Chapter 4
Nonparametric Bayesian Method for Robot Anomaly Monitoring

Abstract In this chapter, we introduce an anomaly monitoring pipeline using the Bayesian nonparametric hidden Markov models after the task representation and skill identification in previous chapter, which divided into three categories according to different thresholds definition, including (i) log-likelihood-based threshold, (ii) threshold based on the gradient of log-likelihood, and (iii) computing the threshold by mapping latent state to log-likelihood. Those method are effectively implement the anomaly monitoring during robot manipulation task. We also evaluate and analyse the performance and results for each method, respectively.

4.1 Introduction

Anomalies, also known as outliers, are observations that deviate too much from other observations to cause suspicion that they are caused by different mechanisms. Anomaly monitoring is a widely studied problem in machine learning, and has important significance in the fields of network intrusion monitoring, credit card fraud monitoring, and medical health monitoring. In robot manipulation, abnormality refers to abnormal behaviors, events, or situations. Robotic abnormality indicates that different from normal execution or observed data that is different from normal, especially in the dynamic environment of human-computer interaction, abnormalities usually stems from changes in the environment and human beings. Assuming that the robot can monitor, classify, and repair these anomalies, it will reduce or prevent the robot from being potentially injured and improve the safety of human-robot collaboration. Due to the diversity and complexity of anomaly types, it is not possible to directly enumerate and model all anomalies, and to implement anomaly monitoring through supervised learning. In general, research on robot abnormality monitoring is often implemented using unsupervised learning methods. In other words, it is abnormal to identify abnormal phenomena from the normal behavior model of the robot.
4.2 Related Work

The premise of performing anomaly monitoring is to endow robots with longer-term autonomy. To this end, we would like to discriminate between nominal and anomalous behavior while the robot executes a task. Researchers usually use two different methods to solve the problem of robot multimodal anomaly monitoring [1, 2]. On the one hand, a probability statistical model is established for each modal signal such as Rodriguez et al. [3] classify the success and failure of the robot in the assembly task by extracting the magnitude of the single-modal output force signal [4]. Ando et al. [5] used the K-Nearest Neighbor (KNN) algorithm to independently measure the differences between modalities to monitor the abnormal behavior of a mobile robot. Pastor et al. [6] used multi-modal sensing signals to monitor the failure of the robot to flip the box with chopsticks. This method first uses DMP to describe the motion of each stage, and then models each modality independently and combines normal statistical features (e.g. mean, variance, etc.) of the data for establishing ASM of each movement primitive to realize abnormality monitoring. The occurrence of abnormality is defined as if the signal value at least five times within three consecutive seconds is lower than a given threshold. Chu et al. [7] proposed an anomaly monitoring approach by training multiple HMM models for each modality given the observed values, and then the log-likelihood of each HMM was calculated as the feature vector for support vector machine model SVM (Support Vector Machine, SVM).

On the other hand, a probabilistic statistical model is established for all modality signals. Clifton et al. [8] used Gaussian Mixture Model (GMM) to model multi-modal signals, and realized anomaly monitoring through extreme value theory. Qiao et al. [9] proposed the use of a sequence of recessive states in the HMM model to monitor abnormal instructions. Zhang et al. [10] proposed a hierarchical HMM (Hierarchical HMM, HHMM) for system intrusion anomaly monitoring under the premise of the HMM method. This method uses HHMM to model normal data and is reflected by anomalies in hidden state sequences. Out the system intrusion event. Park et al. [11–13] carried out an online anomaly monitoring method based on multi-modal sensory signals fusion during robot manipulation, where a parametric HMM (Left-to-right HMM) models a multi-modal time series of a robot performing normal motions, and proposes abnormal thresholds that change by the execution progress to achieve abnormal monitoring. We extend the HMM for anomaly monitoring in the subsequent part of this chapter that inspired by the HMM cases. In addition, the proposed method uses a variety of complementary modal information to detect a variety of potential anomalies in the human-robot interaction process on the system where the robot feeds the disabled, such as human abnormal contact and environmental collision.

Hu et al. [14] proposed an abnormal event monitoring method based on HDP-HMM. This method combines the Fisher Kernel into the OCSVM (One-Class Support Vector Machine) model, which has the advantages of generating models and discriminative models. Dilello et al. [15, 16] first proposed the application of the non-parameterized Bayesian time series model (sHDP-HMM) for abnormal monitoring and fault diagnoses during robot operation, using the model to transmit force/torque
of the robot to perform normal alignment tasks [17, 18]. Sensing data is used for modeling, and the threshold value for abnormality monitoring is calculated from the validation data set. Then, the log-likelihood function value of the online observation data is calculated based on the learned model. Finally, the robot online is compared with the abnormality threshold value for anomaly monitoring.

In this chapter, we define an anomaly whenever a data stream is structurally different from prior learned models [19]. As Pimentel et al. review literature shows, many anomaly monitoring approaches have been proposed in the last decade [1]. However, few works focused on multivariate time series with spatial and temporal dependencies as in our case. In Milacski et al. [20], the method requires the full data sequence for processing, thus precluding the online detection capability. Besides our previous works [21, 22], others have used non-parametric Bayesian approaches to learn HMMs in contact tasks for abnormal detection and recognition. In [16, 23], DiLello et al. used the sticky hierarchical Dirichlet process prior to learn the model parameters of an HMM based on wrench signatures for an industrial robot alignment task. The work learned specific failure modes online. The approach is robust as it is models not just individual contact events but rather the evolution of signals, it does so while incorporating domain knowledge, and adjusting its model complexity according to the data patterns presented. Nonetheless, the approach is limited by the assumptions of the HMM in which observations are conditionally independent given the state. This is often an insufficient condition to capture the temporal dependencies of complex dynamical behaviors. In [24], Niekum et al. used a beta-process prior on the vector auto-regressive (BP-AR-HMM) to extract the state complexity from the data. The BP-AR-HMM has the ability to discover and model primitives from multiply related time-series. Robot skill discrimination was reliable and consistent; however, this work only modeled robot joint angles in a primarily non-contact oriented task. In [25], Hu et al. implemented anomaly monitoring based on HDP-HMM models and analyzed the efficiency and effectiveness of beam sampling in the HDP-HMM model. This implementation however suffers of low computational efficiency. Instead, we explore efficient online inference algorithms instead of the conventional sampling algorithms.

4.3 Problem Statement

Anomaly monitoring is used to continuously monitor the motion status of the robot to identify unexpected events during execution. Based on the recognition of robot motion in Chap. 3, this chapter describes in detail the abnormal event monitoring method for robots based on non-parametric Bayesian models. The proposed method is also applicable to the subsequent anomaly diagnoses. The main implementation ideas are: first, modeling the normally executed multi-modal data; then, test the normal movement behavior data based on this model. In order to obtain the definition of the boundary between normal and abnormal, that is, the calculation of the abnormal threshold; finally, monitoring the testing samples based on the model and the corre-
sponding threshold. In order to better study the theory and methods of multivariate time series anomaly monitoring, this chapter proposes the non-parametric Bayesian models HDP-HMM and sHDP-HMM in different observation models based on the parametric HMM model. An anomaly monitor implemented under the combination of modal information and different anomaly threshold methods. The anomaly decision of the proposed anomaly monitor on robot motion behavior includes: normal, single anomaly, and multiple anomalies.

An ideal online anomaly monitoring system often requires the use of multi-modal fusion methods to respond to different types of anomalous events, such as force/torque sensing information that can directly detect robotic collision anomalies and tactile sensing information that can directly sense objects. And the ability to respond to abnormal event occurrences in a timely manner (high response rate) and low false touch rate. In order to meet such requirements, this chapter addresses the problem of multi-modal fusion robot anomaly monitoring, and proposes the performance of anomaly monitors with three different probability models, different sensor information fusion, and different anomaly thresholds. The overview framework proposed after skill identification in Chap. 3 as shown in Fig. 4.1.

4.4 Log-Likelihood for Anomaly Monitoring

4.4.1 Problem Formulation

The robot introspection model uses non-parametric Bayesian priors along with a Hidden Markov model (HMM) and either Gaussian or Autoregressive emission models. First, HMMs are a doubly stochastic and generative process used to make inference
on temporal data. The underlying stochastic process contain a finite and fixed number of latent states or modes $z_t$, which generate observations $Y = \{y_1, \ldots, y_t\}$ through mode-specific emission distributions $b(y_t)$. These modes are not directly observable and represents sub-skills or actions in a given node of a task. Transition distributions, encoded in transition matrix $A_{ji}$, control the probability of transitioning across modes over time. The model assumes conditionally independent observations given the generating latent state. Given a set of observations, the Baum–Welch algorithm is used to infer model parameters $\Pi = (A, b)$. The fixed-modes assumption in HMMs limits the model’s expressive power as it is unable to derive natural groupings. Bayesian nonparametric priors extend HMM models to learn latent complexity from data [26, 27]. We use Fox’s et al. sticky-Hierarchical Dirichlet Process (sHDP) prior with an auto-regressive switching system [26] to model nominal skills as in our previous work [21] and derive more robust failure identification methods in manipulation tasks, specially in recovery scenarios.

Bayesian statistics are combined with the sHDP prior to both learn model complexity $k$ from the data but also to model the transition distribution of an HMM. The sticky parameter in the prior discourages fast-mode-switching otherwise present. Consider a set of training exemplars $t = \{x_1, \ldots, x_T\}$ of observed multi-modal data $\tau$ consisting of Cartesian pose and wrench values. Then, a mode-dependent matrix of regression coefficients $A^{(k)} = [A_1^{(k)} \ldots A_r^{(k)}] \in \mathbb{R}^{d \times (dr)}$ with an $r$th autoregressive order and $d$ dimensions is used along with a measurement noise $\Sigma$, with a symmetric positive-definite covariance matrix. The generative model for the sHDP-AR-HMM is summarized as:

$$G_0 = \sum_{k=1}^{\infty} \beta_k \delta_{\theta_k} \beta | \gamma \sim GEM(\gamma).$$
$$\theta_k | G_0 \sim G_0.$$  \hspace{1cm} (4.1)

$$G_j = \sum_{k=1}^{\infty} \pi_{jk} \delta_{\theta_k} \pi_j | \alpha, \beta \sim DP(\alpha, \beta).$$

The probability measure $G_j$, which models the transition distribution of the modes $\pi_j$ determines the weights (probabilities) of transitioning between modes $\delta_{\theta_k}$. To avoid fixing the mode complexity, $G_j$ uses a prior $G_0$ that is unbounded and can grow with the complexity of the data. While $G_j$ uses the same set of modes as $G_0$, $G_j$ introduces variations over those points. $G_0$ provides support for a possibly infinite space, but due to the Dirichlet’s process properties (the Chinese Restaurant Process), a finite set of modes is selected. In fact, we can understand the hierarchical specification as $G_j = DP(\alpha, G_0)$.

Different observation models can be included into the HMM. A Gaussian distribution with different covariance models (full, diagonal, and spherical) are considered. For Gaussian models, mode specific means and standard deviations are used $\theta_{zt} = \mathcal{N}(\mu, \sigma^2)$. Additionally, the sHDP-HMM can be used to learn VAR processes, which can model complex phenomena. The transition distribution is defined as in the
sHDP-HMM case, however, instead of independent observations, each mode now has conditionally linear dynamics, where the observations are a linear combination of the past \( r \) mode-dependent observations with additive white noise. A prior on the dynamic parameters \( \{A^{(k)}, \Sigma^{(k)}\} \) is necessary. A conjugate matrix-normal inverse Wishart (MNIW) was used to this end. The generative process for the resulting HDP-AR-HMM is then found in Eq. (4.2).

