Multimodal Execution Monitoring for Anomaly Detection During Robot Manipulation

Daehyung Park*, Zackory Erickson, Tapomayukh Bhattacharjee, and Charles C. Kemp

***Abstract*— Online detection of anomalous execution can be valuable for robot manipulation, enabling robots to operate more safely, determine when a behavior is inappropriate, and otherwise exhibit more common sense. By using multiple complementary sensory modalities, robots could potentially detect a wider variety of anomalies, such as anomalous contact or a loud utterance by a human. However, task variability and the potential for false positives make online anomaly detection challenging, especially for long-duration manipulation behaviors. In this paper, we provide evidence for the value of multimodal execution monitoring and the use of a detection threshold that varies based on the progress of execution. Using a data-driven approach, we train an execution monitor that runs in parallel to a manipulation behavior. Like previous methods for anomaly detection, our method trains a hidden Markov model (HMM) using multimodal observations from non-anomalous executions. In contrast to prior work, our system also uses a detection threshold that changes based on the execution progress. We evaluated our approach with haptic, visual, auditory, and kinematic sensing during a variety of manipulation tasks performed by a PR2 robot. The tasks included pushing doors closed, operating switches, and assisting able-bodied participants with eating yogurt. In our evaluations, our anomaly detection method performed substantially better with multimodal monitoring than single modality monitoring. It also resulted in more desirable ROC curves when compared with other detection threshold methods from the literature, obtaining higher true positive rates for comparable false positive rates.**

I. Introduction

A common approach to robot manipulation is for the robot to execute sequences of stereotyped task-specific robot behaviors (see Fig. 1) [1], [2]. A robot can monitor this process using a separate system that runs in parallel, which is a form of execution monitoring system (an execution monitor) [3]. By monitoring multimodal sensory signals relevant to manipulation, an execution monitor could perform a variety of roles, including detecting success or deciding to switch behaviors. In this paper, we focus on the problem of using an execution monitor to detect when the sensory signals associated with the execution of a manipulation behavior are anomalous. More specifically, the execution monitor should detect when the current sensory signals differ significantly from past sensory signals associated with task success. This is analogous to the conventional problem of finding unexpected patterns in data, called anomaly detection. Anomaly detection has been successfully applied to a variety of real-world problems, including credit-card fraud detection, cyber-intrusion detection, and error detection [4].

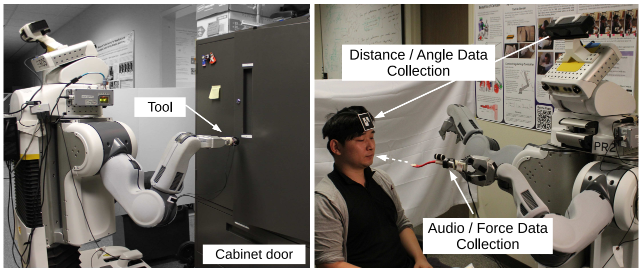

Tool
Cabinet door
Distance / Angle Data Collection
PR2
Audio / Force Data Collection

Fig. 1: Our system enables a PR2 to detect anomalies based on multimodal sensing while performing these and other tasks. **Left:** Pushing a cabinet door closed. **Right:** Assistive yogurt feeding with an able-bodied participant.

While suitable for a narrower set of situations, stereotyped task-specific behaviors tend to have lower variability in their operation than more general methods. This can reduce the variation in the associated multimodal signals, thereby lowering data requirements for data-driven methods and simplifying anomaly detection [1], [5]. However, due to competing performance criteria and the complexities of real-world manipulation, anomaly detection remains challenging. Ideally, an execution monitor would detect anomalies online, alert the robot shortly after the onset of an anomaly, work for long-duration behaviors, detect subtle anomalies, ignore irrelevant task variation, handle multimodal sensory signals, and detect crossmodal anomalies that would not be evident when monitoring modalities independently. In this paper, we present our method for anomaly detection in an effort to address these considerations.

Our method consists of training hidden Markov models (HMMs) using multimodal sensory signals recorded during non-anomalous executions [6], [7]. For a particular HMM, all signals come from executions of a specific robot behavior (e.g., pushing or feeding) applied to a specific task (e.g., closing a door or feeding yogurt) performed with specific objects (e.g., a particular microwave oven or a particular person).[1] At run time, an HMM provides likelihood estimates, which our system compares to a detection threshold that is based on a probabilistic representation of execution progress. If at any time the log-likelihood is below the current detection threshold, our system detects an anomaly.

We evaluated our method's ability to detect real anomalies

D. Park, Z. Erickson, T. Bhattacharjee, and C. C. Kemp are with Healthcare Robotics Lab, Georgia Institute of Technology, Atlanta, GA. *D. Park is the corresponding author deric.park@gatech.edu.

[1] Our approach could potentially generalize to categories of objects, but for this paper we only consider specific objects with which the robot has already had experience.

using haptic and auditory signals while a PR2 robot performed pushing tasks, such as closing a microwave oven, operating a light switch, and depressing a toaster handle. We also evaluated our method with a PR2 robot that assisted able-bodied participants with eating yogurt. While providing assistance, the robot recorded haptic signals, auditory signals, and visually-obtained kinematic estimates. Robotic assistance for people with disabilities during tasks such as feeding and other activities of daily living (ADLs) serves as a motivation for our work [8]. Multimodal anomaly detection could potentially enable an assistive robot to detect a variety of issues, such as undesirable collisions, hardware failures, and emphatic utterances by the user. More generally, anomaly detection might enable assistive robots to operate more conservatively when in close proximity to a person with impairments. In our evaluations, our anomaly detection method performed substantially better with multimodal monitoring than with single modality (unimodal) monitoring. It also resulted in more desirable receiver operating characteristic (ROC) curves when compared with other detection threshold methods from the literature.

