the ALPaCA model (green) to learn the residual and usually helps reduce the oscillations. The best performance is seen when the ALPaCA model is augmented with the nominal model features and the parameters are jointly updated (purple). In this case, the predictions are the smoothest with the lowest average error.

Estimating parameters

One of the main downsides to using a meta-learning data-driven only approach is that we lose the information about what parameters drive the adaptation that we observe. This kind of black-box model is not well-suited to safety critical applications. Therefore, the adaptation of parameter estimates is discussed in this section.

The parameter estimates for the two configurations under consideration are shown in Figure 4. If the physics-based linear model is used by itself, least squares regression can be used to determine the parameters of the system (m_p and l) that best explain the observations. These parameter estimates, shown in blue, only estimate a mean parameter value, not a probability distribution. Further, this type of regression does not take time dependence into account. At each time step, the regressed parameters are the result of fitting all the data up till that point to the model, without consideration for the previous time step's parameter estimates.

If the linear model is simply added on to ALPaCA, the

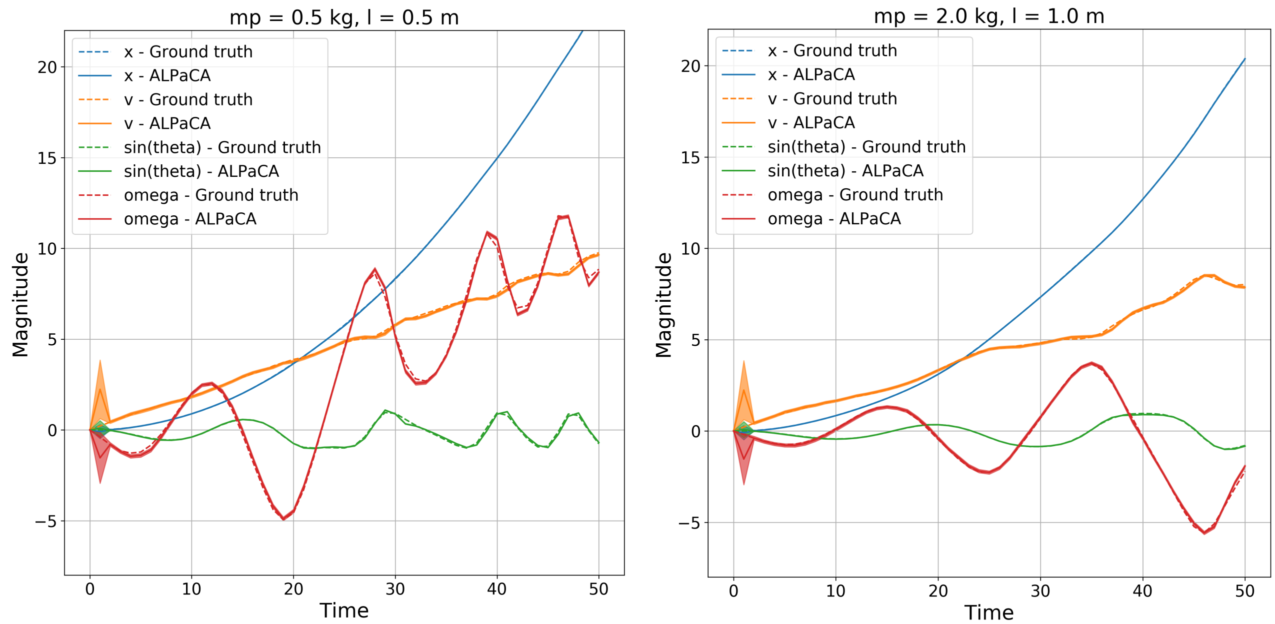

Figure 2. The meta-learning model ALPaCA alone is able to make good dynamics predictions (x, v, θ, ω) that track ground truth closely for 2 different configurations of the cart-pole dynamics. Each configuration has different values for the parameters: mass of the pole m_p and length of the pole l.

only parameters that get updated online for a task are the meta-learned parameters that have no physical significance. Therefore, these parameters (shown in green and with the 90% confidence interval shaded) do not regress to anything meaningful.

