A Comparison of Autoregressive Hidden Markov Models for Multi-Modal Manipulations with Variable Masses

Oliver Kroemer[1], and Jan Peters[2]

***Abstract*—In contact-based manipulations, the effects of the robot's actions change as contacts are made or broken. For example, if a robot applies an increasing upward force to an object, then the force will eventually overcome the object's weight and break the object-table contact. The robot can subsequently raise or lower the height of the object. The transition from resting on the table to not being in contact with the table is an example of a mode switch. The conditions for this mode switch depend on the mass of the object being manipulated. By modeling the mode switch, the robot can estimate the mass of the object based on the conditions when the mode switch occurs. The robot can also use the model to predict when the object will break contact given its mass.**

We evaluated four different autoregressive hidden Markov models for representing manipulations with mass-dependent mode switches. The models were successfully evaluated on pushing and lifting tasks. The evaluations show that the predicted object trajectories and estimated object masses are more accurate when using models that interpolate between different masses, and that consider the observed state for estimating the mode switches.

***Index Terms*—Learning and Adaptive Systems, Model Learning for Control, Perception for Grasping and Manipulation**

I. Introduction

CONTACT-BASED manipulations are inherently discontinuous. As contacts are made or broken, the effects of the robot's actions change. For example, in order to push an object, a robot first has to place its hand on the side of the object. Making contact with the object constrains the hand's movements, but it allows the robot to apply a force onto the object. Applying a sufficiently large force will overcome the object's static friction and allow the robot to slide the object. The robot thus begins moving the object as the contact between the object and the supporting surface starts slipping.

The described pushing task consists of three parts: approaching the object with the hand, loading up the force on the object, and sliding the object. Each of these three parts is referred to as a *mode* [1], [2]. Modes are sometimes also referred to as phases or action phases in the literature [3], [4]. The effects of the robot's actions within each mode are continuous. The robot switches between the different modes by fulfilling certain conditions, i.e., making contact with the object and applying a sufficiently large force.

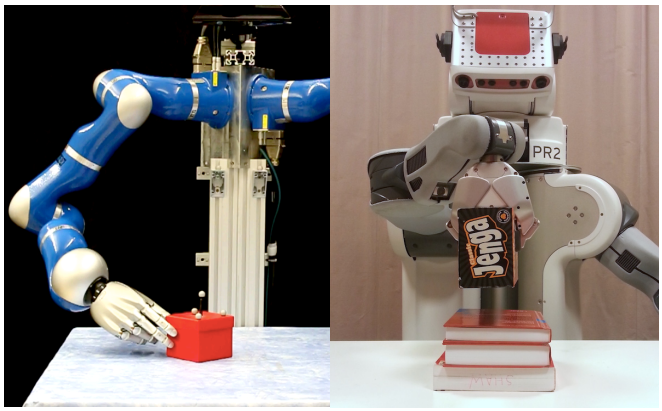

Fig. 1. The robots estimate the boxes' masses by (left) pushing or (right) lifting them and detecting the contact state changes, i.e., the mode switches.

The conditions required to switch from one mode to another often depend on the mass of the object, e.g., the pushing force required to overcome the object-surface static friction depends on the object's weight. Larger forces will often need to be applied to manipulate more massive objects, and some mode switches may not be feasible if an object is too heavy or too light. The conditions for switching between modes thus provide information regarding the object's mass. The robot should therefore be capable of estimating the mass of an object based on the conditions when the mode switches occur. The robot could use this information to, for example, estimate the latent mass of an object by pushing it. Conversely, given the mass of an object, the robot should also be able to predict when mode switches will occur.

In this paper, we explore different models for representing mode switches for manipulating objects with variable masses, e.g., containers. The manipulations are modeled using different types of autoregressive hidden Markov models (ARHMMs), wherein the mode is represented by the model's latent state. We compare two standard types of ARHMMs, as well as two observed state-based transitions ARHMMs (STARHMMs), which estimate the probability of a mode switch based on

Manuscript received: September 10, 2016; Revised December 11, 2016; Accepted January 10, 2017. This paper was recommended for publication by Editor Prof. Dongheui Lee upon evaluation of the Associate Editor and Reviewers' comments. This work was funded in part by the NSF under grant CNS-1213128 and the ONR under grant N00014-14-1-0536. The research leading to these results also received funding from the European Community's Seventh Framework Programme (FP7-ICT-2013-10) under grant agreements 610878 (3rdHand) and 610967 (TACMAN).

[1] Oliver Kroemer is a member of the Robotic Embedded Systems Lab at the University of Southern California, CA, USA okroemer@usc.edu

[2] Jan Peters is a member of the Intelligent Autonomous Systems group at the Technische Universitaet Darmstadt and the Max Planck Institute for Intelligent Systems, Germany peters@ias.tu-darmstadt.de

Digital Object Identifier (DOI): see top of this page.

the observed state. The four models are described in Section III. The models were evaluated on a box pushing task and a box lifting task as shown in Fig. 1. The experiments are explained and discussed in Section IV. The experiments evaluated the accuracy of the models' predicted trajectories given the object's mass. The experiments also evaluated the models' ability to estimate the latent mass of a container based on an observed trajectory.

II. Related Work

Manipulation tasks are often modeled using multiple modes for planning and control. The conditions for switching between modes generally represent sensory subgoals for the overall manipulation tasks. Similar to human grasping [3], Romano et al. [5] proposed a robust pick-and-place controller that switches between discrete modes based on tactile and haptic signals. Hauser et al. [1] and Barry et al. [2] proposed multi-modal planning methods for manipulation tasks. Mode switches included making contact with the side of an object to then be able to push the object. Kroemer et al. [4] presented a method for decomposing manipulations into modes and subsequently learning motor primitive skills for transitioning between these modes. Koval et al. [6] proposed a robust and computationally efficient policy for pushing objects that consists of pre- and post-contact parts. The policy uses tactile feedback to localize the object in the robot's hand. Instead of using mode switches as subgoals for a skill, we are investigating models to capture the dependency of the switching conditions on the object's mass and using these models to estimate the latent mass of manipulated objects.