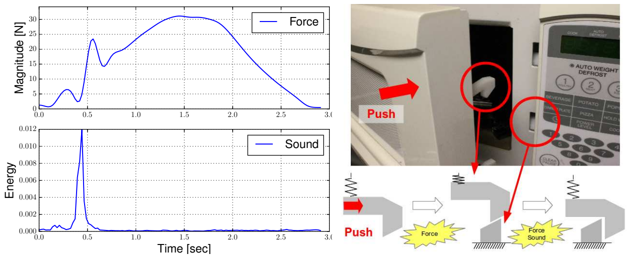
II. Related Work

Researchers have found that distinct human sensory modalities can be closely coupled [9]. Inspired by this research, Fitzpatrick et al. introduced a crossmodal method that enabled a robot to learn relationships between visual and auditory signals while manipulating objects [10]. Wu and Siegel investigated the use of combining acceleration and sound measurements to detect structural defects in airplane components [11].

A number of researchers have used unimodal sensing to detect anomalies, including [12], [13], [14], and [15]. Researchers have also investigated multimodal anomaly detection during robotic manipulation by directly representing sensor signals with respect to time without modeling state-based dynamics [16]. For example, Pastor et al. used multimodal sensing to predict failure while a robot attempted to flip a box using chopsticks. Their method predicted a failure when for 3 consecutive time steps, 5 or more signals failed independent z-tests with respect to signal recordings from successful trials indexed by time [17].

Jain and Kemp used object-centric task-specific state-based representations of applied forces during manipulation for anomaly detection. They investigated a task for which a quasistatic model was appropriate and did not consider signals beyond forces and kinematics [5]. To represent more complex dynamics, our method uses a multivariate HMM. HMMs have been used in a variety of approaches for novelty detection and anomaly detection [18], [4]. Most researchers have used the likelihood of current observations for detection, often with respect to a fixed threshold [19], [20], [21], [22].

Outside of robotics, researchers have used alternative thresholds on likelihood estimates from HMMs for anomaly detection. Ocak et al.'s system reported anomalies when either the likelihood exceeded a fixed threshold or the change in the likelihood between time steps exceeded another fixed threshold [23]. Yeung and Ding used a likelihood threshold that varied based on the current observations [24]. Outside of robotics, researchers have also used the discrete probability distribution over hidden states, which we use to represent execution progress [25], [26].



(a) Force and sound sequences

(b) Locking mechanism

Fig. 2: (a) The graphs show the force magnitude and sound energy while pushing a microwave door closed. (b) The microwave's latch mechanism results in related forces and sounds.

Kappler et al.'s recently published method uses multimodal sensing, including sensed forces, audio, and kinematics, to detect failures during robot manipulation [2]. Unlike an HMM, which models state transitions probabilistically, their method assumes that the current state of execution can be determined based on the current multimodal sensor readings alone. Each state then classifies failures based on supervised discriminative learning from positive and negative examples. They did not provide an evaluation of their method with respect to true positive and false positive rates.

III. HMM for Multimodal Execution Monitoring

In this paper, we consider haptic, auditory, visual, and kinematic sensory signals for execution monitoring. Fig. 2 illustrates how force and sound can be closely related signals during common manipulation tasks. When the PR2 robot pushes the microwave door closed, the door's latch goes through various states with associated forces and sounds. After the latch makes contact, the magnitude of the force begins to go up. When the latch moves far enough, it springs down resulting in a loud sound, reduced force, and the door being secured. For a fixed duration of time, the robot continues to push and then pulls back, resulting in increasing then decreasing force, but no loud sounds. At any point in this process, an anomaly can result in detectable changes in the force, the sound, or both.

To model sensory signals such as this, we use a multivariate left-to-right HMM. Let a random variable $\mathbf{x}_i$ be a four-dimensional observation vector at time step i. A random variable $\mathbf{z}_i^j$ is the jth hidden state out of n different hidden states at time step i. Fig. 3 depicts the architecture of the left-to-right HMM, which requires that the state index for any path remain constant or increase over time. The figure also shows two possible hidden state paths (blue and red) associated with a time series of multidimensional observations. The transition probability $P(\mathbf{z}_{i+1}|\mathbf{z}_i)$ is the probability of

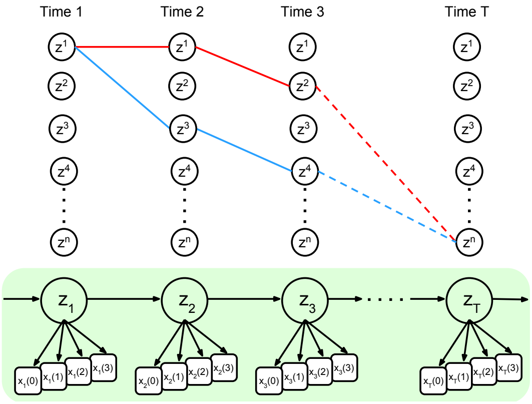


Fig. 3: Architecture of a left-to-right hidden Markov model with multivariate Gaussian emissions.

transitioning from one hidden state $\mathbf{z}_i$ to another $\mathbf{z}_{i+1}$. The emission probability $P(\mathbf{x}_i|\mathbf{z}_i)$ is the probability of output $\mathbf{x}_i$ given a hidden state $\mathbf{z}_i$. To represent correlations among modalities, we use a multivariate Gaussian distribution with a full covariance matrix for the emission probability.

For each behavior, we used the left-to-right HMM architecture with a single Gaussian distribution model implemented in the General Hidden Markov Model library (GHMM) (http://www.ghmm.org/). To train the model using the Baum-Welch algorithm, we initialized the initial state distribution, π; the transition probability matrix, $\mathbf{A}$; the emission matrix, $\mathbf{B}$; and the number of hidden states, n. We set $\mathbf{A}$ to be an upper triangular matrix with linearly decreasing transition probabilities from 0.4 to 0.0. We set the first element of the n-dimensional vector π to 1.0 and all other elements to zero in order to start the HMM in a particular state.