However, when the adaptive linear model parameters are updated alongside the meta-learned parameters, the linear model parameters correspond to physical quantities (the mass of the pole m_p and the length of the pole l). Therefore, we see that the parameter estimates for the adaptive linear model with ALPaCA (shown in purple) converge quite quickly to the correct posterior belief for m_p and l in each of the 2 configurations, despite starting with the same prior for both cases.

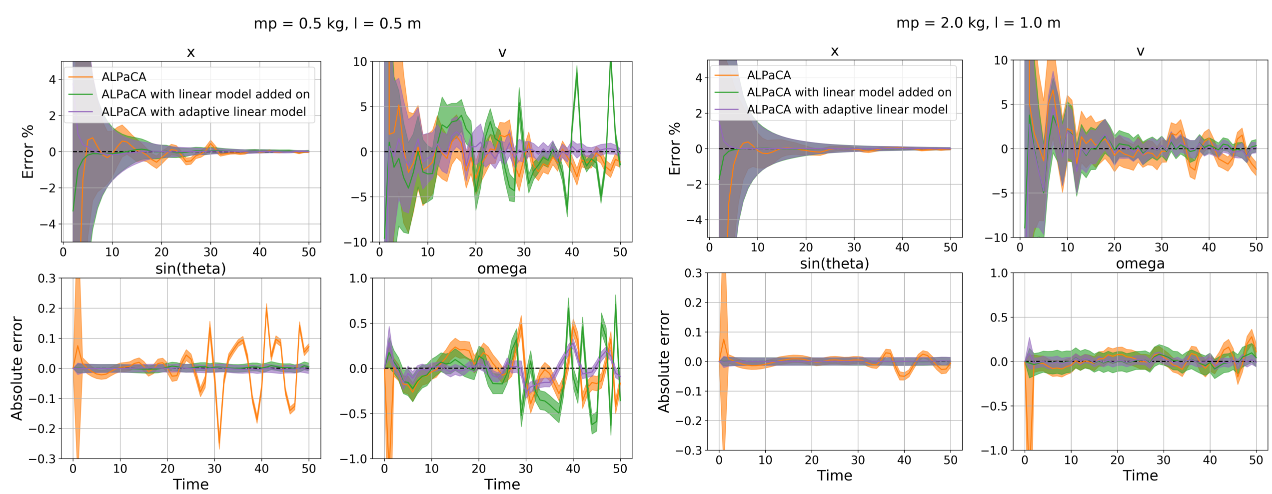

Figure 3. The meta-learning model ALPaCA alone, ALPaCA with the nominal model added on, and ALPaCA with adaptive nominal model parameters all predict dynamics with low error. Here, 0 error is a perfect prediction. The two configurations have different values for the parameters: mass of the pole m_p and length of the pole l.