Estimating the mass of an object from pushes is a form of interactive perception, wherein the robot uses actions to observe specific aspects of its environment. Interactive perception is commonly used to segment scenes into objects [7] and to extract articulation models [8], [9]. It can also be used to estimate the contents of containers from feedback during manipulations. Saal et al. [10] proposed a method for actively estimating the viscosity of a bottle's liquid using tactile signals from different shaking actions. Chitta et al. [11] presented a method for detecting liquids in containers, and determining whether the containers were open or closed, from tactile signals during grasping and rolling actions. Sinapov et al. [12] proposed a framework for learning categories of objects based on visual, audio, and proprioceptive signals from ten exploratory actions. The categories included single-object categories, e.g., the color or contents of a container, as well as categories on pairs and groups of objects, e.g. objects varying by weight. Sinapov et al. [13] proposed a method for learning to order sets of objects according to their weight based on sensory feedback during exploratory behaviors. Our work focuses on estimating the mass of a container based on the mode switches. This information could be fused with additional estimates of the mass and object properties.

The models evaluated in this paper are types of hidden Markov models (HMMs). Different types of HMMs have been widely used in robotics. The models are often used to segment demonstrations of manipulation tasks into skills. Niekum et al. [14] proposed using beta process autoregressive HMMs to segment demonstrations into motion categories. They demonstrated their approach on a complicated assembly task. Kulic et al. [15] presented a segmentation and motion primitive learning framework that uses HMMs to model the motion primitives. Patel et al. [16] use a hierarchical HMM to segment human manipulation tasks. Rozo et al. [17] proposed using a parametric HMM to model a pouring skill that adapts to the initial weight of the bottle. While most of the related work has focused on segmenting demonstrations into skills, we use the HMMs to learn multi-modal transition models that capture the effects of the robot's actions and the conditions for switching between modes.

Identifying the dynamic properties and state transitions of objects and robots is an important aspect of robotics [18], [19], [20]. Wu et al. [21] recently proposed using deep learning and a realistic physics engine to estimate the latent masses and friction coefficients from observed object interactions. Finn et al. [22] proposed a deep learning approach to directly predict the effects of pushing actions on cluttered objects in the vision domain. While most of the work in this research area focuses on modeling the effects within each mode, our evaluation investigates different models for switching between the modes.

The main contribution of this paper is an evaluation of four different models for representing mode switches in manipulation tasks. These models can be used to estimate the latent mass of an object based on an observed trajectory, or to predict a trajectory for a given mass. Our experiments evaluate both of these applications of the models on pushing and lifting tasks.

III. Multi-Modal Manipulation Models

In this section, we present the four manipulation models evaluated in our experiments. The first two models are autoregressive hidden Markov models (ARHMMs), with mass independent and mass dependent mode switches. The other two models are observed state-based autoregressive hidden Markov models (STARHMMs). The graphical models of the four models are shown in Fig. 2.

A. Model Components

In all four models, the manipulation is modeled as a partially observable Markov decision process, wherein the discrete mode is the latent state. We denote the latent mode at time step t as $\rho_t \in \{1, ..., \kappa\}$ and the observed part of the state as $s_t \in \mathcal{S}$, where $\mathcal{S}$ is the space of observed states.

In the pushing experiment, the observed state $s_t \in \mathbb{R}^3$ consists of the hand position, the object position, and the desired hand position along the direction of the pushing motion. For the lifting experiment, the robot used the vertical components of the hand position and the desired hand position to create a $\mathbb{R}^2$ observed state space. The object position is redundant given the hand position, as the robot used the same top grasp for all of the trials. As the robot is using an impedance controller, it can use the difference between the actual and desired hand positions to estimate the force being applied to the object.