Observation Dynamics: \[
y_t = \sum_{i=1}^{r} A^{(z_t)}_i y_{t-i} + e_t(z_t).
\]

\( e_t \sim \mathcal{N}(0, \Sigma). \) (4.2)

Mode Dynamics: \[
z^{(i)}_t \sim \pi^{(i)}_{z_{t-1}}.
\]

By using the model over a set of multi-modal exemplar data, the sHDP-AR-HMM can discover and model shared behaviors in the data across exemplars. Scalable incremental or “memoized” variational coordinate ascent, with birth and merge moves \[27\] is used to learn the posterior distribution of the sHDP-HMM family of algorithms along with mean values for the model parameters \( \theta \) of a given skill, hence \( \theta_{s_m} = \{\Pi, A\}_{s_m}. \)

### 4.4.2 Skill Identification

The robot introspection system simultaneously detects nominal skills and anomaly events. Models are trained for individual skills to capture dynamics from multi-modal observations through vector \( \tau_m \). \( \tau_m \) consists of end-effector pose and wrench values. Scalable incremental or “memoized” variational coordinate ascent, with birth and merge moves \[27\] is used to learn the posterior distribution of the sHDP-HMM along with mean values for the model parameters \( \Pi \) of a given skill \( s \). Hence \( \Pi_s = \{\pi, A\} \): the transition matrix and regressor coefficients.

Given \( S \) trained models for \( M \) robot skills, scoring is used to compute the expected cumulative likelihood of a sequence of observations \( \mathbb{E}[\log P(Y|\Pi_s)] \) for each trained model \( s \in S \). Given a test trial \( r \), the cumulative log-likelihood is computed for test trial observations conditioned on all available trained skill model parameters \( \log P(y_1:y_r|\Pi)s \) at a rate of 200Hz (see Fig. 4.2 for an illustration). In Fig. 4.2, note the 4 probabilistic models. Given an indexed position in the graph, an anomaly threshold corresponding to that skill’s expected cumulative log likelihood. The figure also illustrates how at the end of a skill, all data is reset and restarted. The process is repeated when a new skill \( m \) is started. Given the position in the graph \( s_c \), we can index the correct log-likelihood \( I(\Pi_s = s_c) \) and see if its probability density of the test trial given the correct model is greater than the rest:
4.4 Log-Likelihood for Anomaly Monitoring

Fig. 4.2 Illustration of cumulative log-likelihood ($L$-) curves for four skills $s$ in a task. As such, 4 probabilistic models $\Pi_s$ are derived. Given an indexed position in graph $N_i$, one can generate four L-curves. The curve for the correct model, has a value increasing likelihood, while the other graphs have value decreasing likelihoods. One can also see how at the termination of a skill, all data is reset and restarted. No anomalies are shown in this plot.

\[
\log P(y_{t_1:t_r} | \Pi_{\text{correct}}) > \log P(y_{t_1:t_r} | \Pi_s)v
\]
\[
\forall s (s \in S \land s \neq s_c).
\]

If so, the identification is deemed correct, and the time required to achieve the correct diagnoses recorded. At the end of the cross-validation period, a diagnoses accuracy matrix is derived as well as the mean time threshold value (these results were also reported in [21]).

4.4.3 Anomaly Monitoring

Due to the characteristics of abnormality, diversity, and unpredictability in the robot’s motion behavior, unsupervised learning is the only way to achieve abnormality monitoring. In particular, this section proposes the use of log-likelihood function values based on non-parameterized Bayesian probability statistical models to implement anomaly monitoring. The main reason is that the probability model learned from normal data will behave when testing anomalous values. The cumulative log-likelihood function value decreases sharply. Additionally, the curve of the cumulative log-likelihood function value that calculated from normal movement behavior will show an increasing trend, and the curve calculated with abnormal motion behavior will drop significantly when anomaly occurred. Therefore, a method based on the log-likelihood function of the model is proposed to realize the abnormal monitoring of the robot’s motion behavior by setting the abnormal lower limit threshold.
Anomaly monitoring assumes that the cumulative log-likelihood $L$ of a set of nominal skill exemplars share similar cumulative log-likelihood patterns. If so, the expected cumulative log-likelihood derived in training can be used to implement an anomaly threshold $F_1$. Initially, we consider a likelihood curve generated from training data for a given skill $s$. Then, for each time step in an indexed skill $s_c$, the anomaly threshold is set to

$$F_{1s_c} = \mu(L) - c \times \sigma(L)$$  \hspace{1cm} (4.3)

where $c$ is a real-valued constant that is multiplied by the standard deviation to change the threshold. Here, we are only interested in the lower (negative) bound. Then, an anomaly is flagged if the cumulative likelihood crosses the threshold at any time:

if $\log P(y_{r1}c_r | \Pi_{correct}) < F_{1s_c}$: anomaly, else nominal. However, it is found that the threshold obtained by Eq. (4.3) makes the abnormal monitor frequently trigger falsely (False Positive) during actual application, which means that the normal execution condition monitoring becomes abnormal. In addition, the determination of constant variables requires tedious interactive verification. Aiming at this problem, this section refers to the experience of robot motion recognition, and determines the threshold of abnormality by solving the gradient information of each log-likelihood function value vector.

Upon our initial exploration of recovery schemes, we noticed that after resetting the cumulative log-likelihood observations in the HMM model, false-positive anomalies were triggered at the beginning of the skill. Further examination revealed that the standard deviation of cumulative log-likelihood graphs during training began with small variances but grew over time as shown in Fig. 4.3. Given that variances are small at the beginning of the task, small variations in observations can trigger failure flags.

A second threshold definition was designed to overcome this situation and used in our work instead of $F_1$. As the difference between $L$ and $F_1$ is minimal at first, the new anomaly threshold $F_2$ (for an indexed skill) is focused on computing the derivative of the difference:

$$F_{2s_c} = \frac{\left| \frac{d}{dt} \left| L - F_{1s_c} \right| \right|}{dt}$$  \hspace{1cm} (4.4)

Figure 4.4 illustrates the derivative signal crossing the empirical anomaly threshold as anomalies are triggered by external disturbances.

### 4.4.4 Experiments and Results

Three manipulation experiments are designed to test the robustness of the online decision making system for anomaly recovery. We use accidental human collisions as disturbances for a pick-and-place and open-and-close-drawer tasks, and tool col-
4.4 Log-Likelihood for Anomaly Monitoring

Fig. 4.3 This plot shows how the standard deviation of the cumulative log-likelihoods computed during training grows over time. The standard deviation grows as observations show greater variance due to the accumulation of error.

Fig. 4.4 Example of how the anomaly identification metric crosses the anomaly threshold as anomalies are triggered by external perturbations during a skill execution.
Fig. 4.5 Two examples (pick-and-place (UPPER) and open-and-close-drawer) in which a human collaborator accidentally collides the robot. The introspection system identifies an anomaly (see bottom left plots) and triggers a recovery behavior (see the fluorescent node in the graph on the right).

For motion generation, two techniques were used for the different tasks. For the pick-and-place task, we used Baxter’s internal joint trajectory action server which uses cubic splines for interpolation. The goal target is identified online through image-processing routines. For the open-and-close-a-drawer task, we trained dynamic movement primitives using kinesthetic teaching.

In terms of robot introspection, the sHDP-HMM code with “memoization” variational coordinate ascent, with birth and merge moves was implemented using BNPY [28] and wrapped with ROS. Training used 10-trial batches. Observations used 13 dimensional vectors composed of 7 DoF pose values (position and quaternion) and 6 DoF wrench values. A baseline HMM algorithm was implemented through hmmlearn [28] and wrapped with ROS. The anomaly threshold for each skill was computed through leave-one-out cross-validation.
Figure 4.5 shows a representative image of the Baxter robot attempting a pick operation. A human collaborator accidentally collides with robot before a pick action. Note that collisions were strong enough to move the current pose significantly from the intended path and sometimes collide with other parts of the environment. The robot introspection system identifies an anomaly and triggers a recovery behavior. The lower left part of the image shows the anomaly F2-metric flagging the anomaly. The system then begins recovery as seen in the directed graph on the right (implemented in ROS-SMACH). Video and auxiliary data for the three experiments under the four conditions are available in [21].

For results reporting, two markers are provided: (i) A success recovery rate for the recovery policy and (ii) an F-score, precision, and recall numbers for assessing anomaly identification. In particular, these markers are used under experimental conditions (iii) and (iv) conditions described at the beginning of the experiment and results section. The first two conditions are used as a baseline and compare with the generic recovery behavior success rates.

(1) Human Collisions

**One anomaly per node:** 27 test trials were used for the pick-and-place task and 24 test trials were used for the open-and-close drawer task. Given that both of the tasks have 5 nodes, a total of 5 anomalies were induced in the task, 1 per node. Table 4.1, shows the recovery success rate of the task (represented by the F-score) and the robustness of the identification system through the precision-and-recall quantities for both micro and macro settings. The pick-and-place recovery had an average

|                  | HMM | sHDP-HMM |
|------------------|-----|---------|
|                  | Micro | Macro | Micro | Macro |
|                  | F      | Recall | Precision | Recall | Precision | F      | Recall | Precision |
| Human collision  |        |        |        |        |          |        |        |          |
| Pick & Place     |       |        |        |        |          |        |        |          |
| (1 An./skill)    | 0.88  | 0.88   | 0.96   | 0.88  | 0.97     | 0.82  | 0.81   | 1.00     | 0.82  | 1.00     |
| (Mult. An./skill)| 0.84  | 0.83   | 0.97   | 0.84  | 0.97     | 0.85  | 0.84   | 0.95     | 0.85  | 0.95     |
| Open drawer      |       |        |        |        |          |        |        |          |
| (1 An./skill)    | 0.80  | 0.80   | 1.00   | 0.80  | 1.00     | 0.92  | 0.90   | 0.82     | 0.90  | 0.85     |
| (Mult. An./skill)| 0.72  | 0.72   | 1.00   | 0.72  | 1.00     | 0.93  | 0.93   | 1.00     | 0.93  | 1.00     |
| Tool collision   |       |        |        |        |          |        |        |          |
| Pick & Place     | 0.71  | 0.71   | 1.00   | 0.71  | 1.00     | 0.89  | 1.00   | 0.89     | 1.00  | 0.89     |
success rate of 85% with a maximum of 88%. The precision was 100% (for both macro/micro) indicating strong resistance to false positives. The recall was \(~82\%\) (for macro/micro) indicating the existence of some false-negatives. This might indicate that there might have been some collisions that were of lower magnitudes than the ones we might have trained with and were not detected by the system. For the drawer task, the recovery success rate was 91.67%. The precision was 95% (macro/micro) and recall was \(~84.5\%\). As with the human collision, weaker contact signals in tests compared to training might be the reason for the presence of false-negatives in our system.

**Multiple anomalies per node:** Under this scenario the pick-and-place task ran 42 test trials and the drawer task ran 30 test trials. Five anomalies were induced repeatedly one-after-the other for each node. The pick-and-place recovered 85% of the time with a precision of \(~95\%\) and a recall of \(~84.5\%\) (micro/macro). The drawer task recovered 93.3% of the time with a precision of \(~100\%\), and a recall of \(~93\%\) (micro/macro).

(2) Tool Collisions
For the tool collision experiment, we only tested recovery in the pick-and-place task. The results for this scenario were similar to that of human collision. The recovery success rate was \(~88.89\%\), the precision 100%, and the recall 88.89%.