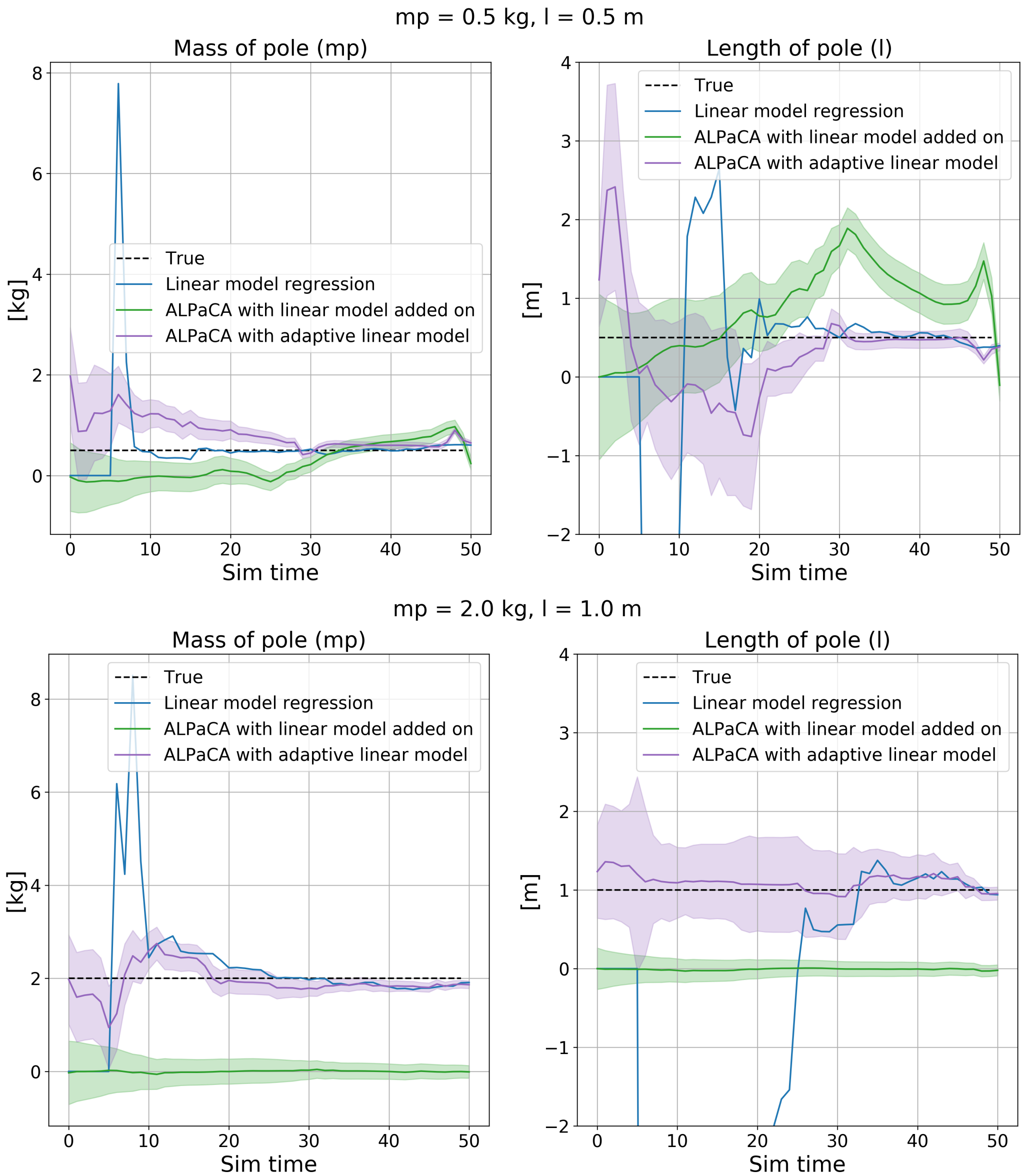

Figure 4. Various methods to estimate parameters such as linear least-squares regression (blue), adding the linear model to a meta-learned model (green), and making the linear model adaptive with the meta-learned model (purple) are compared to test the ability of ALPaCA to estimate unknown parameters. The ground-truth parameter values are shown in black. Each of the 2 configurations has different values for the parameters: mass of the pole m_p, and length of the pole l

.

Encouraging orthogonality of features

Encouraging orthogonality of features using the regularization loss discussed previously did not have a significant impact on the dynamics predictions or the parameter estimates. I suspect that because this was a simple dynamical system, the conditions where nominal features and meta-learned features could overlap just did not arise, and therefore encouraging orthogonality was not required. Since the plots for the model with orthogonality regularization are virtually identical to the plots for ALPaCA with adaptive linear model, I did not include separate figures for this work.