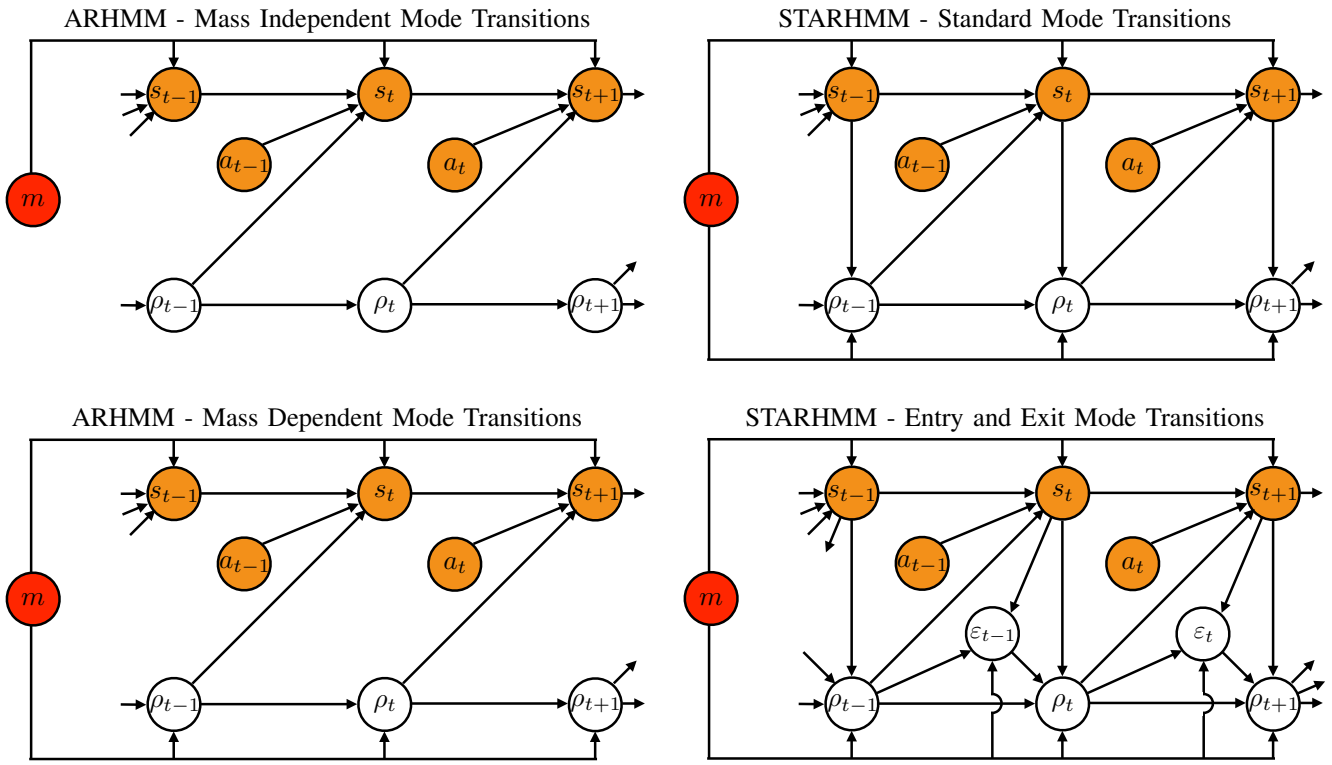

Fig. 2. The graphical models for the four evaluated models. The orange nodes indicate observed variables, while the white nodes indicate hidden variables. The mass is red as it can be either observed or hidden depending on the context.

The state also includes the mass of the object m, which we assume to be constant throughout the manipulation. The mass is latent when the robot is estimating the mass from an observed trajectory, and observed when predicting a trajectory given the mass. Although the object's mass can take any positive real value, one of the evaluated models assumes a discrete number of mass settings.

In addition to the state, the robot also performs an action $a_t \in \mathcal{A}$, where $\mathcal{A}$ is the action space. In our experiments, the robot used a Cartesian impedance controller to perform the manipulations. The actions were defined as the horizontal pushing or vertical lifting shifts in the desired hand position, resulting in $\mathbb{R}$ action spaces. The shifts were the same for each time step, resulting in a constant action across time.

The robot's actions will have different effects depending on the current mode. The models represent the effects using the observed state transition distribution $p(s_{t+1}|s_t, a_t, m, \rho_t)$. We use linear Gaussian models of the form $s_{t+1} \sim \mathcal{N}(A_{\rho_t}[m \;\; a_t \;\; 1]^T + s_t, \Sigma_{\rho_t})$ to represent the transition distribution. Each of the κ modes has its own set of matrices A_{ρ_t} and Σ_{ρ_t} to model the effects of different masses and actions. The parameters for all four of the models are estimated using standard message passing and expectation maximization methods [23], [24].

The models are evaluated on quasi-static pushing and lifting tasks. The state space therefore does not include the accelerations of the object or hand. The observed state transitions should also not rely on the mass of the object in this setting. However, we include the mass in the transition distribution as it may provide additional information for some models. For example, a one-mode model $\kappa = 1$ could capture that lighter objects are lifted higher or pushed further when given the same initial state and action sequences.

The difference between the four evaluated models is their representation of the mode transitions from ρ_t to ρ_{t+1}. The mode switches are important because they correspond to events such as when an object makes or breaks contact with another object, or when it begins to slip. The conditions for these events often depend on the mass of the object being manipulated.

B. *Autoregressive Hidden Markov Models*

The first two models that we evaluate are ARHMMs. Their graphical models are shown on the left of Fig. 2. The first ARHMM assumes that the mode transitions are independent of the mass and only depend on the previous latent mode $p(\rho_{t+1}|\rho_t)$. Given the mass independence of the model's mode transitions, we refer to this model as ARHMM-MassIndep. The transition model is represented by a single transition matrix $T \in \mathbb{R}^{\kappa \times \kappa}$, where the probability of transitioning from mode i to j is given by the element in the ith row and jth column $p(\rho_{t+1} = j|\rho_t = i) = [T]_{ij}$. The distribution over the first mode $p(\rho_1)$ is modeled as a discrete distribution over the κ modes.

The second model assumes that the transitions depend on the mass $p(\rho_{t+1}|\rho_t, m)$, and we denote this model as ARHMM-MassDep. For this model, we assume that the mass can only take one of a discrete set of values. The robot learns a separate transition matrix $T_m \in \mathbb{R}^{\kappa \times \kappa}$ for each of these mass values. In our experiments, the robot learned a single transition

matrix for the ARHMM-MassIndep and five for the ARHMM-MassDep, which correspond to the five discrete mass settings. The distribution over the first mode $p(\rho_1|m)$ is modeled as a separate discrete distribution over the κ modes for each mass setting.

C. Observed State-based Transitions Autoregressive HMMs

The remaining two models are observed state-based transitions autoregressive hidden Markov models (STARHMMs). These models differentiate themselves from the previous models by having the mode transitions depend on the observed state $p(\rho_{t+1}|\rho_t, s_t, m)$.

The first of the STARHMMs uses the standard logistic regression model to represent the transitions between the modes [25]. We refer to this model as STARHMM-STD. The probability of transitioning from mode $\rho_t = i$ to $\rho_{t+1} = j$ is given by

$$p(\rho_{t+1} = j|\rho_t = i, s_{t+1}, m) = \frac{\exp(\theta_{ij}^T \phi_{t+1})}{\sum_{k=1}^{\kappa} \exp(\theta_{ik}^T \phi_{t+1})},$$

where θ_{ik} with $k \in \{1, ..., \kappa\}$ are vectors of learned parameters that determine the transition conditions from the ith mode, and ϕ_{t+1} is a vector of features describing the observed state s_{t+1} and mass m. The features ϕ_t for the lifting and pushing tasks include the distance between the actual and desired hand positions, the distance between the hand and the box for pushing, and the mass of the box. The distribution over the first mode $p(\rho_1|s_1, m)$ is modeled as an additional logistic regression over the κ modes.

The final model is a variant of the STARHMM that models the entry and exit conditions of the modes [4]. We refer to this model as STARHMM-EE. The model introduces an additional binary termination variable $\varepsilon_t \in \{0, 1\}$. The distribution over the termination variable is given by

$$p(\varepsilon_t = 1|\rho_t = i, s_{t+1}, m) = \left(1 + \exp\left(\hat{\theta}_i^T \phi_{t+1}\right)\right)^{-1},$$

where $\hat{\theta}_i$ is a vector of parameters that define the termination probability for different conditions in the ith mode.