### 4.4.5 Discussion

This work showed the ability of a system to recover from unmodeled and accidental external disturbances that can’t be anticipated. Such disturbances will be more common in shared human-robot workspaces. Our work demonstrated that our recovery strategy in connection with our previous introspection framework recovered 88% of the time from accidental human and tool collisions under single-anomaly and multiple-anomaly scenarios per node. The results indicate the system can recover at any part of the task, even when it is abused and multiple anomalies occur consecutively. From the videos in [21], we can see that even when in cases where the robot is in constant duress, the robot recovers consistently. Such robustness will play a role in enabling robots have increasing levels of long-term autonomy.

We by using a more expressive model to do robot introspection, our recovery ability also increased. We will continue to explore improved models that can better capture spatio-temporal relationships of high-dimensional multi-modal data. As well as looking for representations that scale over time in order to acquire a useful and practical library of skill identification and motion generation.

Yet there is much work to be done. Manual annotations for state-dependency are an important weakness. Crucially we wish to move towards modeling human understanding for decision making in the midst of robot-object-environment interactions. Scalability is an important factor in this domain as the system must scale to ever growing number of learned tasks. Learning how anomalies and recovery
decisions are made and re-used across similar scenarios will be important. The manual approach will not scale. Adaptability and not only reverting is also important. Incremental learning is also key. The current work is limited to reverting. By simply revering we don’t model recovery behaviors and also do not learn how to adapt. We also cannot handle new scenarios. We must develop action-confidence metrics that let us learn new scenarios on demand.

4.5 Gradient of Log-Likelihood for Anomaly Monitoring

Based on the implementation in Sect. 4.4, it is difficult to determine the constant parameters $c$ in actual applications, which leads to frequent false positives. To address the problem, this section explores an anomaly monitoring method based on the log-likelihood function gradient value and mathematically verifies its implementation process to prove its feasibility and effectiveness. After identifying the motion behavior $s$, this method assumes the parameters of the specific motion behavior and its established probability model are known as $\Theta$. From Sect. 2.4.1, we can know that the log-likelihood function value for the observation sequence can be determined by the probability in all credible states, and the simplified description is

$$\mathcal{L}(t) = \log \sum_{i=1}^{N} \alpha_i(t) = \log \sum_{i=1}^{N} \exp(\log \alpha_i(t)) \quad (4.5)$$

Therefore, given an HMM model $\Theta$ and an incoming multimodal time series $Y_{1:T}$, the natural logarithm of the filtered belief state associated with the forward model for latent state $i \in \{1, 2, \ldots, N\}$ can be represented according to Eq. (4.5). To compute $\mathcal{L}_i$, we first compute $\log \alpha_i(t)$. According to the forward-algorithm, we derive:

$$\alpha_i(1) = \pi_i b_i(y_1),$$
$$\alpha_i(t + 1) = b_i(y_{t+1}) \sum_{j=1}^{N} \alpha_j(t) A_{ji}. \quad (4.6)$$

From Eqs. (4.5) and (4.6), we know that $\log \alpha_i(t)$ can be computed recursively through $\log \alpha_s(t - 1)$. Expanding the $\log$ in Eq. (4.5) we have:

$$\log \alpha_i(t) = \log b_i(y_i) + \log \sum_{j=1}^{N} \exp(\log \alpha_j(t - 1) + \log A_{ji}) \quad (4.7)$$
4.5.1 The Hidden Markov Model Forward Gradient

- **Viterbi Path in HMMs**

The Viterbi algorithm, expanded in Eq. (4.8), attempts to estimate the most likely state sequence. Viterbi uses dynamic programming to estimate the underlying state sequence \( \hat{z}_{1:t} \) through MAP computations given a sequence of observations \( y_{1:t} \):

\[
\hat{z}_{1:t} = \arg \max_{z_{1:t}} P(z_{1:t} \mid y_{1:t})
= \arg \max_{z_t} b_{z_t}(y_t)
= \arg \max_{z_{t-1}} (A_{z_{t-1} z_t} P(z_{1:t-1}, y_{1:t-1}))
= \arg \max_{z_t}(b_{z_t}(y_t))
\]

(4.8)

First, we introduce simplified notation for the Viterbi algorithm, where a node \( j \) has a maximal belief state \( \delta_t(j) = \max_{z_{1:t}} P(z_{1:t} = j \mid y_{1:t}) \) with associated traceback \( \hat{z}_t \).

**Theorem 4.1** For an incremental time series \( Y \), a good HMM model outputs an incremental Viterbi path that stably expands on the previous one. The stable expansion of the Viterbi path is as follows: given a Viterbi path “11223” for an input \( Y[1:t] \), then the path at \( Y[1:t+1] \) becomes “11223*”, where * is the newly appended hidden state.

Good models are those that predict their data as accurately as possible and can be achieve through two steps: (i) HMM parameters optimization: the Baum–Welch algorithm given a proper initialization or similar algorithm incrementally optimize HMM parameters until a local maximum of likelihood is reached; minimizing the perplexity of the model. (ii) Model selection optimization: this consists in selecting the number of hidden states and values for observation models. Many techniques exist including BIC, MCMC, Variational Bayes, or non-parametric HMMs [29].

**Proof** Consider, without loss of generality, a Viterbi graph where we examine two consecutive time-steps \( (t-1, t) \) along two possible latent states \( l, k \) (the analysis generalizes to HMMs with more states). We also assume \( \forall i, i = \arg \max_{j} A_{ij} \); that is, all hidden states states tend to self-transition. Also at time \( t-1 \):

\[
\delta_{t-1}(l) > \delta_{t-1}(k).
\]

(4.9)
We also define the following symbol:

\[ w_{ji}(t) = A_{ji} * b_i(y_t). \] (4.10)

Due to our first assumption, we have: \( \max_j w_{ji}(t) = w_{ii}(t) \). Then, at time \( t \), the \( \delta \) values are:

\[
\delta_t(l) = \max(\delta_{t-1}(l) * w_{ll}(t), \delta_{t-1}(k) * w_{kl}(t)) \quad (4.11)
\]

\[
\delta_t(k) = \max(\delta_{t-1}(l) * w_{lk}(t), \delta_{t-1}(k) * w_{kk}(t)) \quad (4.12)
\]

According to (4.9) and our max weight formulation, the max function in (4.11) is: \( \delta_t(l) = \delta_{t-1}(l) * w_{ll}(t) \). So, the max state \( l \) at time \( t - 1 \) will contribute to itself instead of \( k \) at time \( t \). Therefore, there is only one condition under which the Viterbi sequence breaks:

\[
\delta_t(k) = \delta_{t-1}(k) * w_{kk}(t) \quad \text{and} \quad \delta_t(k) > \delta_t(l)
\]

In other words, given our original assumption, the Viterbi sequence breaks if the following inequalities are met:

\[
\delta_{t-1}(k) * w_{kk}(t) > \delta_{t-1}(l) * w_{lk}(t)
\]

and

\[
\delta_{t-1}(k) * w_{kk}(t) > \delta_{t-1}(l) * w_{ll}(t)
\]

In ratio form:

\[
\frac{w_{kk}(t)}{w_{lk}(t)} > \frac{\delta_{t-1}(l)}{\delta_{t-1}(k)} \quad (4.13)
\]

and

\[
\frac{w_{kk}(t)}{w_{ll}(t)} > \frac{\delta_{t-1}(l)}{\delta_{t-1}(k)} \quad (4.14)
\]

If an observation is emitted by state \( l \) and it is not undergoing a state switch, \( (\delta_{t-1}(l) > \delta_{t-1}(k) \text{ and } w_{ll}(t) > w_{kk}(t)) \), inequality (4.14) fails and the Viterbi sequence does not break.

When we transition from state \( l \) to \( k \) and begin emitting \( w_{kk}(t) > w_{ll}(t) \). However, the momentum in \( \delta_t(k) \) and \( \delta_t(l) \) prevent their inequality relationship to switch. Nonetheless, after \( p \) time steps, the inequality \( \delta_{t-1}(l) > \delta_{t-1}(k) \) becomes \( \delta_{t-1+p}(l) < \delta_{t-1+p}(k) \). The latter is reasonable when state \( k \) has been emitting for some time. It is before this time \( t - 1 + p \) that inequalities (4.13) and (4.14) are met. To see why, one will notice the left hand side inequalities in (4.13) and (4.14) are larger than 1 while the right hand side ratios become smaller than 1 at time \( t - 1 + p \), thus their inequality relationships must’ve swapped before this time. Note that the
order in which inequalities (4.13) and (4.14) are met matters. If (4.14) is met but (4.13) is not, a clean cut occurs since we have:

\[ \delta_{t-1}(k) \ast w_{kk}(t) < \delta_{t-1}(l) \ast w_{lk}(t) \]  

(4.15)

and

\[ \delta_{t-1}(k) \ast w_{kk}(t) > \delta_{t-1}(l) \ast w_{ll}(t) \]  

(4.16)

Equation (4.16) asserts a switch from \( l \) to \( k \). Equation (4.15) states that \( \delta_{t-1}(l) \) contributes to \( \delta_t(k) \), implying the previous max state contributes to the next max state. And since the max state has changed, the roles of \( l \) and \( k \) swap and inequality (4.14) begins to fail and renders sequence break unattainable. This yields a clean transition cut. If (4.13) and (4.14) are met simultaneously, a sequence break occurs. But since (4.14) is met, the states switch and the roles of \( l \) and \( k \) swap and preclude a further sequence break.

If, let’s say at time \( t_a \), inequality (4.13) is met but (4.14) is not, a future sequence break is destined to happen at the future moment when (4.14) is met, let’s say at time \( t_b \). When that sequence break occurs, the history between \( t_a \) and \( t_b \) flips and after the sequence breaks the state transitions from \( l \) to \( k \). Again, roles switch and no further sequence breaks occur. Above all, we can safely conclude that the during the execution of the stable period of a hidden state, no sequence break will occur. It is only during state transitions that a sequence could break and even if it does, it only last for a single time step—the time step when inequality (4.14) is met. The analysis extends to HMM models with more than 2 hidden states, so long as we apply the analysis to pairs of hidden states composed of the current max state and another non-max state.

**Corollary 4.1** Given Theorem 4.1 the gradient of the log-likelihood of the forward algorithm (from now on referred to as the forward gradient) will depend solely on the latest emission probabilities and the transition matrix.

**Proof** According to the log-exp-sum trick [29], the approximation \( \log \sum_{i=1}^{N} \exp(y_i) \approx \max_{i \in \{1, \ldots, N\}} y_i \) is best approached for larger values of \( y_i \). Applying this approximation to Eqs. (4.5) and (4.7), which is supported by Theorem 4.1, we have:

\[ \mathcal{L}_t \approx \max_{i \in \{1, \ldots, N\}} (\log \alpha_i(t)) \]  

(4.17)

\[ \log \alpha_i(t) \approx \log b_i(y_t) + \max_{j \in \{1, \ldots, N\}} (\log \alpha_j(t - 1) + \log A_{ji}) \]  

(4.18)

Substitute Eq. (4.18) into Eq. (4.17), rename \( i \) and \( j \) to \( z_t \) and \( z_{t-1} \), then recursively decompose \( \log \alpha_i \), we have:
\[ L_t = \max_{z_t \in \{1, \ldots, N\}} \left( \log \alpha_{z_t}(t) \right) \]
\[ = \max_{z_t \in \{1, \ldots, N\}} \left( \log b_{z_t}(y_t) + \max_{z_{t-1} \in \{1, \ldots, N\}} \left( \log A_{z_{t-1}z_t} + \log \alpha_{z_{t-1}}(t-1) \right) \right) \]
\[ = \max_{z_t \in \{1, \ldots, N\}} \left( \log b_{z_t}(y_t) + \max_{z_{t-1} \in \{1, \ldots, N\}} \left( \log A_{z_{t-1}z_t} + \log b_{z_{t-1}}(y_{t-1}) + \ldots \right) \right) \]
\[ = \log b_{\hat{z}_t}(y_t) + \log A_{\hat{z}_{t-1}\hat{z}_t} + L_{t-1}. \quad (4.19) \]

Equation (4.19) is the log version of Eq. (4.8). This suggests that, after approximations by Eqs. (4.17) and (4.18), the computation of the log-likelihood is the same as the computation of the Viterbi path using the Viterbi algorithm.