Changing task parameters at test time

The different models are tested when a fundamental assumption of meta-learning is violated, i.e. when parameters or tasks change within the same rollout. The task parameters are changed midway through a rollout, as seen in

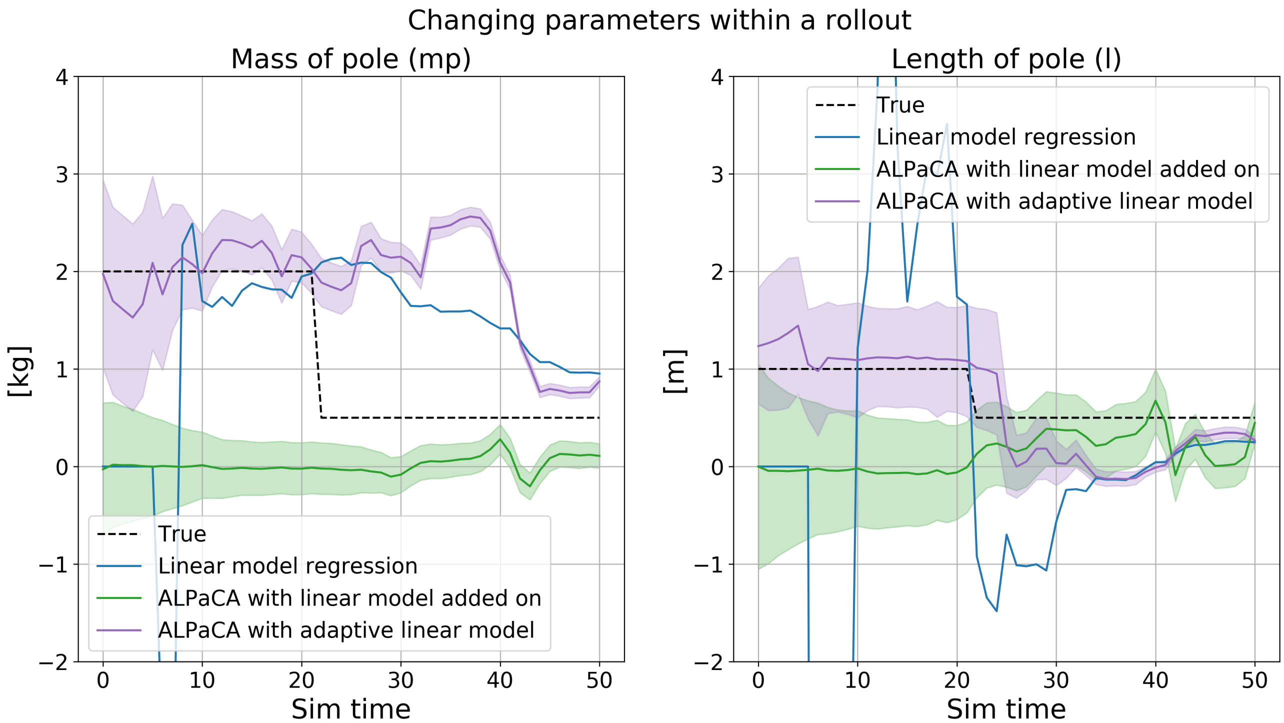

Figure 5. When task parameters — mass of the pole and length of the pole — change within a rollout (ground truth shown as a black dashed line), both the linear regression model (shown in blue) and ALPaCA with an adaptive nominal model (shown in purple) are able to adapt to the change, with the adaptive model sometimes adapting faster.

Figure 5, without resetting the model so the model continues operating with a low uncertainty. We expect that the linear regression model should be able to regress to the new parameters once sufficient data has been collected. As previously noted, the ALPaCA model with the linear model just added on only adapts the meta-learned parameters at test time which have no physical interpretation so those estimates are meaningless. Somewhat surprisingly, we observe that the ALPaCA model augmented with an adaptive nominal model is able to estimate the new parameters as well, though the uncertainty estimates are no longer reliable. Theoretically, this result makes sense since a sufficient number of observations of the new dynamics will update the posterior belief. However, this framework alone does not produce satisfactory results since the mean estimate takes some time to change and the uncertainty estimate is no longer meaningful. New techniques must be adapted if tasks change without warning at test time, some of which we discuss in the future work section.