If the termination variable is low $\varepsilon_t = 0$ at time t, then the mode does not switch $p(\rho_{t+1} = \rho_t|\rho_t, s_t, m, \varepsilon_t = 0) = 1$. If the termination variable is high $\varepsilon_t = 1$, then the distribution over the next mode is given by the initiation distribution

$$p(\rho_{t+1} = j|\rho_t, s_{t+1}, m, \varepsilon_t = 1) = \frac{\exp(\check{\theta}_j^T \phi_{t+1})}{\sum_{k=1}^{\kappa} \exp(\check{\theta}_k^T \phi_{t+1})},$$

where $\check{\theta}_j$ is a vector of parameters that determine the initiation conditions for the jth mode. The initiation distribution is shared amongst all modes and does not depend on the previously terminated mode. The robot can transition to the same mode $\rho_{t+1} = \rho_t$ even if the termination variable is high $\varepsilon_t = 1$. The initiation distribution is used to define the distribution over the first mode $p(\rho_1|s_1, \varepsilon_0 = 1, m)$. Given the termination and initiation distributions, one can compute the phase transition distribution for the STARHMM-EE by marginalizing out the termination variable

$$p^{\text{EE}}(\rho_t|s_t, \rho_{t-1}, m) = \sum_{\varepsilon_{t-1}=0}^{1} p(\rho_t|\rho_{t\text{-}1}, \varepsilon_{t\text{-}1}, m) p(\varepsilon_{t\text{-}1}|s_t, \rho_{t\text{-}1}, m).$$

$p(s_{1:N+1}, a_{1:N}, \rho_{1:N}, m) =$ $p(m)p(s_1\|m)p^{\text{init}} \prod_{t=1}^{N} p(s_{t+1}\|s_t, a_t, \rho_t, m)p(a_t) \prod_{t=2}^{N} p_t^{\text{mode}}$		
	p^{init}	p_t^{mode}
ARHMM-MassIndep	$p(\rho_1)$	$p(\rho_t\|\rho_{t\text{-}1})$
ARHMM-MassDep	$p(\rho_1\|m)$	$p(\rho_t\|\rho_{t\text{-}1}, m)$
STARHMM-STD	$p(\rho_1\|s_1, m)$	$p(\rho_t\|s_t, \rho_{t\text{-}1}, m)$
STARHMM-EE	$p(\rho_1\|s_1, \varepsilon_0 = 1, m)$	$p^{\text{EE}}(\rho_t\|s_t, \rho_{t\text{-}1}, m)$

TABLE I
THE FOUR MODELS' JOINT DISTRIBUTIONS OVER THE STATES, ACTIONS, MODES, AND MASS FOR A TRAJECTORY WITH N STEPS.

While the STARHMM-STD learns a separate mapping from the observed state and mass to the next mode for each mode, the STARHMM-EE learns a single mapping that is shared across all modes, i.e., the initiation distribution. The termination variable allows the model to suppress this mapping to remain in the current mode until the additional mode-specific termination conditions are fulfilled.

The joint probability $p(s_{1:N+1}, a_{1:N}, \rho_{1:N}, m)$ for a trajectory of N steps is given in Table I for each of the models. This table highlights the different mode switching models.

D. Model Initialization

To initialize the expectation maximization (EM) algorithm for computing the model parameters, we cluster the samples of the trajectories using spectral clustering [26]. Spectral clustering requires a similarity value for each pair of samples. We compute the similarity using a squared exponential kernel function with the inputs given by the samples' changes in the observed state $s_{t+1} - s_t$. The kernel's length scales for each dimension are given by the standard deviations of the input data. The actions do not need to be taken into consideration as they are constant for all of the samples. The cluster assignments were subsequently used as the initial mode assignments of the samples and the initial model parameters were computed accordingly. For the STARHMM-EE, we initially assumed a low termination probability of 0.01 when consecutive samples were assigned to the same mode.

E. Trajectory Prediction

Given an initial observed state s_1, the object mass m, and a sequence of actions $a_{1:N}$, the robot uses the models to predict the states for the next N steps of the sequence. The robot begins by sampling the initial mode ρ_1. It subsequently samples the next observed state s_2 from the observed state transition distribution, followed by the mode ρ_2 based on the mode transition distribution. The robot continues to iterate between sampling the variables s_{t+1} and ρ_{t+1} until the end of the sequence. For the STARHMM-EE, the robot samples the termination variable ε_t according to the termination distribution before sampling the next mode ρ_{t+1} from the initiation distribution if $\varepsilon_t = 1$.

To evaluate the accuracy of the predicted trajectories, we compare the predicted sequence of states to the ones that the robot observed during the actual execution of the full sequence. For each test trajectory, 50 prediction trajectories were sampled as described above. The prediction performance

was then defined by the root mean squared error (RMSE) in the hand and box positions across all of the time steps for all 50 prediction trajectories and all of the trajectories in the test set.

F. Mass Estimation

Given a sequence of observed state $s_{1:N+1}$ and actions $a_{1:N}$, the robot can use the models to estimate the latent mass m of the manipulated object. We want to compute the Maximum a posteriori (MAP) estimate of the mass m^* given the observations, which is equivalent to computing

$$m^* = \arg\max_m [p(s_{1:N+1}, a_{1:N}|m)p(m)].$$

In our experiments, the prior over the mass $p(m)$ is a uniform distribution. This problem is equivalent to finding the mass that maximizes the joint distribution $p(s_{1:N+1}, a_{1:N}, m)$. We can compute this distribution from the joint distribution $p(s_{1:N+1}, a_{1:N}, \rho_{1:N}, m)$, as given in Table I, by marginalizing out the latent mode variables $\rho_{1:N}$. The marginalization can be performed efficiently using a forward message passing algorithm [23], [24]. In our evaluations, we use a discrete set of possible mass settings. The robot therefore evaluates the probability density $p(s_{1:N+1}, a_{1:N}, m)$ for each of the mass settings and selects the one with the highest value as the estimate m^*. The STARHMMs are not restricted to discrete mass settings. The robot could therefore perform a 1D search over the mass value to find the estimate. However, we restrict the search to discrete sets of mass settings for our evaluations.