Now, since Theorem 4.1 shows that in general the Viterbi path at time \( t \), \( \hat{z}_{1:t} \), expands on the Viterbi path at time \( t-1 \), \( \hat{z}_{1:t-1} \), we have:
\[ L_t = \max_{z_t \in \{1, \ldots, N\}} \left( \log b_{z_t}(y_t) + \max_{z_{t-1} \in \{1, \ldots, N\}} \left( \log A_{z_{t-1}z_t} + \log \alpha_{z_{t-1}}(t-1) \right) \right) \]
\[ = \log b_{\hat{z}_t}(y_t) + \log A_{\hat{z}_{t-1}\hat{z}_t} + L_{t-1}. \quad (4.20) \]

Then, the forward gradient can be derived from Eq. (4.20) as:
\[ \nabla L_t = \nabla \nabla L_t - L_{t-1} = \log b_{\hat{z}_t}(y_t) + \log A_{\hat{z}_{t-1}\hat{z}_t}. \quad (4.21) \]

Equation (4.21) supports Corollary 4.1, where the forward gradient depends on the latest emission probability \( b_{\hat{z}_t}(y_t) \) and transition probability from hidden state \( \hat{z}_{t-1} \) to \( \hat{z}_t \). Also, given that good HMM models have strong inertia (high probabilities of self-transitions), state-switching should be sparse and then \( \hat{z}_{t-1} \) will equal to \( \hat{z}_t \) most of the time.

- **Normal Skill Identification Based on the Forward Gradient**

Corollary 4.1 led us to design a new method for skill identification. If we use \( n \) HMM models to represent \( n \) robot skills, with observations coming from a certain skill, the HMM model corresponding to that skill \( \hat{m} \), should output a value-increasing forward log-likelihood curve that is greater than the rest of the HMM models. This also means model \( \hat{m} \) will output a larger forward gradient value compared to other models.

The forward gradient depends on the latest emission probabilities, which in turn depend on the latest observation. The largest probabilities and thus gradients will belong to the HMM model of a currently executing robot skill. The key insight however is the inverse relationship: the use of the forward gradients to infer the
currently executing skill, and then it can validate the strong correlation between the forward gradient and skill observations. This is contrasted with the log-likelihoods of the observations \( \log P(Y_{1:t} | \Pi) \) do not. The forward-gradient measure for skill identification is defined as follows: given \( p \) skills \( s_1 : s_p \), we have HMM models \( m_s \) for \( s \in \{1, \ldots, p\} \) and an input time series \( Y \), then the most probable skill \( \hat{s} \) generating \( Y[t] \) is inferred as:

\[
\hat{s} = \arg \max_{s \in \{1, \ldots, n\}} (\nabla L_{m_s}^t(Y))
\]  

(4.22)

where, \( \nabla L_{m_s}^t(Y) \) is the forward gradient output by model \( m_s \) at time \( t \) computed using time series \( Y \).

4.5.2 Anomaly Monitoring Based on the Forward Gradient

We now build on the premise established in Eq. (4.22). Furthermore, consider a set of nominal observations for an executing skill, we know that the corresponding skill HMM model will output a value-increasing forward log-likelihood curve, and hence, a stable positive forward gradient. So, when an anomaly occurs, the forward gradient decreases significantly as illustrated in Fig. 4.6. Given that anomalies influence the forward gradient value, we propose a gradient-based metric for HMM anomaly monitoring.

Consider an HMM model \( m \) representing a skill \( s \) with \( n \) time-series trials \( Y_i \) for \( i \in \{1, \ldots, n\} \) collected from successful executions of skills \( s \in S \). To detect anomalies in a new time series \( Y \) we can first derive:

\[
\nabla L_{\text{max}} = \max_{i \in \{1, \ldots, n\}} (\max_{t \in \{1, \ldots, T_i\}} (\nabla L_{m}^t(Y_i)));
\]

\[
\nabla L_{\text{min}} = \min_{i \in \{1, \ldots, n\}} (\min_{t \in \{1, \ldots, T_i\}} (\nabla L_{m}^t(Y_i)));
\]

\[
\nabla L_{\text{range}} = \nabla L_{\text{max}} - \nabla L_{\text{min}};
\]

Fig. 4.6 For each skill (denoted in different background color), the log-likelihood gradient can be used for skill identification.
where $T_i$ is the time length of trial $Y_i$ and $\nabla L_t^m(Y_i)$ is the forward gradient output by model $m$ at time $t$ computed using time series $Y_i$. Then, we use an empirically-derived test to trigger an anomaly for $Y$:

$$\nabla L_t^m(Y) < \nabla L_{min} - \frac{\nabla L_{range}}{2}.$$  \hspace{1cm} (4.23)

This test detects if the gradient is an outlier compared with gradients of successful skill executions.

### 4.5.3 Experiments and Results

As for experimental setup, a dual armed humanoid Baxter robot was used to perform a pick-and-place operation. The robot consisted of a Robotiq force-torque sensor and standard Baxter fingers. 5 nominal trials were used for training the HMM model. In testing, 5 trials were used for skill identification and anomaly monitoring respectively. The pick-and-place task consists of 5 skills:

1. hover over the picking position,
2. grasp the object,
3. lift the object,
4. hover to the placing position, and
5. place the object.

Figure 4.7 shows the experimental setup and the execution of the five skills.

For training, the observation vector concatenates a 7-dimensional Cartesian end-effector pose and a 6-dimensional wrench. For each skill, we train corresponding HMM models using the Baum–Welch algorithm. The number of hidden states is selected such that emission probabilities are maximized leading to distinct and uniquely grouped hidden states.

For results reporting, we use the three factors identified by Pettersson’s survey on Event Detection [30]. Namely: diagnoses accuracy, robustness (false-positive rate), and reaction time (the time it takes to identify a skill from the beginning of a skill execution). Note that for anomaly identification, internal and external perturbations are used including: unexpected movement of target object, object absence, slippery picks, and human collisions.

**(A) Skill Identification Performance**

Figure 4.8 presents The skill identification confusion matrix. Skills 1 and 3 were recognized with 100% accuracy, 2 and 4 with 99% accuracy, and skill 5 with the largest surface contacts with 94% accuracy. Overall accuracy was 98.4%.
Fig. 4.7 The Baxter humanoid robot performing a pick-and-place task. 5 independent skills used to perform the task. Executed skill motions are sketched with red arrows.

Fig. 4.8 Skill identification confusion matrix for pick-and-place.
(B) Reaction Time Performance

In terms of reaction time, a percentage is computed to assess the time it takes for the identification to execute from the beginning of a skill. The reaction percentage, using t as “true”, is computed as: $\text{offset} = \text{predicted-}t_{\text{beginning}}$, $\text{length}=t_{\text{end}}-t_{\text{beginning}}$, and $\text{reaction}=\text{offset}/\text{length}$. The closer the reaction percentage is to 0% the better the identification method. A negative reaction percentage means the predicted start occurs earlier than ground truth, while a positive percentage implies a delayed identification. We assess two forms to determine the beginning of a skill (i) use the “first skill” occurrence, or (ii) use the “first 10 successive skill” occurrences. The reaction percentage for these two formats is found in Table 4.2. The average reaction time for absolute values (i.e. looking at the average time difference of the prediction, whether early or late) the “first skill” is 2.70% across all skills and the average reaction time for the “first 10 skills” is 0.97%. Between the two measures, we have a total average of 1.84%.

(C) Anomaly Monitoring Performance

For anomaly monitoring performance of our gradient-based method, we use two environments: (i) anomaly identification as it occurs and any false-positives before an actual anomaly occurs, and (ii) an external collision is given to the robot to trigger a recovery. Then, when the robot completes its recovery behavior, we count how many false-positives are triggered before moving to the next skill execution. The robot recovery behavior is detailed in [31]. Five nominal and five anomalous trials are used for the analysis. The results are compared with two other baseline methods: the magnitude-based metric from the description in normal skill identification based on the forward gradient, and the derivative-of-difference metric from the definition in anomaly monitoring based on the forward gradient.

The five anomalous trials contain a total of 14 anomalies, consisting of:

(i) one anomaly caused by the displacement of the target object
(ii) one anomaly caused by no target object
(iii) two anomalies caused by slippery picks
(iv) five anomalies caused by human collisions to the robot gripper during each skill execution
(v) five anomalies caused by human collisions to the robot arm during each skill execution.

| Table 4.2 | Reaction time as a duration percentage of a skill |
|-----------|-----------------------------------------------|
| Method    | Average reaction percentage                   |
|           | Skill 1 (%) | Skill 2 (%) | Skill 3 (%) | Skill 4 (%) | Skill 5 (%) |
| First skill occurrence | 0.00 | 0.23 | –3.23 | 0.28 | –9.77 |
| First 10 occurrences | 0.00 | 0.23 | –2.36 | 0.65 | 1.61 |
### Table 4.3 Anomaly monitoring metrics performance comparison

| Detection metric                        | Micro F-score (%) | Precision (%) | Recall (%) |
|----------------------------------------|-------------------|---------------|------------|
| Proposed forward-gradient measure      | 100               | 100           | 100        |
| Derivative-of-difference measure       | 82.35             | 70            | 100        |
| Magnitude-based measure                | 60.47             | 44.83         | 92.86      |

- **(D) Pre-recovery Performance**

For pre-recovery performance computation, during each trial, we record the anomalies triggered by the testing metric and count its true positives, false positives and false negatives. The results are shown in Table 4.3. The result shows our proposed forward gradient detected all anomalies and triggered no false positives or false negatives. The other two baseline methods suffer from false positives though they deliver high true positives.

- **(E) Post-recovery Performance**

For post-recovery performance metrics, we trigger an intentional anomaly, after recovery is completed, we count any false-positives before next skill execution. Both the forward gradient method and the derivative-of-difference method had not false-positives. Whilst the magnitude-based metric had more than 10 and prevented the system from continuing its task execution.

- **Comparison with Related Works**

Comparisons across works is challenging as results use different formats across experiments. Table 4.4 is an effort to harmonize results across related literatures. The comparison should be done loosely as different tasks (small levels of contact vs. large levels of contacts, structured environment vs. unstructured environment) present different challenges to event detection. For skill identification, our current approach ranks 2nd behind the tool breakage work that identified anomalies in structured milling processes. Our work did better than [16, 22], albeit these works modeled more complex dynamical phenomena. Similar statements can be made about anomaly identification. As for reaction times, our approach offers about double the speed-up compared to the only other work that reported this number. In conclusion, based on internal and external evidence, we hold that our measure is the most robust, stable, and fastest measure reported to date.

#### 4.5.4 Discussion

This work presented a theoretically derived event detection measure useful for nominal and anomalous behavior identification, even in post-recovery actions. Our results
showed very strong performance compared with state-of-the-art results across the board. More experimental validation is certainly necessary: both in number of trials and robotic tasks. This work also remains to be tested in the area of anomaly diagnoses. The latter is concerned not only with the identification problem with the grouping of anomaly types which is more challenging. We anticipate working in conjunction with machine or deep learning models for the diagnoses of this signals. Some works [11, 12, 16, 23] already provide some characterizations.