IV. Anomaly Detection

Our execution monitoring system uses an HMM to model non-anomalous execution of a manipulation behavior performing a task with particular objects. We use λ to represent the parameters of the trained HMM. Our execution monitor performs anomaly detection by comparing the log-likelihood, $\log P(X|\lambda)$, of the observations, X, with a threshold, $\tau(\gamma)$, that depends on the estimated execution progress, γ. At any time during the execution, if $\log P(X|\lambda) < \tau(\gamma)$, then the execution monitor detects an anomaly, otherwise it considers the execution to be non-anomalous. Our system represents execution progress, γ, using the probability mass function over hidden states given the current observations, $\gamma = P(\mathbf{z}_t|X,\lambda)$. Our system trains a mapping from the execution progress, γ, to the threshold, $\tau(\gamma)$. For our implementation, $\tau(\gamma) = \mu - c\sigma$, where μ and σ are the estimated mean and standard deviation of the log-likelihood given non-anomalous execution with progress γ. c is a constant used to adjust the sensitivity of the detector. Increasing c will tend to result in a lower false-positive rate and a lower true-positive rate.

A. Representing Execution Progress, γ

Even during non-anomalous executions, likelihood tends to vary significantly with the number of observations, which reduces the effectiveness of a constant detection threshold. In practice, during non-anomalous executions, the likelihood tends to vary in consistent ways. In order to model this variation in the likelihood, we use a representation of execution progress, γ. As addressed in the literature, the likelihood for an HMM can be expressed as a sum of joint distributions [27],

$$P(\mathbf{X}|\lambda) = \sum_{\mathbf{Z}} P(\mathbf{X},\mathbf{Z}|\lambda), \tag{1}$$

where λ represents the parameters for the trained model and $\mathbf{Z}$ is a state path over hidden state space, $\mathbf{Z} = \{\mathbf{z}_1, ..., \mathbf{z}_t\}$. Our system uses a left-to-right model with time-series data, so the hidden states must be in non-decreasing order, such as $\mathbf{Z} = \{\mathbf{z}_1^1, \mathbf{z}_2^1, \mathbf{z}_3^2, \mathbf{z}_4^3, \mathbf{z}_5^4, ..., \mathbf{z}_t^n\}$. Also, the HMM always starts in the first hidden state, $\mathbf{z}^1$. As such, if the true states were known, their indices could represent the progress of the behavior. Compared to directly using time, this would have the advantage of handling variability in the timing of a behavior's execution.

However, since the true state path is hidden from the observer, our method uses a probabilistic representation. One approach would be to use the maximum likelihood state at any given moment, but this would neglect uncertainty. Instead, we represent execution progress using the probability distribution over hidden states (the hidden-state distribution) at time t, $\gamma(t) = P(\mathbf{z}_t|\mathbf{X},\lambda)$. We compute the n -dimensional vector $\gamma(t)$ with the forward and backward procedures of the EM algorithm [6],

$$\gamma(t) = \frac{\alpha(t)\cdot\beta(t)}{P(\mathbf{X}|\lambda)}, \tag{2}$$

where $\alpha(t) = P(\mathbf{X}(1:t), \mathbf{z}_t|\lambda)$, $\beta(t) = P(\mathbf{X}(t+1:T)|\mathbf{z}_t,\lambda)$, and T is the last time sample of $\mathbf{X}$.

B. Mapping Execution Progress, γ, to Log-Likelihood Predictions, $\hat{\mu}$ and $\hat{\sigma}$

Our system maps execution progress, γ, to a prediction of the log-likelihood, L, associated with successful execution. To create this mapping, we first generate data consisting of pairs of γ and L by applying the trained HMM to sensor signals from successful executions. Given this data, a number of algorithms could potentially provide useful mappings. In this paper, we use K clusters of execution progress vectors. Each cluster is a time-based soft cluster that represents execution progress vectors that occurred at similar times during execution (see Fig. 4). When monitoring execution, the system finds the cluster that best matches the current execution progress vector, $\gamma(t)$. It then uses that cluster's associated log-likelihood model to decide if the current log-likelihood, $L(t) = logP(\mathbf{x}_1, ..., \mathbf{x}_t|\lambda) = logP(\mathbf{X}_t|\lambda)$, is anomalous. Each cluster's log-likelihood model consists of an estimated mean and standard deviation, $\hat{\mu}$ and $\hat{\sigma}$.

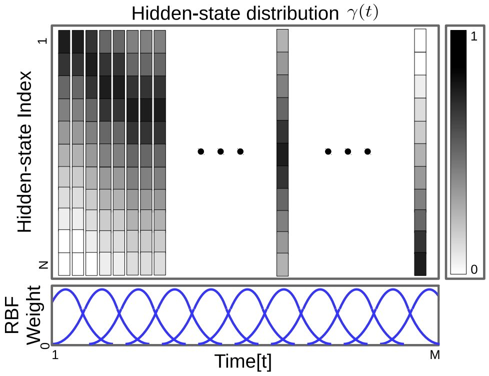


Fig. 4: Illustration of the K clusters of execution progress vectors (i.e., hidden-state distributions). Each cluster, k, has an associated RBF used to weight execution progress vectors, $\gamma(t)$, and their associated log-likelihoods, $L(t)$, based on when they occurred in time. These weights are used to compute $\hat{\gamma}_k$, $\hat{\mu}(L_k)$, and $\hat{\sigma}(L_k)$ for cluster k.