Given a model and a set of test trajectories, the robot computes the MAP estimate of the mass for each trajectory individually. If two or more mass settings had equal posterior probabilities, we used the average mass as the estimate. The performance of the mass prediction is then given by the root mean squared error between the estimated masses and the true masses across all of the trajectories in the test set.

IV. Evaluations

The four models were evaluated on a pushing task using the robot shown on the left in Fig. 1. The robot consists of two Kuka light-weight robot arms, and two compliant five-fingered DLR hands [27]. We also evaluated the models on a lifting task using a PR2 robot as shown on the right in Fig. 1. The robot does not have tactile sensors or wrist-mounted force torque sensors. For both tasks, the robots' arms were controlled using Cartesian impedance control. The experiments evaluated how accurately the models predict the hand and object trajectories over multiple time steps. We also evaluated the models' accuracy when estimating the latent mass of a container based on an observed trajectory.

A. Pushing Experiment - Experimental Setup

For the pushing task, the robot moved its hand across the table in a straight line using an impedance controller with a proportional gain of 20N/m in the pushing direction. The desired trajectory moved at a constant speed of 2cm/s and samples were extracted at 1cm increments. A box was placed in the path of the robot's hand at the start of each pushing trial, such that the hand pushed the box across the table. The box weighs 100g and was tracked using a set of OptiTrack markers. Additional masses were added to the box in 500g increments, up to a maximum of 2kg, resulting in five different mass settings. The mass alters the amount of static friction between the box and the table and thus changes the mode transition conditions. For each of the five mass settings, the box was initially placed over a 48cm range at 2cm increments giving a total of 125 pushing trajectories, i.e., 25 trajectories per mass setting.

The observed state $s_t \in \mathbb{R}^3$ included the box position, the hand position, and the desired hand postion along the pushing direction. For each mode, the models learned a linear Gaussian mapping from the box's mass and the constant action to the change in the hand and box positions. The change in the desired hand position was excluded, as it is known. The STARHMMs used three features for the mode transition distributions: the distance between the box and the hand, the distance between the hand and the desired hand position, and the object's mass.

The trajectory prediction and mass estimation evaluations were performed using five-fold-cross validation. Each test set contained 5 samples per mass setting with the initial box positions spaced 10cm part. The training sets were then sampled from the remaining 100 samples with equal numbers of samples per mass setting, and samples with the same initial box positions across the mass settings.

We evaluated varying the number of training samples and modes κ. The standard values for the fixed parameters were 15 training samples and $\kappa = 3$ modes. The results for the trajectory prediction and the mass estimation are shown in Fig. 3 and Fig. 4 respectively. The prediction errors for different prediction steps is shown on the right of Fig. 3. The plot on the right of Fig. 4. shows the mass estimation performance when excluding the mass m from the observed state transition distribution $p(s_{t+1}|s_t, a_t, \rho_t)$.

B. Pushing Experiment - Discussions

The STARHMM-STD achieved the best performance when using $\kappa = 3$ modes. These modes tend to correspond to the approaching, loading, and sliding modes, as shown in Fig. 5, that one would expect for this task. The performance tends to decrease when using more modes. For the STARHMM-EE, $\kappa = 6$ modes resulted in the best performance values. However, the performance for this model tends to plateau between $\kappa = 3$ and $\kappa = 7$. Based on the marginal probabilities $p(\rho_t)$, we computed the proportion of samples assigned to each mode for the $\kappa = 7$ STARHMM-EE. Even though the model includes seven modes, 85.5% of the samples are assigned to only three of the modes. The model is therefore effectively only using three modes.

When using more than $\kappa = 3$ modes, the STARHMM-STD's error increase, while the performance of the STARHMM-EE stays level. This difference suggests that the STARHMM-EE's mode transition model $p^{\text{EE}}(\rho_t|s_t, \rho_{t-1}, m)$ is better at handling superfluous modes in the model. The

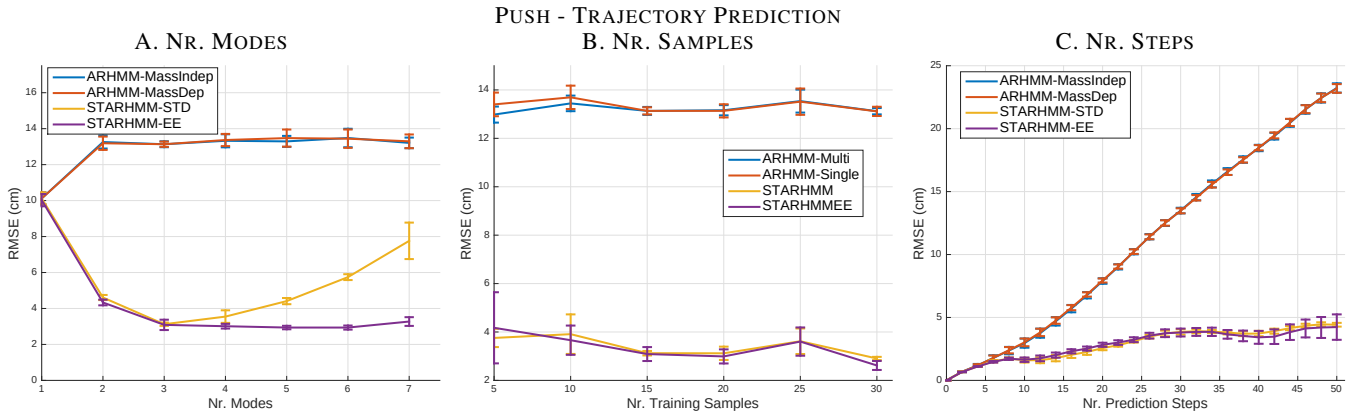

Fig. 3. The plots show the trajectory prediction performance of the four models when using different numbers of (left) modes κ and (middle) training samples. The plot on the right shows the error over different prediction horizons. The error bars indicate $+/-$ two standard errors.