4.6 Mapping Latent State to Log-Likelihood for Anomaly Monitoring

4.6.1 Hierarchical Dirichlet Process Vector Auto-regressive Hidden Markov Models

HMMs segment sequential data into interpretable hidden states that are assumed Markovian. They are limited by its a priori determination of hidden states (and fixed throughout the modeling process) as well as the HMMs limited ability to model complex dynamics due to the assumption that observations are conditionally independent
given the state. Instead, this section uses Bayesian non-parametric techniques that model HMMs through priors that can model an unbounded number of hidden states. In particular, we use HDP priors to learn HMMs with infinite hidden states. We also use the switching vector auto-regressive (HDP-VAR-HMM) to model multimodal observation sequences and its element consists of features extracted from an F/T sensor, joint encoders, and a tactile sensor (for a discussion on the used features see experimental setup). Figure 4.9 illustrates graphical model representations for the sHDP-HMM and the sHDP-VAR-HMM.

Assume that we record \( N \) sequential executions for each skill and expect to learn the common dynamics among them. We utilize the HDP-VAR-HMM to jointly model those sequences, where sequence \( n \in N \) has multidimensional observation data \( y_n = [y_1, y_2, \ldots, y_T] \) at time \( t \). For instance, \( y_t \in \mathbb{R}^d \) could be an instant of wrench signatures and end effector velocity, where \( d \) is the dimensionality of observed features. The HDP-VAR-HMM interprets each observation \( y_t \) by assigning it a single hidden state \( z_t \). The hidden state value is derived from: (i) a countably infinite set of consecutive integers \( z_t = k \in \{1, 2, 3, \ldots\} \), (ii) an initial probabilistic state distributions \( \pi_0 \) generated with Markovian dynamics, and (iii) a transition distribution \( \{\pi_k\}_{k=1}^{\infty} \). Then, the Markovian structure on the state sequence dictates that for all \( t > 1 \):

\[
\begin{align*}
  z_t &\sim \pi_{z_{t-1}}, \\
  p(z_1 = k) &= \pi_0, \\
  p(z_t = k' | z_{t-1} = k) &= \pi_{kk'}.
\end{align*}
\]

\( (4.24) \)

Since \( z_t \sim \pi_{z_{t-1}} \), we see that \( z_{t-1} \) indexes all observations \( y_t \) are assigned with \( z_{t-1} \). We can draw the observation \( y_t \) given specific hidden state \( z_t = k \) from its corresponding observation likelihood functions. In our case, we consider the first-order auto-regressive Gaussian likelihood. As such, observations \( y_t \) are considered to be a noisy linear combination of the previous observation plus additive white noise \( \varepsilon \) and can be expressed as:

\[
y_t = A_k y_{t-1} + \varepsilon_t(z_t = k), \quad \varepsilon_t(k) \sim \mathcal{N}(0, \Sigma_k) \quad (4.25)
\]

where, each state \( k \) has two dynamic parameters \( \{A_k, \Sigma_k\} \): \( A_k \), a \( d \times d \) matrix of regression coefficients that defines the expected value of each successive observation as \( \mathbb{E}[y_t | y_{t-1}] = A_k y_{t-1} \), and \( \Sigma_k \), a \( d \times d \) symmetric positive definite matrix that defines the covariance matrix of state \( k \).

Therefore, the Gaussian regression observation model explains a observed data pairs of input \( y \) and output \( z \). Each input is a vector of length \( y \in \mathbb{R}^d \), while each output is a scalar. In the paper, we assume that the input data \( y \) are fixed dependence on previous observation and focus on a generative model for learning the output data \( z \) which depends on \( y \). Various generative models, such as full-covariance Gaussian, diagonal-covariance Gaussian, are possible for the observed input data. These can be directly define the multivariate Gaussian log-likelihood function given specific state \( z_k \),
4.6 Mapping Latent State to Log-Likelihood for Anomaly Monitoring

Fig. 4.9 Graphical representation of HDP-HMM and HDP-VAR-HMM. a The graphical representation of a sticky HDP-HMM over $T$ time steps. The latent, discrete-valued Markov process $z_t$ captures the temporal dependencies in the observations $y_t$. b The graphical model for a sticky HDP-VAR ($r$)-HMM over $T$ time steps. The latent, discrete-valued Markov process $z_t$ dictates the linear dynamical model at time $t$. For an order $r$ switching VAR process ($r = 1$ in this paper), the conditionally linear dynamics are completely determined by previous observations $y_{t-1}$.
\[
\log p(y_t|\theta_k, y_{t-1}) = \log p(y_t|A_k, \Sigma_k, y_{t-1}) = \log \mathcal{N}(y_t|A_k y_{t-1}, \Sigma_k)
\]
\[
= -\frac{d}{2} \log(2\pi) - \frac{1}{2} \log |\Sigma_k| - \frac{1}{2} (y_t - A_k y_{t-1})^T \Sigma_k^{-1} (y_t - A_k y_{t-1}).
\]
\[(4.26)\]

However, it is often the case that we don’t know the parameters of this distribution, but instead to estimate them. We need to learn the approximate values \(\theta_k = \{A_k, \Sigma_k\}\) for each state by defining a conjugate prior distribution on it. Particularly, the Matrix Normal Inverse Wishart (MNIW) distribution is conjugate for both the \(A_k\) and \(\Sigma_k\) are uncertain. This distribution assume that when only the covariance \(\Sigma_k\) is uncertain, the conjugate prior is defined as \(d\)-dimensional Inverse Wishart (IW) distribution with covariance parameter \(\Delta\), a symmetric, positive definite scale matrix, and \(\nu\), the degrees of freedom, i.e.
\[
\Sigma_k \sim \mathcal{I}W(\nu, \Delta),
\]
\[(4.27)\]
where the first moment is given by the mean
\[
\mathbb{E}[\Sigma_k] = \frac{\nu \Delta}{\nu - d - 1}.
\]
\[(4.28)\]

After that, we place a conditionally Matrix-Normal (MN) prior on \(A_k\) given \(\Sigma_k\), which is given by
\[
A_k \sim \mathcal{M}\mathcal{N}(M, \Sigma_k, V),
\]
\[(4.29)\]
where \(M\) is the mean matrix of \(A_k\) and \(V\) is a symmetric, positive definite matrix that determine the covariance of \(A_k\) relative to \(\Sigma_k\). Therefore, we derive the useful expectations
\[
\mathbb{E}(A_k) = M,
\]
\[
\text{Cov}[A_k] = \frac{1}{\nu - d + 1}(V^{-1} \times \Delta).
\]
\[(4.30)\]

The resulting log-likelihood function of joint distribution of \(A_k, A_k\) is defined as
\[
\log p(A_k, \Sigma_k|\nu, \Delta, M, V^{-1})
\]
\[
= F_{\mathcal{I}W}(\nu, \Delta) + F_{\mathcal{M}\mathcal{N}}(M, V) + F,
\]
\[(4.31)\]
where,
\[
F_{\mathcal{I}W}(\nu, \Delta) = -\frac{d(d-1)}{4} \log \pi - \frac{d}{2} \log(2\pi v) - \sum_{i=1}^{d} \log \Gamma\left(\frac{\nu + 1 - d}{2}\right) + \frac{\nu}{2} \log |\Delta|,
\]
\[
F_{\mathcal{M}\mathcal{N}}(M, V) = -\frac{d^2}{2} \log 2\pi + \frac{d}{2} \log |V|,
\]
\[
F = \frac{\nu - 1}{2} \log |A_k| - \frac{1}{2} tr(A_k^T A_k V) + tr(A_k A_k V M^T) - \frac{1}{2} tr(A_k (\Delta + M V M^T)).
\]
\[(4.32)\]
where $\Gamma()$ is represented the gamma function. Therefore, all the optional hyperparameters of HDP-VAR-HMM are specified by $\nu, \Delta, M, V$. Apparently this straightforwardly method is computationally feasible when those four parameters are known. We will specify the detailed options in experimental section through all our experiments.

With reference to the Eq. (4.24), for the probabilistic representation of transition distribution $\pi_j$ under the HDP-VAR-HMM prior, the number of states is unbounded and every observation assigned a unique state. The HDP prior encourages sharing states over time through a latent root probability vector $\beta$ over the infinite set of states (see Fig. 4.9). Consider the sticking-breaking construction of the prior on $\beta$ and first draws independent variable $\mu_j \sim Beta(1, \gamma)$ for each state $j$, and then sets $\beta_j = \mu_j \prod_{i=1}^{j-1} (1 - \mu_i)$, where the concentration parameter $\gamma$ controls the relative proportion of the weights $\beta_j$. Here, we interpret $\mu_j$ as the conditional probability of choosing state $j$ among states $\{j, j+1, j+2, \ldots\}$.

In expectation, we always describe the first $K$ orders in stick-breaking as the most common states and represent their probabilities with the vector $[\beta_1, \beta_2, \ldots, \beta_K, \beta_{>K}]$, where $\beta_{>K} = \sum_{j=K+1}^{\infty} \beta_j$. Given this (K+1) dimensional probability vector $\beta$, we simplify the HDP-VAR-HMM transition distributions $\pi_j$ for each state $j$ from a Dirichlet distribution with mean equal to $\beta$ and variance governed by concentration parameter $\alpha > 0$

$$[\beta_1, \beta_2, \ldots, \beta_K, \beta_{>K}] \sim Dir(\alpha \beta_1, \alpha \beta_2, \ldots, \alpha \beta_{>K}),$$

(4.33)

Generally, we define the starting probability vector $\pi_0$ from the same prior, but with much smaller variance, $\pi_0 \sim Dir(\alpha_0 \beta)$ with $\alpha_0 \gg \alpha$, which denotes that few starting states are observed.

To capture the sufficient dynamics of given time series $n$ with $T_n$ observations. Our goal is to interpret these observation with the hidden state $z_j$. However, instead of assigning variable states in a short time period, we should treat each observation in the context of its predecessors for improving the temporal consistency. A “sticky” parameter $\kappa$ of favors self-transition by assigning extra prior mass on the transition probability $\pi_{jj}$ was introduced. Thus, the transition representation will in a temporally consistent way, that is

$$[\beta_1, \beta_2, \ldots, \beta_K, \beta_{>K}] \sim Dir(\alpha \beta_1, \ldots, \alpha \beta_j + \kappa, \ldots, \alpha \beta_{>K}).$$

(4.34)

where $\kappa > 0$ governs the degree of self-transition bias. Informally, what we are doing is increasing the expected probability of self-transition for keeping the dynamical phenomenon consistency for many time-steps.

After constructing the sHDP-VAR-HMM model, we utilize the state-of-the-art variational inference algorithm for learning the parameters, which named as Split-Merge Monte Carlo method by Michael C. Hughes et al. More information can be found in [28].
4.6.2 Anomaly Monitoring Based on Hidden State

The anomaly detector identifies whether a skill execution contains: no anomalies, one anomaly, or multiple anomalies. In this section, we introduce the mathematical modelling that makes up our detector.

- (A) Hidden State Representation

Hidden state space modeling of multivariate time-series is one of the most important tasks in representation learning by dimensional reduction. In this work, we propose the anomaly detector is a thresholding marginal log-likelihood with Bayesian non-parametric hidden Markov model, which leads to tackle the anomaly monitoring problem in a more natural way to meet the need of real-world applications.