Each cluster has an associated Gaussian radial basis function (RBF) in time that weights the membership of execution progress vectors (see Fig. 4). The number of non-anomalous time series and the length of each time series in $\mathbf{X}_{train}$ are N and M, respectively. We use $k \in \{1, ..., K\}$ to denote the kth cluster and its associated RBF. The kth RBF is defined by the following function over time:

$$\phi(t, w_k) = e^{-\epsilon(t-w_k)^2},$$

where w_k is the center of the kth RBF and ϵ is a constant. Similar to receptive fields, $\phi(t, w_k)$ describes a weight where the kth RBF is active. In this work, we omit its normalization denominator since we use $\phi()$ as a weighting function. We also use evenly distributed RBFs such that $w_k = (M/K) \cdot (k-1)$ assuming stereotyped manipulation behaviors progress consistently. For each cluster, we compute a weighted average of execution progress vectors, $\hat{\gamma}_k$, using the following equation:

$$\hat{\gamma}_k = \frac{1}{N}\sum_{i=1}^{N}\Big[\frac{1}{\eta_k}\sum_{t=1}^{M}\gamma^{(i)}(t)\phi(t, w_k)\Big], \tag{3}$$

where η_k is a normalization factor, $\eta_k = \sum_{t=1}^{M}\phi(t, w_k)$, and $\gamma^{(i)}(t)$ denotes the execution progress vector at time t for time series i.

For each cluster, we then compute the weighted mean and variance of the associated log-likelihood using the following equations:

$$\begin{aligned}
\hat{\mu}(L_k) &= \frac{1}{N}\sum_{i=1}^{N}\Big[\frac{1}{\eta_k}\sum_{t=1}^{M}L^{(i)}(t)\phi(t, w_k)\Big],\\
\hat{\mu}(L_k^2) &= \frac{1}{N}\sum_{i=1}^{N}\Big[\frac{1}{\eta_k}\sum_{t=1}^{M}L^{(i)}(t)^2\phi(t, w_k)\Big],\\
\hat{\sigma}(L_k) &= \sqrt{\hat{\mu}(L_k^2) - (\hat{\mu}(L_k))^2}.
\end{aligned} \tag{4}$$

Using Equation (4) and (3), we can then represent the K clusters and their associated log-likelihood models as

$$\{(\hat{\gamma}_1, \hat{\mu}(L_1), \hat{\sigma}(L_1)), ..., (\hat{\gamma}_K, \hat{\mu}(L_K), \hat{\sigma}(L_K))\}. \tag{5}$$

C. Mapping Execution Progress, γ, to a Threshold, τ

Our method preprocesses the incoming data in the same manner as the training process and then computes both $\gamma(t)$ and the corresponding log-likelihood, $L(t) = logP(\mathbf{X}_{test}|\lambda)$, using Equation (2) and (1) respectively. The system detects an anomaly when the log-likelihood is lower than the execution progress dependent threshold, $\tau(\gamma(t)) = \hat{\mu}(L_{k^*}) - c\hat{\sigma}(L_{k^*})$, where c is a real-valued gain and k^* is the index of the best matching RBF. To find the best matching RBF, our system compares the cross-entropy between γ and each of the K RBFs using Kullback-Leibler divergence,

$$k^* = \underset{1,...,K}{\arg\min}\, D_{KL}(\gamma(t)||\gamma_k), \tag{6}$$

where $D_{KL}(P||Q)$ is a measure of the information lost when Q is used to approximate P. This approach also extends to online detection. Given an HMM and sensory signals, the detector can recursively compute $\gamma(t)$ and $logP(\mathbf{X}_t|\lambda)$ at each time step t, and then perform the following comparison:

$$\begin{aligned}
&\text{IF } logP(\mathbf{X}_t|\lambda) < \hat{\mu}(L_{k^*}) - c\hat{\sigma}(L_{k^*}), \text{ then } \textit{anomaly}\\
&\text{else } \textit{no anomaly}.
\end{aligned} \tag{7}$$

V. Evaluation with Two Manipulation Behaviors

We evaluated our approach with a pushing behavior and an assistive feeding behavior. The pushing behavior performed tasks such as closing doors and flipping light switches. The assistive feeding behavior brought spoonfuls of yogurt to the mouths of able-bodied participants (see Fig. 5). Prior to recruiting participants for our feeding behavior evaluation, we obtained approval for our study from the Georgia Tech Institutional Review Board (IRB).

We performed multiple cross-validation steps that divided all data into training and testing data sets. Training consisted of first fitting an HMM to the specific behavior and the particular object or human user. Using this HMM and non-anomalous training data, we then computed a mapping from execution progress to estimates for the mean and standard deviation of the log-likelihood.

We compared the performance of our system when using all available modalities versus using only a single modality. We also compared the performance of our time-varying likelihood threshold to two baseline methods from the literature: likelihood change detection [23] and fixed-threshold likelihood detection [19]. We report all of our results as receiver operating characteristic (ROC) curves in order to assess the tradeoff between false positive and true positive rates. To produce each ROC curve we varied a single parameter, the constant c, from our threshold function $\tau(\gamma)$.

A. Instrumentation for Multimodal Sensing

For all experiments, we used a PR2 robot from Willow Garage (see Fig. 1) which moved only a single arm during each trial. The PR2 is a 32-DOF mobile manipulator with

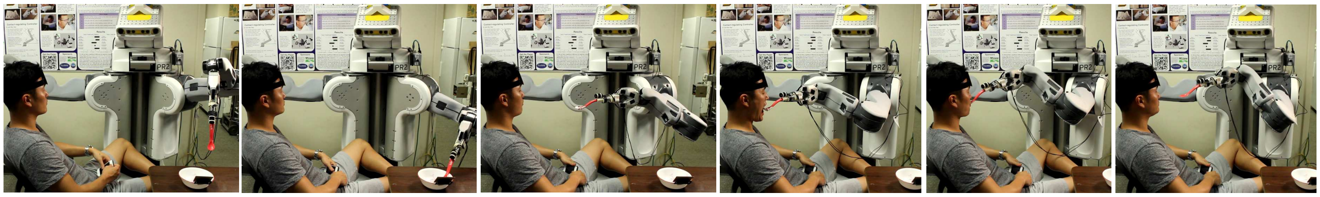


Fig. 5: Yogurt scooping followed by assistive feeding with an able-bodied participant. The PR2 uses an instrumented tool with a force-torque sensor and microphone. It estimates the pose of the bowl and the person's head using ARTags and a Microsoft Kinect v2.