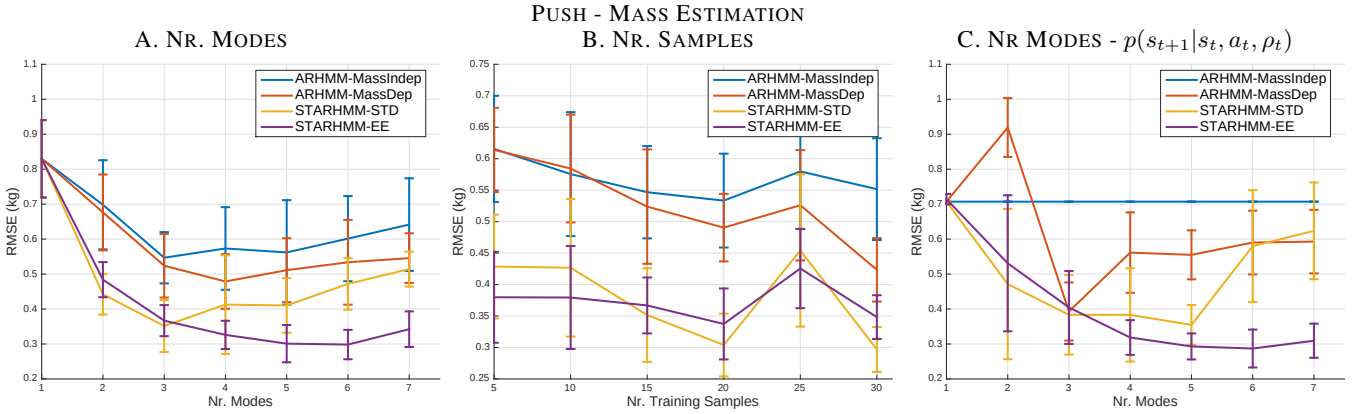

Fig. 4. The plots show the mass estimation performance of the four models when using different numbers of (left) modes κ and (middle) training samples. The plot on the right shows the errors when excluding the mass from the observed state transitions. The error bars indicate $+/-$ two standard errors.

STARHMM-STD's model is more complicated, as it learns a separate mapping from each mode. By contrast, the STARHMM-EE shares the initiation distribution $p(\rho_{t+1} = j|\rho_t, s_{t+1}, m, \varepsilon_t = 1)$ between all of the modes and is therefore easier to learn.

The extra modes in the models tend to capture either outlier samples or redundant modes, i.e., multiple modes that have similar observed state transition distributions $p(s_{t+1}|s_t, a_t, \rho_t, m)$. For the outlier samples, the model transitions to the extra mode for one or two samples before switching back to one of the main three modes. These outliers may be due to variations in the action executions or the friction between the box and the table. The outlier samples' modes and the redundant modes are visually similar the modes in Fig. 5.

The results of excluding the mass from the observed state transitions $p(s_{t+1}|s_t, a_t, \rho_t)$ are shown in Fig. 4C. The difference in the ARHMM-MassIndep's performance indicates that including the mass in the transition distribution does provide some additional information. The overall performances of the STARHMMs are similar between the two transition models. The $\kappa = 3$ ARHMM-MassDep performs better when excluding the mass. The $\kappa = 2$ ARHMM-MassDep performs worse as it tends to capture only the first mode switch, which does not depend on the mass.

Both of the ARHMMs performed poorly for predicting the hand and object trajectories. As the mode transitions of the ARHMMs rely mainly on the previous mode, the models will often predict a mode transition before the hand has made contact with the object or when the hand is already inside of the object. The STARHMMs tend to avoid these mistakes by incorporating the observed state into the mode transitions. This result shows the importance of capturing the mode transition conditions for predicting trajectories.

The ARHMMs performed worse than the STARHMMs when estimating the mass of the box. The ARHMM-MassDep achieved an RMSE of 0.479kg, while the STARHMM-EE achieved an RMSE of 0.298kg. These mass estimation errors may seem large. However, through empirical evaluations we estimated the coefficient of static friction between the table and box to be 0.274. A mass difference of 0.5kg therefore corresponds to only 1.34N of horizontal pushing force. The robot could potentially further increase the accuracy of the mass estimate by incorporating dynamic tactile sensing to determine precisely when the mode switch occurs [28], [29].

The evaluated models assume a constant coefficient of static friction between the table and the box. The learned models would therefore over estimate the weight of the box if the table were given a high-friction rubber coating. If the robot has an

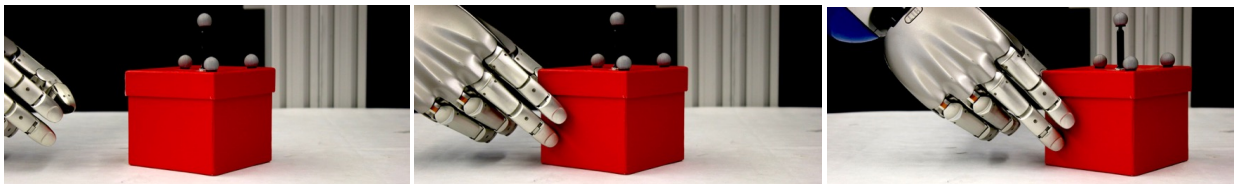

Fig. 5. The figure shows the three modes of the pushing task. The three modes correspond to the hand approaching the object, loading up the force on the stationary object, and sliding the object over the table. The transition from the second to the third mode depends on the box's mass.

estimate of the friction coefficient, then it can use the product of the coefficient and the mass as the input to the model. If the coefficient and mass are both varying between trails and unknown, then the robot can only recover their product and not the mass. Another limitation of the models is that they assume constant gains for the impedance controller. Higher gains would result in the second mode switch occurring earlier and with a smaller distance between the hand and desired hand positions. The robot would therefore underestimate the mass of the box. This problem could be alleviated by incorporating an explicit force estimate into the mode transition features ϕ_t for the STARHMMs. The models currently also employ linear transition models for the individual modes. In the future, we will explore using Gaussian processes, with mode-specific hyperparameters, to model the observed state transitions.