As we discussed above, the sHDP-V AR-HMM interpreted each observation $y_t$ by assigning to a single hidden state $z_t$, where the hidden state value is derived from a countably infinite set of consecutive integers $z_t = k \in \{1, 2, 3, \ldots, K\}$. Figure 4.10 shows the low-dimensional hidden state space of the multi-modal time series generated by the robot during manipulation, which contains 5 hidden states.

We denote $\Theta_s$ to represent all the parameters of the trained sHDP-VAR-HMM from nominal demonstrations, including the hyperparameters for learning the posterior and the parameters for the observation model definition. Our anomaly detector performs anomaly monitoring by comparing the marginal log-likelihood $\log p(y_t|\Theta_s)$ based on current executing skill of the observation $y_t$ with a learned threshold $\rho(z)$, that derives from the most likely estimated hidden state given current observations.

$$z = \arg \max_{1,...,K} p(z_t|y_t, \Theta_s). \quad (4.35)$$

![Fig. 4.10](image) An illustration for the representation of multimodal time series in low dimensional hidden state space
Here, the $z$ would be included variable values, that is, we should synchronously update the threshold in the same skill period. Let’s say the definition of an anomaly data point at any time during the manipulation is one that $logp(y_t|\Theta_s) < \rho(z)$, otherwise it deems the manipulation to be nominal.

The proposed method generates the mapping pairs of marginal log-likelihood and the most likely hidden state at any time. As specific hidden state $z$, we calculated the fixed log-likelihood threshold from testing the nominal demonstrations by subtracting half of range between maximum and minimum from the minimum. Unlike the anomalous log-likelihood was deviate by a certain standard deviation from the mean in \cite{11, 12, 21}, that is $\rho(z) = \mu - c\sigma$, where the symbol $c$ is a constant value for adjusting the sensitivity of the detector, $\mu$ and $\sigma$ are the estimated mean and standard deviation of the log-likelihood. Our implementation don’t need to take the constant $c$ into consideration for each skill or hidden state.

- **(B) Joint Probability of the Observed Sequence**

The threshold $\rho(z)$ is associated with the joint probability of the observed sequence. Firstly, in order to derive the forward-backward procedure of first-order auto-regressive HMM for the joint state sequence $z_{1:T}$ given observation $y_{1:T}$, plus the initial observation $y_0, (T \geq 1)$. We have this problem when dealing with conventional HMM, here the auto-regressive structure of ARHMM makes this situation more complicated, a modified forward-backward method could solve the problem efficiently. Let’s first write the joint distribution with the Markov property $p(y_t|y_{t-1}, \ldots, y_1) = p(y_t|y_{t-1})$, that is

\[
p(y_0:T) = \sum_{t=1}^{T} p(y_t|y_{1:t-1}),
\]

\[
p(y_t|y_{1:t-1}) = \sum_{k=1}^{K} p(z_t = k|y_{1:t-1}) p(y_t|z_t).
\]

Where $\theta_z$ represents the observation parameters for the trained HMM and the $z_{1:T}$ is a state path over the hidden state space. As such, if the state path is estimated, the maximum probability of the observation sequence can be obtained. However, since the true state path is hidden from the observations. For the computational convenience, the general approach would be to use the maximum likelihood state at any given moment, but this would neglect uncertainty. Instead, we use a marginal probabilistic representation over hidden states at each time step (see details in hidden state estimation of auto-regressive model section). Thus, the log-probability at time $t$ is derived by computing the logarithm of the sum of exponentials over all hidden states

\[
\mathcal{L}_t = \log \sum_{k=1}^{K} \exp \left( \frac{\alpha_t(k) \cdot \beta_t(k)}{p(y_{0:T})} \right),
\]
where $\alpha_t(k) = p(z_t = k | y_{0:t})$, $\beta_t(k) = p(y_{t+1:T} | z_t)$, are presented the forward message passing and backward message in the standard forward-backward algorithm, respectively.

$$
\alpha_t(k) = \frac{1}{p(y_t | y_{0:t-1})} p(y_t | z_t = k) p(z_t = k | y_{1:t-1}),
$$

$$
p(z_t = k | y_{0:t-1}) = \sum_j p(z_t = j | z_{t-1} = j) p(z_{t-1} = j | y_{0:t-1}).
$$

The denominator represents the probability of a sequence observations, which can be calculated by Eq. (4.36). Assume that we recorded $N$ sequential data for each skill and the length of sequence $n \in \mathbb{N}$ is $T_n$. Thus, we can get a set of log-probabilities vector of each skill

$$
\mathcal{L}^{1:N} = \{ \mathcal{L}_1^1, \mathcal{L}_2^2, \ldots, \mathcal{L}_N^N \}.
$$

The above implies that the forward—backward procedure can be used to find the most likely state for any moment $t$. It cannot, however, be used to find the most likely sequence of states.

- \textbf{(C) Hidden State Estimation of Auto-regressive Model}

For offline constructing the state and log-likelihood pairs, we are interested in the most likely state sequence upon an observation sequence $y_{0:T}$. One might be derived from copulating the MAP sequence by maximizing the marginal probability at each observation $i$

$$
\hat{z}_i = \arg \max_{z_i} p(z_i | y_{0:T}).
$$

We note this combined single observation estimated sequence might yield unlikely (even 0-probability) sequences. We therefore consider

$$
\hat{z}_{1:T} = \arg \max_{z_{1:T}} p(z_{1:T}, y_{0:T}).
$$

where, $p(z_{1:T}, y_{0:T})$ was computed in Eq. (4.36). Actually, we note the most likely state sequence that minimum the cost path to $z_t$ is equivalent to the minimum cost path to node $z_{t-1}$ plus the cost of a transition from $z_{t-1}$ to $z_t$, and the cost incurred by observation $y_j$.

For convenience, we abbreviate the joint log-likelihood function $\mathcal{L}(z_{1:T}) \triangleq \log p(z_{1:T}, y_{0:T})$, and the transition probability $\pi_{z_{t-1}} \triangleq z_{t-1}$, then

$$
\mathcal{L}_T(z_{1:T}) = s_1(z_1) + \sum_{t=2}^T s_t(z_t, \pi_{z_{t-1}}),
$$

where
4.6 Mapping Latent State to Log-Likelihood for Anomaly Monitoring

\[ s_1(z_1) = \log (\pi_{z_1} p(y_1|\theta_{z_1}, y_0)), \quad t = 1 \]
\[ s_t(z_t, z_{t-1}) = \log (p(z_t|z_{t-1}) p(y_t|\theta_{z_t}, y_{t-1})), \quad t \geq 2 \]

(4.43)

Obviously, \( \mathcal{L}_t(z_{1:t}) = \mathcal{L}_{t-1}(z_{1:t-1}) + s_t(z_t, z_{t-1}) \), and set

\[ v_t(z_t) = \max_{z_{t-1}} \mathcal{L}_t(z_{1:t}), \quad z_t \in \{1, 2, 3, \ldots, K\} \]
\[ v_t(z_{t-1}) = \max_{z_t} \{v_{t-1}(z_t) + s_t(z_t, z_{t-1})\} \]

(4.44)

From the above, The maximizing state sequence is iteratively obtained, for \( n \in \{T-1, T-2, \ldots, 1\} \)

\[ \hat{z}_T = \arg \max_{z_{t-1}} v_n(z_{t-1}) \]

(4.45)

Assume that we recorded \( N \) sequential data for each skill and the length of sequence \( n \in N \) is \( T_n \). Thus, we can get a set of the most probable states vector of each skill

\[ \mathcal{Z}^{1:N} = \{ \mathcal{Z}^1_{T1}, \mathcal{Z}^2_{T2}, \ldots, \mathcal{Z}^N_{TN} \} \]

(4.46)

• (D) Threshold Calculation

Having computed the joint probability of the observations and the most probable states for each skill \( s \), we can concatenate them to compute the threshold of specific state \( z = k \), that is \( \rho(z_k) \). Each state consists of its maximum \( \mathcal{L}_{zk}^{\text{max}} \) and minimum \( \mathcal{L}_{zk}^{\text{min}} \). We cluster the log-likelihoods with the same state, which represents the clustering data by HMM belong to the same statistical distribution.

\[ \mathcal{L}_{zk}^{\text{max}} = \max_{i \in \{1, \ldots, N\}} \mathcal{L}_i^n 1(z_k = k), \]
\[ \mathcal{L}_{zk}^{\text{min}} = \min_{i \in \{1, \ldots, N\}} \mathcal{L}_i^n 1(z_k = k), \]
\[ \mathcal{L}_{zk}^{\text{range}} = \mathcal{L}_{zk}^{\text{max}} - \mathcal{L}_{zk}^{\text{min}}, \]

(4.47)

where the notation \( 1(\eta) \) to represent an indicator random variable that is 1 if the event \( \eta \) occurs and 0, otherwise. Using Eq. (4.47), we can set the anomaly monitoring threshold of each active state \( z_k \) at any moment according to:

\[ \rho(z_k) = \mathcal{L}_{zk}^{\text{min}} - \frac{\mathcal{L}_{zk}^{\text{range}}}{2}. \]

(4.48)

In testing implementation, we preprocessed the incoming multimodal data in the same scheme as the training process and then get the most possible state \( z_{test} \) by recursively compute the filtered marginals and the corresponding log-likelihood \( \mathcal{L}_{test} \) using Eqs. (4.38) and (4.37) respectively.

\[ z_{test} = \arg \max_{1, \ldots, K} \{ p(z_t = 1|y_{0:t}), \ldots, p(z_t = K|y_{0:t}) \}. \]

(4.49)
Here, this approach detects an anomaly when the log-likelihood is lower than the state dependent threshold $\rho(z_{test})$. This approach also extends to online detection. Given the trained sHDP-V AR-HMM and streaming sensor data, the detector can recursively compute the $z_t$ and $L_t$ at time $t$, and then perform the comparison, if $L_t < \rho(z_{test})$, then anomaly, else nominal.

### 4.6.3 Experiments and Results

- **Finite State Machine Based Kitting Experiment**

In this section, we describe the kitting experimental setup, external disturbances, data collection, and anomaly monitoring and diagnoses model parameters.

A kitting experiment is setup, where a robot and a human co-worker closely collaborate to place a set of goods in a packaging box. Specifically, a human co-worker is tasked to place a set of 6 objects on the robot’s “collection bin” (located in front of the robot) in a one-at-a-time fashion. The objects may accumulate in a queue in front of the robot. As soon as the first object is placed on the table, the robot identifies the object and places it in a packaging box located to the right of the robot. Thus, the robot picks an object and transports it towards the box, after which, the robot appropriately places each of the six objects in different parts of the box. The whole implementation is shown in Fig. 4.11.

The kitting experiment consists of 6 skills (Skill 3) : Home $\rightarrow$ Pre-pick; (Skill 4) : Pre-pick $\rightarrow$ Pick; (Skill 5) : Pick $\rightarrow$ Pre-pick; (Skill 7) : Pre-pick $\rightarrow$ Pre-place; (Skill 8) : Pre-place $\rightarrow$ Place; (Skill 9) : Place $\rightarrow$ Pre-place that were effectively modeled by DMP, respectively. Objects that need to be packaged are placed by a human collaborator before the robot in a collection bin. The shared workspace affords possibilities for accidental contact and unexpected alteration of the environment. The robot is tasked to pick each one of the objects and place them in the packaging box to its right. The visualization module uses the ALVAR tags to provide a consistent global pose with respect to the base of the robot. Ten nominal executions and one snapshot of each skill are shown.
modeled by the dynamic movement primitives and the ROS-SMACH.\textsuperscript{1} One-shot kinesthetic demonstrations were used to train DMP models for each of the skills of the kitting experiment. Note that a skill’s starting and ending pose can be adapted if necessary thus providing a flexible and generalizable skill encoding procedure. The trajectory adaptations however increase the difficulty of assessing the sensory data for both anomaly monitoring and diagnoses—due to the variability in the data collection even from nominal executions.