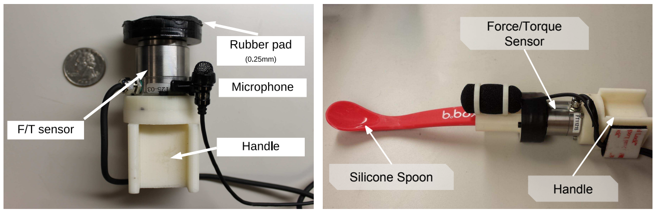


Fig. 6: Each instrumented tool has a force-torque sensor and microphone mounted on a 3D-printed handle. The handle is designed to be held by the PR2 gripper. **Left:** A tool for pushing that has a rubber-padded plastic circle. **Right:** A tool for feeding that has a flexible silicone spoon.

two 7-DOF back-drivable arms and powered grippers that are controlled by a 1 kHz low-level PID controller. Its maximum payload and grip force are listed as $1.8kg$ and $80N$, respectively.

For each behavior, the robot held a specialized instrumented tool with a 3D-printed handle designed for the PR2's grippers (see Fig. 6). The tools incorporate a force/torque sensor (ATI Nano25) and a unidirectional microphone in order to monitor haptic and auditory modalities during manipulation (see Fig. 6 **Left**). As the robot performed behaviors, our system recorded the 6-axis force/torque measurements at a 1 kHz sampling rate, and simultaneously recorded audio from the microphone at a 44.1 kHz sampling rate.

For the assistive feeding task, we also affixed an ARTag [28] to the person's head, so that the robot could use a Microsoft Kinect v2 to estimate and record the pose of the person's head. In addition, the robot recorded the pose of the spoon tool using forward kinematics.

B. Sensory Preprocessing

The force sequence is a time-series vector for which each element, denoted as f, represents the magnitude of a three-dimensional force vector. The sound sequence is a time-series vector for which each element, denoted as $\mathcal{E}$, represents the energy of an audio frame s. We use the "Yaafe audio features extraction toolbox" [29] to convert s into a numeric value for energy using the root mean square (RMS),

$$\mathcal{E} = \sqrt{\frac{\sum_{i=1}^{N_{frame}} (s(i)/I_{max})^2}{N_{frame}}}, \tag{8}$$

where N_{frame} is audio frame size 1,024 and I_{max} is 32,768—the maximum value of a 16-bit signed integer format. The lengths of these two sequences are different due to the differing sampling rates. Thus, while collecting training data, the force sequence was interpolated to match the length of the sound sequence. This process resulted in a sequence of tuples, $\{(f_1, \mathcal{E}_1), (f_2, \mathcal{E}_2), ...\}$, where f_i is the magnitude of the observed force and $\mathcal{E}_i$ is the energy of the observed sound at time i.

For the feeding task, we added two more object-centric kinematic modalities: distance and angle. For distance, the robot computed the Euclidean distance between the estimated position of the person's mouth and the silicone spoon. For angle, it found the angular difference between a unit vector pointed away from the robot's gripper along the length of the spoon and a unit vector pointed into the person's mouth. During training we downsampled or interpolated each modality to have 100 samples in time over the duration of the task.

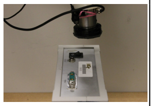
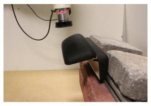
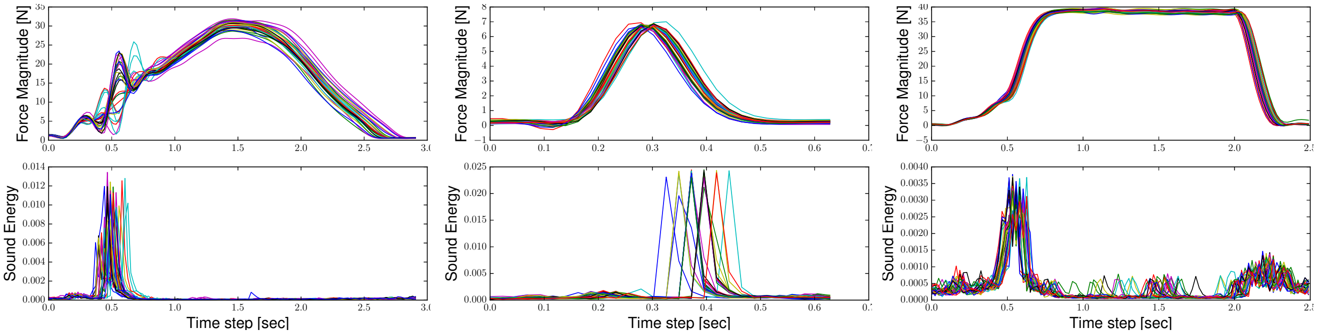
Microwave Door (30, 6)	**Microwave Door** (30, 5)	**Cabinet Door** (30, 10)	**Light Switch** (30, 10)
Device Switch  (31, 10)	**Outlet Switch** (30, 10)	**Toaster Switch** (33, 12)	**Diaper Case** (30, 10)
Wipe Case (31, 11)		**Glasses Case** (32, 9)	

TABLE I: This table shows the objects the robot pushed in our experiments. The PR2 pushed each object while recording haptic and auditory data. The numbers in parentheses represent the number of non-anomalous and anomalous trials we conducted with each object.