C. Lifting Experiment - Experimental Setup

In this experiment, the PR2 robot raised its hand 30cm, in 1cm increments, to lift a box as shown in Fig. 1. The robot used an impedance controller with a proportional gain of 80N/m for this task. The box has a mass of 0.114kg and we added additional weights at 0.206kg increments up to 0.938kg, resulting in five mass settings. The desired trajectory always started at the table height. However, to vary the trials, we placed the object on the table or on stacks of books that increased the height by 3.5cm, 6.0cm, or 9.5cm, resulting in four height settings. The robot executed the lifting skill three times for each of the 20 combinations of mass and height settings, resulting in a total of 60 samples.

The observed state s_t includes the hand position, and the desired hand position. The mode transition distributions for the STARHMMs used two features: the difference between the observed and desired hand positions, and the mass of the object. Given the impedance controller, the first feature provides the robot with an estimate of the applied force.

The evaluations were performed using four-fold cross validation, such that each test set contains the 15 samples for one height setting. The robot sampled 15 training samples from the remaining 45 samples, with three training samples per mass setting. The trajectory prediction and mass estimation results for different numbers of modes κ are shown in Fig. 6.

As an additional evaluation of the STARHMMS, we collected an additional five samples for each height setting. The masses for these five samples are 0.103kg more than for the original five mass settings. Given these additional samples, the test sets now contain 20 samples and the robot evaluates 10 mass settings. The training sets for this experiment are the same as for the main evaluation and only include samples from the original five mass settings. The results for this evaluation are shown in Fig. 6 and indicated by Mass10.

D. Lifting Experiment - Discussions

The prediction and estimation errors drop for the STARHMMs when the number of modes is increased from $\kappa = 1$ to $\kappa = 2$, as the models can then capture both the loading and the lifting modes of the task shown in Fig. 6C. The STARHMMs' prediction performances are slightly better when using a third mode $\kappa = 3$, which tends to correspond to the first one or two samples after liftoff that exhibit slightly shorter translations. Including multiple modes again decreases the prediction performance for the ARHMMs, as they do not capture the mode switching conditions.

The mass estimates are overall more accurate for the lifting task than the pushing task, with the STARHMMs correctly estimating all of the masses of the test trajectories when using $\kappa = 2$ or $\kappa = 3$ modes. Friction is inherently difficult to model, which results in coarser estimates for the pushing task. Given a firm grasp, the mode switch in the lifting task allows the robot to estimate the mass of the object independent of the coefficient of friction between the object and the table.

The increased performance of the STARHMMs over the ARHMM-MassDep can be partially attributed to the fact that the STARHMMs interpolate between the mass values. For example, training trajectories for the 0.114kg box and the 0.938kg box also provide information about the mode switching conditions for the 0.526kg box. By contrast, the ARHMM-MassDep learns a separate mode transition model for each mass setting and no information is shared between these five transition models.

The evaluations with the 10 mass settings show that the STARHMMs can interpolate between mass values. There is a small drop in performance for estimating the object's mass, but the difference in the prediction performance is negligible.

V. Conclusion

We explored the use of four different autoregressive hidden Markov models for representing mode transitions in manipulation tasks. The models were used to capture the dependency of mode switches, such as slipping and breaking contact, on the mass of the manipulated objects. Using these models, the robot predicted the trajectories of the manipulated object given its mass. The robot also used the models to estimate the latent mass of a container based on the observed mode switches.

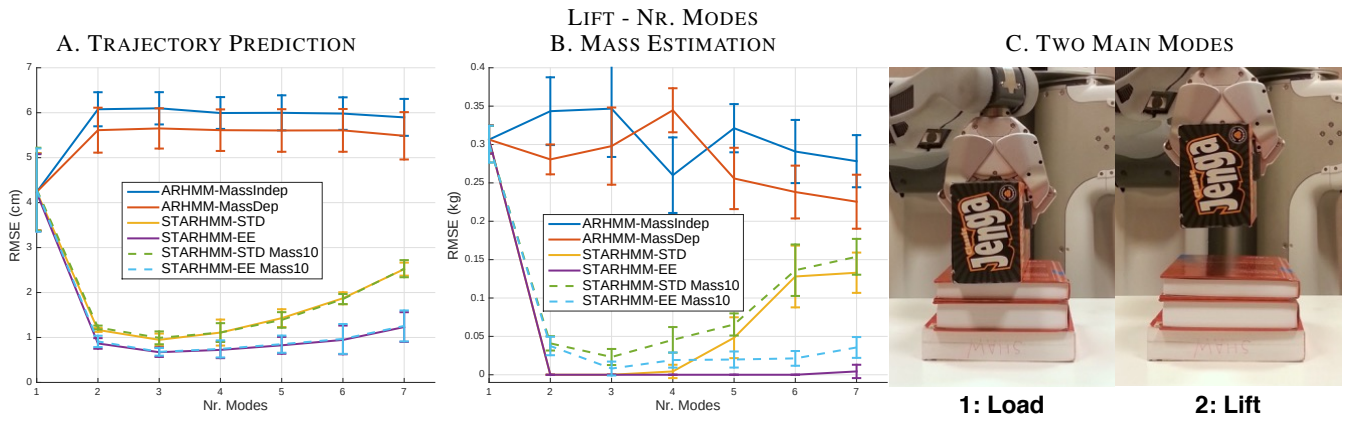

Fig. 6. The plots show (left) the trajectory prediction performance and (middle) the mass estimation performance for different numbers of modes κ. The error bars indicate $+/-$ two standard errors. The plot on the right shows the two main modes for the box lifting task: loading and lifting.

The evaluations on the pushing and lifting tasks showed that models that interpolate between different mass values achieved better prediction and estimation performances. Including the observed states in the mode transition distributions also improved the prediction and estimation accuracies. The best performances were achieved by the STARHMM-EE model, which additionally shares transition information between modes.