Conclusions

In this work, we apply meta-learning to adapt to a dynamics system that has unknown varying parameters with the goal of identifying the system online as quickly as possible. We propose ways to embed known simplified physics models in the meta-learning process. We also show that if the physics models are affine in physically significant parameters, the evolution of those parameters can be used to explain and probe the adaptation of the otherwise black-box dynamics adaptation. Lastly, we show how the same prior beliefs for the meta-learned model ALPaCA are updated using just a few online observations to produce the correct posterior distributions for dynamical systems with different unknown parameters.

Future work

One avenue of future work is to extend this framework to work with non-affine nominal models, i.e. to stack the non-affine nominal models with the meta-learned models while still allowing for adaptation of the nominal model parameters. Another direction of future work is to analyze why the confidence for some meta-learned model predictions becomes very small even when the mean is wrong. Lastly, another avenue of future work is to autonomously detect when the dynamics parameters change, e.g. when the winds change for a drone, so that the meta-learning approach can be extended to continually varying tasks. A starting point for this direction of future work would be to consider meta-learning without task segmentation (Harrison et al., 2020).

Code submission

The code for this project is available at https://github.com/somritabanerjee/cs330-final-project. This is a private GitHub repository but I have added all the TAs (Rafael, Dilip, Mason, Albert, Karen, Nikita, and Suraj) as collaborators. I left the repository private in case this work is extended for a future publication. I would like to thank Professor Finn and all the TAs for a great learning experience!

References

Andrews, D. F. A robust method for multiple linear regression. *Technometrics*, 16(4):523–531, 1974.

Bekker, M. G. *Introduction to Terrain-vehicle Systems*. First edition, 1969.

Estrada, M. A., Hockman, B., Bylard, A., Hawkes, E. W., Cutkosky, M. R., and Pavone, M. Free-flyer acquisition of spinning objects with gecko-inspired adhesives. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4907–4913. IEEE, 2016.

Finn, C., Abbeel, P., and Levine, S. Model-agnostic meta-learning for fast adaptation of deep networks. *arXiv preprint arXiv:1703.03400*, 2017.

Harrison, J., Sharma, A., Calandra, R., and Pavone, M. Control adaptation via meta-learning dynamics. In *Workshop on Meta-Learning at NeurIPS*, volume 2018, 2018a.

Harrison, J., Sharma, A., and Pavone, M. Meta-learning priors for efficient online bayesian regression. In *Workshop on Algorithmic Foundations of Robotics*, 2018b.

Harrison, J., Sharma, A., Finn, C., and Pavone, M. Continuous meta-learning without tasks. *Advances in neural information processing systems*, 33, 2020.

Nagabandi, A., Clavera, I., Liu, S., Fearing, R. S., Abbeel, P., Levine, S., and Finn, C. Learning to adapt in dynamic, real-world environments through meta-reinforcement learning. In *International Conference on Learning Representations*, 2018.

O'Connell, M., Shi, G., Shi, X., and Chung, S.-J. Meta-learning-based robust adaptive flight control under uncertain wind conditions. 2019.

Shibly, H., Iagnemma, K., and Dubowsky, S. An equivalent soil mechanics formulation for rigid wheels in deformable terrain, with application to planetary exploration rovers. *Journal of terramechanics*, 42(1):1–13, 2005.

Wang, Z. and Koprinska, I. Solar power forecasting using dynamic meta-learning ensemble of neural networks. In *International Conference on Artificial Neural Networks*, pp. 528–537. Springer, 2018.