C. Pushing Tasks

We collected pushing task data from ten everyday objects, including a microwave and toaster (see Table I). The PR2



(a) Microwave (white)

(b) Light switch

(c) Toaster

Fig. 7: Visualization of the force and sound sequences recorded in three representative manipulation tasks: closing a microwave door, turning off a light switch, and turning on a toaster.

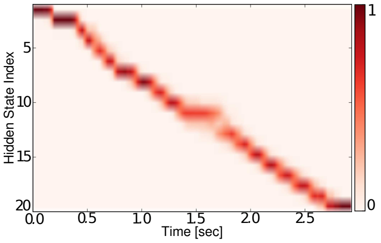


Fig. 8: Visualization of the execution progress vectors (i.e., hidden-state distributions) over time. This shows the average of all the vectors from the non-anomalous trials during which the robot pushed the white microwave closed. Each left-to-right HMM had 20 hidden states.

pushed each object with the instrumented tool and a predefined linear end effector trajectory for an object-specific amount of time and then pulled its end effector back for an object-specific amount of time. To produce anomalous events, we

- placed the tool at an incorrect location from which it could not properly contact the target mechanism,
- fixed the mechanism to prevent movement, or
- blocked the mechanism using an obstacle such as a metallic plate, a wooden stick, a rubber pad, a bundle of paper, a cable, a towel, a stapler, a roll of duct tape, a finger, or a screw.

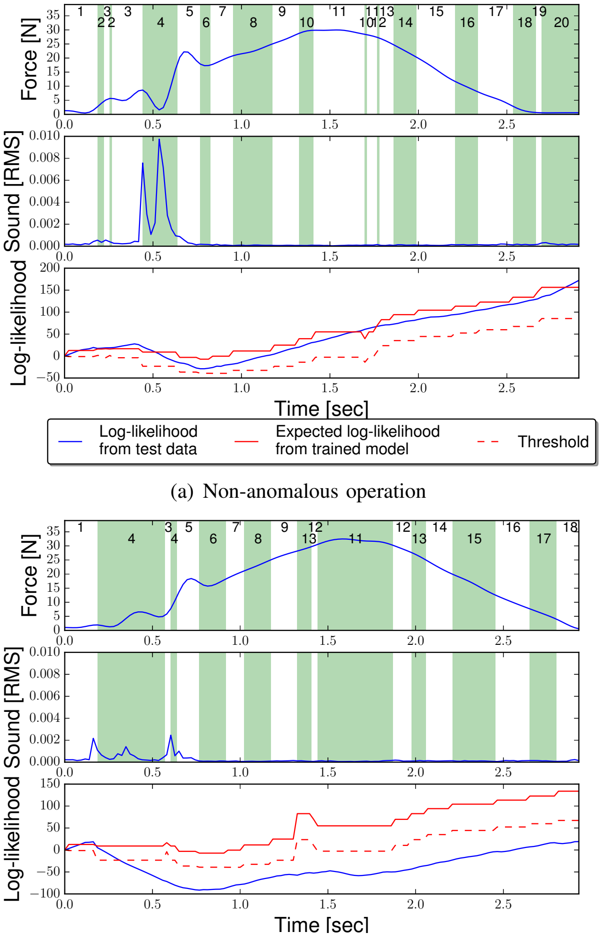
Fig. 7 provides a visualization of the force and sound data recorded for all non-anomalous executions with three different objects. The sensing modalities show consistent patterns over time for each of the three objects. As we previously described, the microwave's latching mechanism makes a sharp sound in conjunction with changes in the force. The sound associated with operating the light switch shows temporal variability, in part because of the shorter overall duration of the task (0.6 seconds) and preprocessing that included aligning the sensory data in time based on the recorded forces.

Fig. 8 provides a visualization of execution progress over time averaged across all non-anomalous trials for the white microwave closing task. Execution progress changes in an intuitive way with respect to time with the index of the most likely state progressively increasing.

Similar to k-fold cross-validation, we randomly split non-anomalous and anomalous data into k folds. A fold from both the non-anomalous and anomalous data were paired to form the test data, with the remaining $k-1$ folds of non-anomalous data used for training. We repeated this process k^2 times, so that each possible pair was used exactly once as test data. Note that we used $k=3$ in this pushing task. Depending on the length of the training sequences we used either 10 or 20 hidden states and the same number of RBFs.

Fig. 9 illustrates our system's operation during a non-anomalous and an anomalous trial of the white microwave closing task while using a constant detection threshold, c. For the non-anomalous execution, the mean log-likelihood based on execution progress (solid red curve) moves in conjunction with the log-likelihood resulting from the ongoing trial (solid blue curve). The standard deviation based on execution progress (related to the dashed red curve) tends to increase over time. For the anomalous execution, the behavior failed to generate a sharp sound at the appropriate time and instead generated a lower magnitude sharp sound early in the behavior's execution in conjunction with lower forces than anticipated. The log-likelihood went below the threshold early on in the execution, triggering the detection of an anomaly.