In the future, we will explore using the mass estimates from the initial pushing and lifting skills to adapt and predict the effects of subsequent mass-dependent manipulation skills. We will also explore estimating object masses based on other types of multi-modal manipulation skills using the STARHMMs.

References

[1] K. Hauser and V. Ng-Thow-Hing, "Randomized multi-modal motion planning for a humanoid robot manipulation task." *International Journal of Robotic Research*, vol. 30, no. 6, pp. 678–698, 2011.

[2] J. Barry, L. P. Kaelbling, and T. Lozano-Pérez, "A hierarchical approach to manipulation with diverse actions," in *IEEE International Conference on Robotics and Automation*, 2013.

[3] R. S. Johansson and R. J. Flanagan, "Coding and use of tactile signals from the fingertips in object manipulation tasks." *Nature Reviews Neuroscience*, vol. 10, no. 5, pp. 345–359, 2009.

[4] O. Kroemer, C. Daniel, G. Neumann, H. van Hoof, and J. Peters, "Towards learning hierarchical skills for multi-phase manipulation tasks," in *IEEE International Conference on Robotics and Automation*, 2015.

[5] J. Romano, K. Hsiao, G. Niemeyer, S. Chitta, and K. J. Kuchenbecker, "Human-inspired robotic grasp control with tactile sensing," *IEEE Transactions on Robotics*, vol. 27, pp. 1067–1079, 2011.

[6] M. Koval, N. Pollard, and S. Srinivasa, "Pre- and post-contact policy decomposition for planar contact manipulation under uncertainty," in *Robotics: Science and Systems*, 2014.

[7] H. van Hoof, O. Kroemer, and J. Peters, "Probabilistic segmentation and targeted exploration of objects in cluttered environments," *IEEE Transactions on Robotics*, 2014.

[8] D. Katz, Y. Pyuro, and O. Brock, "Learning to manipulate articulated objects in unstructured environments using a grounded relational representation," in *Robotics: Science and Systems*, 2008.

[9] K. Hausman, S. Niekum, S. Osentoski, and G. Sukhatme, "Active articulation model estimation through interactive perception," in *IEEE International Conference on Robotics and Automation*, 2015.

[10] H. P. Saal, J.-A. Ting, and S. Vijayakumar, "Active estimation of object dynamics parameters with tactile sensors," in *IEEE International Conference on Intelligent Robots and Systems*, 2010.

[11] S. Chitta, J. Sturm, M. Piccoli, and W. Burgard, "Tactile sensing for mobile manipulation," *IEEE Transactions on Robotics*, vol. 27, no. 3, pp. 558–568, 2011.

[12] J. Sinapov, C. Schenck, and A. Stoytchev, "Learning relational object categories using behavioral exploration and multimodal perception," in *IEEE International Conference on Robotics and Automation*, 2014.

[13] J. Sinapov, P. Khante, M. Svetlik, and P. Stone, "Learning to order objects using haptic and proprioceptive exploratory behaviors," in *International Joint Conference on Artificial Intelligence*, 2016.

[14] S. Niekum, S. Osentoski, G. D. Konidaris, S. Chitta, B. Marthi, and A. G. Barto, "Learning grounded finite-state representations from unstructured demonstrations," *International Journal of Robotics Research*, vol. 34, no. 2, 2015.

[15] D. Kulic, C. Ott, D. Lee, J. Ishikawa, and Y. Nakamura, "Incremental learning of full body motion primitives and their sequencing through human motion observation," *International Journal of Robotics Research*, 2011.

[16] M. Patel, J. V. Miro, D. Kragic, C. H. Ek, and G. Dissanayake, "Learning object, grasping and manipulation activities using hierarchical hmms," *Autonomous Robots*, vol. 37, no. 3, pp. 317–331, 2014.

[17] L. Rozo, P. Jiménez, and C. Torras, "A robot learning from demonstration framework to perform force-based manipulation tasks," *Intelligent Service Robotics*, vol. 6, no. 1, pp. 33–51.

[18] A. Christiansen, M. T. Mason, and T. Mitchell, "Learning reliable manipulation strategies without initial physical models," in *IEEE International Conference on Robotics and Automation*, 1990.

[19] C. G. Atkeson, C. H. An, and J. M. Hollerbach, "Rigid body load identification for manipulators," in *IEEE Conference on Decision and Control*, 1985.

[20] D. Nguyen-Tuong and J. Peters, "Model learning in robotics: a survey," *Cognitive Processing*, vol. 12, no. 4, pp. 319–340, 2011.

[21] J. Wu, I. Yildirim, J. J. Lim, B. Freeman, and J. Tenenbaum, "Galileo: Perceiving physical object properties by integrating a physics engine with deep learning," in *Advances in Neural Information Processing Systems*, 2015.

[22] C. Finn, I. Goodfellow, and S. Levine, "Unsupervised learning for physical interaction through video prediction," in *Advances in Neural Information Processing Systems*, 2016.

[23] L. E. Baum, "An equality and associated maximization technique in statistical estimation for probabilistic functions of Markov processes," *Inequalities*, vol. 3, pp. 1–8, 1972.

[24] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

[25] O. Kroemer, H. van Hoof, G. Neumann, and J. Peters, "Learning to predict phases of manipulation tasks as hidden states," in *IEEE International Conference on Robotics and Automation*, 2014.

[26] F. R. K. Chung, *Spectral Graph Theory*. American Mathematical Society, 1997.

[27] Z. Chen, N. Y. Lii, T. Wimboeck, S. Fan, M. Jin, C. Borst, and H. Liu, "Experimental study on impedance control for the five-finger dexterous robot hand DLR-HIT ii." in *IEEE International Conference on Intelligent Robots and Systems*, 2010.

[28] M. R. Tremblay and M. R. Cutkosky, "Estimating friction using incipient slip sensing during a manipulation task," in *IEEE International Conference on Robotics and Automation*, 1993.

[29] Z. Su, O. Kroemer, G. E. Loeb, G. S. Sukhatme, and S. Schaal, "Learning to switch between sensorimotor primitives using multimodal haptic signals," in *International Conference on Simulation of Adaptive Behavior*, 2016.