- **The Robot, Objects, and Visual System**

A Baxter humanoid robot’s right arm is used to pick commonplace objects set before him. The robot is equipped with a 6 DoF Robotiq FT sensor, 2 Baxter-standard electric pinching fingers, as well as Baxter’s left hand camera. Each finger is further equipped with a multimodal tactile sensor composed of: (i) a $4 \times 7$ taxel matrix that yields absolute pressure values, (ii) a dynamic sensor which provides a single capacitive reading in millivolts (mV) useful to detect tactile events, and (iii) an IMU and gyroscope [34]. Baxter’s left hand camera is placed flexibly in a region that can capture objects in the collection bin with a resolution of $1280 \times 800$ at 1 fps (we are optimizing pose accuracy and lower computational complexity in the system). The use of the left hand camera facilitated calibration and object tracking accuracy.

A set of 6 common household objects consisting of box-like shapes and bottles were used in our work. The objects ranged in weight from 0.0308 to 1.055 kg and in volume from $3.2 \times 10^{-04}$ m$^3$ to $1 \times 10^{-03}$ m$^3$. The object’s surfaces also varied slightly: some heavier objects had sleeker surfaces to incite object slips—we believe not an unreasonable determination as warehouses contain a wide variety of objects—whilst other objects had rougher surfaces. Across executions, object locations and order was varied to promote generalization.

Alvar tags,\textsuperscript{2} with 0.06 m sides, were placed around the circumference of the objects for robust visual recognition (ALVAR can handle change in lighting conditions, optical flow-based tracking, and good performance for multitag scenarios) regardless of orientation.

- **External Disturbances**

External disturbances may be introduced into a system for a variety of reasons. We list possible scenarios in which disturbances may occur in real-systems, as shown in Fig. 4.12:

1. For collaborative jobs (such as kitting in warehouses, see the description in experiment setup section), humans may experience monotony leading to boredom and loss-of-attention. In such cases, it is possible for a human co-worker to accidentally collide with the robot manipulator or alter the environment in unexpected ways.

\textsuperscript{1}http://www.ros.org/SMACH.
\textsuperscript{2}http://wiki.ros.org/ar_track_alvar.
(2) The user may also accidentally collide or unintentional move packaging objects in ways a robot may not anticipate. Such variations in the world may lead to tool-collisions.

(3) Picked objects may slip from a robot’s gripper if the grasp is not optimal; or if upon motion, inertial forces acting on the object cause slippage that breaks the grasp.

(4) Object slips may also be caused due to human collisions. Such phenomena depicts a chain of anomaly reactions where one anomaly leads to another.

(5) The robot may also collide with packaging box during kitting.

(6) Missed-grasps (where the robot pinches air) are possible when the target object moved without the robot’s notice.

(7) Additionally, it is possible to suffer from false-positives from the anomaly detector. False-positives may occur for numerous reasons: system errors, unreachable objects, unfeasible inverse kinematic solutions, unidentifiable objects from the visual system, to name a few.

The seven types of anomalies we just described will be declared more succinctly in the rest of the paper as: Human-Collisions (HC), Tool-Collisions (TC), Object Slips (OS), Human-Collision-with-Object (HCO), Wall-Collision (WC), No-Object (NO), and Other (OTHER) respectively. In Sect. 4.6.1, the introspection methodology used to model robot skills, continued by a presentation of anomaly monitoring in Sect. 4.6.2 and anomaly diagnoses in Chap. 5.
• Dataset Collection and Sensory Preprocessing

It is undeniable that bias exists on the side of robot researchers as to which anomalies may exist in the task. In this work, we first attempt to discover likely anomalies in this task by emulating a collaborative kitting task in which the human collaborator experiences tedium and monotony resulting in undesired events that lead to anomalous events.

We tasked 5 participants—whose ages range from 23 to 28—to act as a collaborator in the kitting task under the monotonous conditions mentioned in experiment setup section. One expert user and four novice users participated in the experiments. Novice users learned from the expert to induce anomalies by observing a round of executions. Each participant performed 1 nominal and 6 anomalous executions by placing the set of six household objects in a one-by-one fashion. For training, when an anomaly occurs, the robot execution is halted and the data stream is labelled with an anomaly class (namely the labels with HC, TC, OS, NO, etc.) for supervised classification during testing. Subsequently, anomalous time-stamps are also extracted. In total, we collected a data set with 18 nominal executions and 180 anomalous executions (each anomalous trial has at least one anomaly across the entire task), and 38 failure trails are ignored.

As for our multimodal observation vector, it is comprised of 6 DoF force-torque signals, 6 DoF Cartesian velocity signals (from Baxter’s right end-effector), and 56 DoF tactile signals from both the left and right tactile sensors. Empirical feature extraction is performed on this observation vector to improve diagnoses performance. We now describe how features for each modality at time \( t \) were engineered.

**Wrench modality:** Assume the force and torque sequence is a time-series vector and that each element represents the magnitude of each dimension \((f_x, f_y, f_z, t_x, t_y, t_z)\).

We wish to extract structural information from the wrench instead of simply using raw values. In this way, we can find signal patterns that may occur across the different DoFs. To this end, we computed the norm of both the force \( n_f \) and the torque \( n_t \) as features respectively:

\[
 n_f = \sqrt{f_x^2 + f_y^2 + f_z^2}, \quad n_t = \sqrt{t_x^2 + t_y^2 + t_z^2}. \tag{4.50}
\]

**Velocity modality:** Similarly, we take the Cartesian linear \((l_x, l_y, l_z)\) and angular \((a_x, a_y, a_z)\) velocities and compute their norm \( n_l \) and \( n_a \) respectively:

\[
 n_l = \sqrt{l_x^2 + l_y^2 + l_z^2}, \quad n_a = \sqrt{a_x^2 + a_y^2 + a_z^2}. \tag{4.51}
\]

**Tactile modality:** Due to the computational cost of processing the tactile sensor’s high dimensionality, we empirically tested a number of features for each tactile sensor, they include: the maximum taxel value, the largest five taxel values, the mean of all taxel values, and the standard deviation for all taxel values. It was the standard deviation, which proved to be the most useful feature for anomaly monitoring and diagnoses. The standard deviation for each tactile sensor \( s_l \) and \( s_r \) are defined as:
\[ s_l = \sqrt{\frac{1}{28} \sum_{i=1}^{28} (l_i - \mu_l)}, \quad s_r = \sqrt{\frac{1}{28} \sum_{i=1}^{28} (l_i - \mu_r)}. \quad (4.52) \]

where \( \mu_l = \frac{1}{28} \sum_{i=1}^{28} l_i \) and \( \mu_r = \frac{1}{28} \sum_{i=1}^{28} r_i \) are the mean of each tactile sensor respectively. Equations (4.50)–(4.52) facilitate the identification of diverging signals during anomalous situations. We empirically concatenate all the valuable features for representing the robot executions both in nominal and anomalous cases. If raw signals are used directly, they result in high false-positive detection rates (FPR) in our experimentation. For instance, when a robot carries objects of varying weights, the F/T signals for such operations vary significantly making it difficult to use raw values for detection. Hence, a standardization method is used to scale the extracted features through a normalization constant. We ultimately end with an feature vector of length 18 as shown below:

\[ y_t = \begin{bmatrix} f_x, f_y, f_z, t_x, t_y, t_z, n_f, n_t, l_x, l_y, l_z, a_x, a_y, a_z, n_t, n_a, s_l, s_r \end{bmatrix}. \quad (4.53) \]

Note, that all training and testing data are pre-processed in the same manner. We will also use the same pre-processing approach when comparing with other baseline approaches.

• Basic Parameter Settings of Each Model

For anomaly monitoring and diagnoses using the sHDP-VAR-HMM with a first-order autoregressive Gaussian likelihood, each state \( k \) has two parameters to be defined: the regression coefficient matrix \( A_k \) and the precision matrix \( \Lambda_k \). The conjugate prior is the MNIW for which four parameters \( \nu, \Delta, V, M \) are assigned values in a fashion similar to the Motion-Capture-Dataset in [28]. In order to guarantee the prior has a valid mean, the degrees-of-freedom variable is set with \( \nu = d + 2 \) and we set \( \Delta \) by constraining the mean or expectation of the covariance matrix \( \mathbb{E}[A_k^{-1}] \) under the Wishart prior in Eq. (4.30).

\[ \mathbb{E}[A_k^{-1}] = s_F \sum_{n=1}^{N} \sum_{t=1}^{T_n} (y_t - \bar{y})(y_t - \bar{y})^T. \quad (4.54) \]

Assume that we record \( N \) sequential data for each skill and the length of sequence \( n \in \mathbb{N} \) is \( T_n \). Thus, we can easily define the parameter \( \Delta \) accordingly as

\[ \Delta = (\nu - d - 1)\mathbb{E}[A_k^{-1}]. \quad (4.55) \]

Specifically, we placed a nominal prior on the mean parameter with mean equal to the empirical mean and expected covariance equal to a scalar \( s_F \) times the empirical covariance, here \( s_F = 1.0 \). This setting is motivated by the fact that the covariance is computed from polling all of the data and it tends to overestimate mode-specific
covariances. A value slightly less than or equal to 1 of the constant in the scale matrix mitigates the overestimation. Also, setting the prior from the data can move the distribution mass to reasonable parameter space values. The mean matrix $M$ and $V$ are set such that the mass of the Matrix-Normal distribution is centered around stable dynamic matrices while allowing variability in the matrix values (see Sect. 4.6.1 for details).

\[
M = 0_d, \\
V = 1.0 \times I_d.
\]  

(4.56)

where $0_d$ and $I_d$ are the matrix of all zeros and the identity matrix, respectively, each of size $d \times d$. For the concentration parameters, a $\text{Gamma}(a, b)$ prior is set on HDP concentration parameters $\alpha$. A $\text{Beta}(c, d)$ prior is set on the self-transition parameter $\mu$ and the degree of self-transition bias $\kappa$ is set to 50. We choose a weekly informative setting and choose: $a = 0.5$, $b = 5$, $c = 1$, $d = 5$. Specifically, the initial transition proportion parameter is set $\mu \sim \text{Beta}(1, 10)$. The Split-Merge Monte Carlo method sampling parameter maximum iterations is set to 1000. The truncation states number is set to $K = 5$ for anomaly monitoring, and $K = 10$ for anomaly diagnoses. Then, the HDPHMM model of each anomaly class or nominal skill is trained. Note, that as with pre-processing, we also use the same model parameters in training and testing as well as in other baseline approaches.

- **Results and Analysis of Anomaly Monitoring in Kitting Experiment**

According to Sect. 4.6.2, the anomaly detector implementation consists of representing multimodal observations through latent states, concatenating the derived log-likelihood values of a latent state, and calculating the anomaly monitoring threshold.

We first generate the latent state partitions for nominal demonstrations with varying speeds and goals. Notice how the non-parametric method yields varying dynamics across executions reflecting statistical variations in the robot motion trajectory due to changes in the desired trajectory completion time and varying goal distances.