Results: Throughout our evaluation, the robot only used training data and testing data from the same object. We first compared the performance of our method using multimodality sensing versus unimodal sensing with force or audio alone. Fig. 10(a) **Left** shows ROC curves used to evaluate the relationship between the false positive rate (FPR) and true positive rate (TPR). For any given true-positive rate, multimodal sensing resulted in a lower false-positive rate when compared to unimodal sensing. Force sensing alone was better than audio alone, but using both resulted in better performance. Note that we used our method of time-varying thresholds for this comparison and obtained the ROC curves by varying the gain c in Equation (7). To evaluate the effectiveness of our method, we also compared it against two



(a) Non-anomalous operation

(b) Anomalous operation

Fig. 9: Comparison of non-anomalous and anomalous observations in the door-closing task for a microwave (white). The upper two graphs for (a) and (b) show the force and sound observations over time. Each white or green band denotes a period of time over which the most likely hidden state remained constant. The small black number in each band is the index for the most likely hidden state over the band's duration. These indices need not increase monotonically, since we computed them in an online fashion using only prior observations. The bottom graphs in (a) and (b) illustrate the mean log-likelihood based on execution progress (solid red curve), the log-likelihood resulting from the ongoing trial (solid blue curve), and the standard deviation based on execution progress (related to the dashed red curve). For this comparison, we set $c = 2.0$ and blocked the door using a rubber pad for the anomalous operation.

baseline methods. The change detection method determines anomalies when the decrease of the log-likelihood is larger than a given threshold. The fixed-threshold detection method detects an anomaly if the log-likelihood is lower than a constant threshold. The ROC curves in Fig. 10(a) **Right** show that for any given false-positive rate, our method had a higher true-positive rate than the two baseline methods.

D. Feeding Task

For our feeding behavior evaluation, we recruited 6 able-bodied participants, none of whom had prior experience with robotic feeding. For safety, the robot used low-impedance control. The robot also held a flexible silicone spoon designed for assistive applications (see Fig. 6 **Right**). We recorded 20 successful and 12 anomalous feeding attempts for each of the 6 able-bodied participants. To produce anomalous events, we

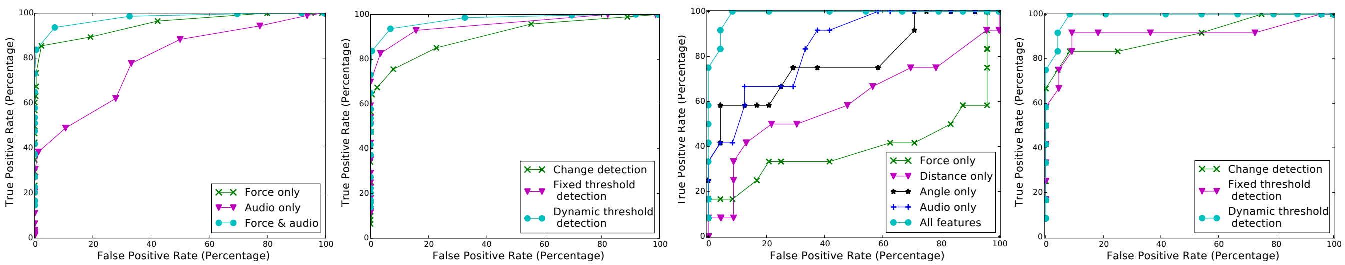
- added uniform-random noise to the detected mouth pose (i.e. position noise from 3 cm to 8 cm and angular noise from $-15°$ to $15°$),
- asked each subject to push any part of the spoon or PR2's arm during the feeding process,
- asked each subject to yell "stop" at any moment, and
- asked each subject to perform a random movement that prevents feeding, such as rotating their head or moving backwards.

To build our evaluation data set, we recorded each of these anomalous events 3 times for a total of 12 anomalous attempts per participant. To account for a low number of non-anomalous observations, we performed 6-fold cross-validation 6 consecutive times to improve stability and accuracy of results. Although we do not describe it in any detail, we also used our execution monitoring system to detect anomalies during the yogurt scooping behavior that precedes the feeding behavior.

Results: Throughout our evaluation, the robot only used training and testing data from the same user. We first compared the performance of our method using multimodal sensing versus unimodal sensing with force, distance, angle, or audio data alone. The ROC curves in Fig. 10(b) **Left** show that the use of multimodal sensing outperformed unimodal sensing. Interestingly, the force sensing alone performed relatively well in pushing tasks, but performed poorly on its own during assistive feeding. This may be due in part to the magnitude of the force fluctuating when the spoon is in a person's mouth. We compared our proposed method against the two baseline methods as depicted in Fig. 10(b) **Right**. The results were similar to the results from the pushing task with our method outperforming the two baseline methods.

VI. Conclusion

We introduced a new method for multimodal anomaly detection during robot manipulation. Our method uses a multimodal HMM to model the sensory readings associated with non-anomalous execution of a task-specific behavior. Since the likelihood for non-anomalous executions varies significantly over time, our method also learns a mapping from execution progress to non-anomalous log-likelihood. It uses this mapping to generate a time-varying likelihood threshold with which it detects anomalies. We evaluated our method with respect to object pushing and assistive yogurt feeding manipulation tasks. Multimodal anomaly detection outperformed unimodal anomaly detection. Our method also outperformed two baselines methods by providing higher true-positive rates at comparable false-positive rates.



(a) Pushing task

(b) Feeding task

Fig. 10: Receiver operating characteristic (ROC) curves for the pushing task and assistive feeding task. The left figures for (a) and (b) show ROC curves that compare the performance of multimodal and unimodal sensing for anomaly detection. The right figures for (a) and (b) compare the performance of our anomaly detection method versus two baseline methods.

Acknowledgment*: We thank Chris Birmingham, Hyder Hasnain, Youkeun Kim, Caleb Little, and Megan Rich for their assistance throughout this project. This work was supported in part by NSF Awards IIS-1150157, EFRI-1137229, and EEC-1263049, as well as NIDILRR grant 90RE5016-01-00 via RERC TechSAge.*

References

[1] A. Jain and C. C. Kemp, "El-e: an assistive mobile manipulator that autonomously fetches objects from flat surfaces," *Autonomous Robots*, vol. 28, no. 1, pp. 45–64, 2010.

[2] D. Kappler, P. Pastor, M. Kalakrishnan, M. Wuthrich, and S. Schaal, "Data-driven online decision making for autonomous manipulation," in *Proceedings of Robotics: Science and Systems*, July 2015.

[3] O. Pettersson, "Execution monitoring in robotics: A survey," *Robotics and Autonomous Systems*, vol. 53, no. 2, pp. 73–88, 2005.

[4] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection," *ACM Computing Surveys*, vol. 41, pp. 1–58, July 2009.

[5] A. Jain and C. C. Kemp, "Improving robot manipulation with data-driven object-centric models of everyday forces," *Autonomous Robots*, vol. 35, no. 2-3, pp. 143–159, 2013.

[6] L. R. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," in *PROCEEDINGS OF THE IEEE*, pp. 257–286, 1989.

[7] T. P. Minka, "From hidden markov models to linear dynamical systems," tech. rep., Tech. Rep. 531, Vision and Modeling Group of Media Lab, MIT, 1999.

[8] T. L. Chen, M. Ciocarlie, S. Cousins, P. M. Grice, K. Hawkins, K. Hsiao, C. C. Kemp, C.-H. King, D. A. Lazewatsky, H. Nguyen, *et al.*, "Robots for humanity: A case study in assistive mobile manipulation," 2013.

[9] K. Alho, T. Kujala, P. Paavilainen, H. Summala, and R. Näätänen, "Auditory processing in visual brain areas of the early blind: evidence from event-related potentials," *Electroencephalography and clinical neurophysiology*, vol. 86, no. 6, pp. 418–427, 1993.

[10] P. Fitzpatrick, A. Arsenio, and E. R. Torres-Jara, "Reinforcing robot perception of multi-modal events through repetition and redundancy and repetition and redundancy," *Interaction Studies*, vol. 7, no. 2, pp. 171–196, 2006.

[11] H. Wu and M. Siegel, "Correlation of accelerometer and microphone data in the coin tap test," *Instrumentation and Measurement, IEEE Transactions on*, vol. 49, no. 3, pp. 493–497, 2000.

[12] C. Piciarelli and G. L. Foresti, "On-line trajectory clustering for anomalous events detection," *Pattern Recognition Letters*, vol. 27, no. 15, pp. 1835–1842, 2006.

[13] A. Rodriguez, D. Bourne, M. Mason, G. F. Rossano, and J. Wang, "Failure detection in assembly: Force signature analysis," in *Automation Science and Engineering (CASE), 2010 IEEE Conference on*, pp. 210–215, IEEE, 2010.

[14] V. Sukhoy, V. Georgiev, T. Wegter, R. Sweidan, and A. Stoytchev, "Learning to slide a magnetic card through a card reader," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pp. 2398–2404, IEEE, 2012.

[15] O. Rosen and A. Medvedev, "An on-line algorithm for anomaly detection in trajectory data," in *American Control Conference (ACC), 2012*, pp. 1117–1122, IEEE, 2012.

[16] F. Marcolino and J. Wang, "Detecting anomalies in humanoid joint trajectories," in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pp. 2594–2599, IEEE, 2013.

[17] P. Pastor, M. Kalakrishnan, S. Chitta, E. Theodorou, and S. Schaal, "Skill learning and task outcome prediction for manipulation," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pp. 3828–3834, IEEE, 2011.

[18] M. Markou and S. Singh, "Novelty detection: a reviewpart 1: statistical approaches," *Signal processing*, vol. 83, no. 12, pp. 2481–2497, 2003.

[19] N. Vaswani, A. K. Roy-Chowdhury, and R. Chellappa, ""shape activity": a continuous-state hmm for moving/deforming shapes with application to abnormal activity detection," *Image Processing, IEEE Transactions on*, vol. 14, no. 10, pp. 1603–1616, 2005.

[20] M. S. Reddy, K. Nathwani, and R. M. Hegde, "Probabilistic detection methods for acoustic surveillance using audio histograms," *Circuits, Systems, and Signal Processing*, vol. 34, no. 6, pp. 1977–1992, 2014.

[21] A. G. Baghdasaryan *et al.*, "Automatic phoneme recognition with segmental hidden markov models," in *Signals, Systems and Computers (ASILOMAR), 2011 Conference Record of the Forty Fifth Asilomar Conference on*, pp. 569–574, IEEE, 2011.

[22] E. Di Lello, M. Klotzbucher, T. De Laet, and H. Bruyninckx, "Bayesian time-series models for continuous fault detection and recognition in industrial robotic tasks," in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pp. 5827–5833, IEEE, 2013.

[23] H. Ocak and K. A. Loparo, "Hmm-based fault detection and diagnosis scheme for rolling element bearings," *Journal of Vibration and Acoustics*, vol. 127, no. 4, pp. 299–306, 2005.

[24] D.-Y. Yeung and Y. Ding, "Host-based intrusion detection using dynamic and static behavioral models," *Pattern recognition*, vol. 36, no. 1, pp. 229–243, 2003.

[25] H. Hermansky, D. P. Ellis, and S. Sharma, "Tandem connectionist feature extraction for conventional hmm systems," in *Acoustics, Speech, and Signal Processing, 2000. ICASSP'00. Proceedings. 2000 IEEE International Conference on*, vol. 3, pp. 1635–1638, IEEE, 2000.

[26] G. Bernardis and H. Bourlard, "Improving posterior based confidence measures in hybrid hmm/ann speech recognition systems," in *Proceedings of International Conference on Spoken Language Processing (ICSLP" 98)*, no. EPFL-CONF-82494, pp. 775–778, 1998.

[27] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2006.

[28] M. Fiala, "Artag, a fiducial marker system using digital techniques," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 2, pp. 590–596, IEEE, 2005.

[29] B. Mathieu, S. Essid, T. Fillon, J. Prado, and G. Richard, "Yaafe, an easy to use and efficient audio feature extraction software.," in *ISMIR*, pp. 441–446, 2010.