Then, latent state log-likelihood values are concatenated and the corresponding thresholds computed. To this end, we visualized the performance of the proposed method as well as a baseline by assessing the anomaly time reaction of anomaly flags; that is, which algorithm can make a correct detection at a time closer to the ground truth. For training, ten nominal executions were used along with 3-fold cross validation to obtain nominal sHDP-VAR-HMM models for each skill. The other 20 nominal executions were used for testing.

The execution varying threshold for anomaly monitoring was set according to Eq. (4.48). The results for skill 3 are visualized in Fig. 4.13. As seen in Fig. 4.13, latent state $z\hat{=} 2$ is the most prominent latent state (the majority of the observations in the skill are ascribed to this state). We now analyze the time sensitivity of our system in detecting anomalies. Anomaly identification sensitivity is intimately connected to threshold setting. We compare the results of our execution varying threshold with a fixed threshold presented in [35]. The comparison is seen in Fig. 4.14. Under nominal situations both methods have the same robustness in avoiding false-positives. In anomalous situations, there are multiple occasions in which our varying
Fig. 4.13 All the log-likelihood values of individual hidden state of skill 3 when the truncation states number is set to $K = 5$. Each subplot shows the mean value (black dotted line) and the corresponding log-likelihood threshold value (red solid line) in Eq. (4.48).

threshold yielded better time sensitivity and detection accuracy. From Fig. 4.14, we see that for executions 2 and 4, our varying threshold detector identified anomalies that the fixed threshold method could not. Notice that the distance between the $anomaly_t\_by\_human$ and the $anomaly_t\_by\_detector$ is larger for the fixed threshold (recall from the dataset collection section that ground truth time stamps were obtained during training). This improved sensitivity reduces false-negatives and helps us flag anomalies at times closer to the ground truth.

Finally, Table 4.5 shows the anomaly monitoring performance for our hidden-state detector (HSD) and the baseline gradient-based detector (GD) for the six skills used in the kitting task for 368 nominal executions and 136 anomalous executions. Our results indicate that while the GD system had the HSD resulted in higher 16.6% better precision, our system had 76.0% better recall. The difference is significant in reducing false-negatives and significantly raising the sensitivity of our system. The sensitivity improvement is almost four times bigger than the difference in precision. For this reason, our system reflects an F1-score that is 25.0% larger and an accuracy that is $F1\_score$, and an accuracy that is 5.7% higher standing at 91.0%.
4.6 Mapping Latent State to Log-Likelihood for Anomaly Monitoring

The performance of different anomaly detectors

Fig. 4.14 The comparison of two anomaly detectors for five robot anomalous executions. Each row represents the same testing execution. Left column: results of our hidden state-based anomaly detector. Right column: results of the gradient-based anomaly detector [21]. The ground truth (anomaly_t_by_human) are represented by green dashed lines and the detected anomalies are represented by yellow solid lines.

4.6.4 Discussion

Comprehensive experimental results showed robust, sensitive, and better timed anomaly monitoring for a wide range of anomalous situations in an unstructured co-bot scenario where a human and a robot collaborated to complete kitting tasks. The improved anomaly detector exploits the mapping relationship between latent states and the marginal log-likelihood values from the sHDP-VAR-HMM for monitoring anomalies. Our evaluations showed that we could detect anomalies reliably (overall accuracy of 91.0%).
With regards to anomaly monitoring, Table 4.5 indicated that our proposed detector achieved unexpectedly superior recall measurements compared to the baseline (the gradient-based method), an improvement of 61.7%. By adjusting the anomaly monitoring threshold during the task’s progress execution task as a function of the maximum and minimum of the log-likelihood of observations for latent states, we gained enhanced sensitivity. Human-centric safety approaches dictate that increased sensitivity is highly desirable in human-robot collaboration tasks so as to provide higher safety guarantees for a human. Robotic workcells on the other hand may work better with approaches like the gradient-based detector that was less sensitive but had higher precision. Here the reduction in the false-positive rate may support longer-term autonomy in a workcell. The anomaly detector exhibited not only a better accuracy but also more accurately timed anomaly flags. That is, our anomaly flags occurred closer to the ground truth time. This would be useful. Timeliness may prove a critical role in cases where anomalies are triggered as a chain reaction. That is anomalies that cause more anomalies. By acquiring more timely detection, we may be able to also provide more timely recovery techniques or even prevention so as to minimize the possibility of a cascade of anomalies.

### 4.7 Summary

In this chapter, there are three kinds of anomaly monitoring methods in various threshold are presented based on the Bayesian nonparametric HMMs, namely: log-likelihood-based threshold, threshold based on the gradient of log-likelihood, and computing the threshold by mapping latent state to log-likelihood. All proposed methods and implementation processes were validated on the experimental platforms of “HIRO-NX robot performs electronic component assembly tasks” and “Baxter robot performs object packing tasks” using the ROC curve or accuracy-precision-recall rate-F1 score as a performance evaluation indicator. The results are as follows:

1. The anomaly monitoring method using log-likelihood-based threshold can effectively realize the online monitoring, but there are a large number of false pos-
itive (false triggering) and has an uncertain constant coefficient $c$. For the task of HIRO-NX robot performing the electronic components assembly, sHDP-VAR(2)-HMM obtains the optimal performance where the constant value of the threshold is $c = 3$ and the observation model is the second-order linear Gaussian regression model, resulting in a false positive rate of about 5.0% and a true positive rate of about 95.0%.

(2) The method of threshold based on the gradient of log-likelihood can effectively solve the problem of false triggering, but resulting in low sensitivity and monitoring success rate with a fixed lower threshold. For the task of Baxter robot to performing object kitting experiment, we obtain the average monitoring accuracy of a total of 67 anomaly events is 93.8%, far better than the 59.2% of the log-likelihood-based threshold.

(3) The anomaly monitoring method based on computing the threshold by mapping latent state to log-likelihood uses the dynamic threshold to solve the problem of false triggering, low sensitivity and success rate, but the dynamic threshold depends on the inference of the belief state, there will be a rapid transition phenomenon, resulting in a higher risk of false positive rate. For the task of Baxter robot to performing object kitting experiment, resulting in the average monitoring accuracy of a total of 136 abnormal events in the is 91.0%, which is much better than the 86.0% of the threshold based on the gradient of log-likelihood.

References

1. Pimentel MAF, Clifton DA, Clifton L, et al. A review of novelty detection. Signal Process. 2014;99:215–49.
2. Chandola V, Banerjee A, Kumar V. Anomaly detection: a survey. ACM Comput Surv (CSUR). 2009;41(3):15.
3. Rodriguez A, Bourne D, Mason M, et al. Failure detection in assembly: force signature analysis. In: 2010 IEEE conference on automation science and engineering (CASE). IEEE;2010. p. 210–5.
4. Xu Z, Li S, Zhou X, Wu Y, Cheng T, Huang D. Dynamic neural networks based kinematic control for redundant manipulators with model uncertainties. Neurocomputing. 2019;329(1):255–66.
5. Ando S, Thanomphongphan T, Hoshino D, et al. ACE: anomaly clustering ensemble for multi-perspective anomaly detection in robot behaviors. In: Proceedings of the 2011 SIAM international conference on data mining. SIAM;2011. p. 1–12.
6. Pastor P, Kalakrishnan M, Chitta S, et al. Skill learning and task outcome prediction for manipulation. In: 2011 IEEE international conference on robotics and automation (ICRA). IEEE;2011. p. 3828–34.
7. Chu V, McMahon I, Riano L, et al. Using robotic exploratory procedures to learn the meaning of haptic adjectives. In: 2013 IEEE international conference on robotics and automation (ICRA). IEEE;2013. p. 3048–55.
8. Clifton DA, Hugueny S, Tarassenko L. Novelty detection with multivariate extreme value statistics. J Signal Process Syst. 2011;65(3):371–89.
9. Qiao Y, Xin XW, Bin Y, et al. Anomaly intrusion detection method based on HMM. Electron Lett. 2002;38(13):1.
10. Zhang X, Fan P, Zhu Z. A new anomaly detection method based on hierarchical HMM. In: 2003 proceedings of the 4th international conference on parallel and distributed computing, applications and technologies. PDCAT’2003. IEEE;2003. p. 249–52.

11. Park D, Erickson Z, Bhattacharjee T, et al. Multimodal execution monitoring for anomaly detection during robot manipulation. In: 2016 IEEE international conference on robotics and automation (ICRA). IEEE;2016. p. 407–14.

12. Park D, Kim H, Hoshi Y, et al. A multimodal execution monitor with anomaly classification for robot-assisted feeding. In: 2017 IEEE/RSJ international conference on intelligent robots and systems (IROS). IEEE;2017. p. 5406–13.

13. Park D, Kim H, Kemp CC. Multimodal anomaly detection for assistive robots. Auton Robot. 2018;1–19.

14. Hu DH, Zhang X-X, Yin J, et al. Abnormal activity recognition based on HDP-HMM Models. In: IJCAI. 2009. p. 1715–20.

15. Di Lello E, De Laet T, Bruyninckx H. Hierarchical Dirichlet process hidden Markov models for abnormality detection in robotic assembly. 2012.

16. Di Lello E, Klotzbucher M, De Laet T, et al. Bayesian time-series models for continuous fault detection and recognition in industrial robotic tasks. In: 2013 IEEE/RSJ international conference on intelligent robots and systems (IROS). IEEE;2013. p. 5827–33.

17. Xu Z, Li S, Zhou X, Cheng T. Dynamic neural networks based adaptive admittance control for redundant manipulators with model uncertainties. Neurocomputing. 2019;357(1):271–81.

18. Xu Z, Li S, Zhou X, Yan W, Cheng T, Dan H. Dynamic neural networks for motion-force control of redundant manipulators: an optimization perspective. IEEE Trans Ind Electron Early Access. 2020. https://doi.org/10.1109/TIE.2020.2970635.

19. Süelch M, Bayer J, Ludersdorfer M, van der Smagt P. Variational inference for on-line anomaly detection in high-dimensional time series. 2016. arXiv:1602.07109.

20. Milacski ZA, Ludersdorfer M, Lőrincz A, Van Der Smagt P. Robust detection of anomalies via sparse methods. In: International conference on neural information processing. Cham: Springer;2015. p. 419–26.

21. Wu H, Lin H, Guan Y, Harada K, Rojas J. Robot introspection with Bayesian nonparametric vector autoregressive hidden Markov models. In: 2017 IEEE-RAS 17th international conference on humanoid robotics (Humanoids). IEEE;2017. p. 882–8.

22. Rojas J, Luo S, Zhu D, Du Y, Lin H, Huang Z, Kuang W, Harada K. Online robot introspection via Wrench-based action grammars. In: 2017 IEEE/RSJ international conference on intelligent robots and systems (IROS). IEEE;2017. p. 5429–36.

23. Di Lello HE, De Laet T, Bruyninckx H. HDP-HMM for abnormality detection in robotic assembly. In: NIPS workshop on Bayesian nonparametric models for reliable planning and decision-making under uncertainty. 2012.

24. Niekum S, Osentoski S, Konidaris G, Chitta S, Marthi B, Barto AG. Learning grounded finite-state representations from unstructured demonstrations. Int J Robot Res. 2015;34(2):131–57.

25. Hu DH, Zhang XX, Yin J, Zheng VW, Yang Q. Abnormal activity recognition based on HDP-HMM models. In: 21st international joint conference on artificial intelligence. 2009.

26. Hughes MC, Stephenson WT, Sudderth E. Scalable adaptation of state complexity for nonparametric hidden Markov models. In: Advances in neural information processing systems. 2015. p. 1198–206.

27. Murphy KP. Machine learning: a probabilistic perspective. Cambridge: MIT Press;2012.

28. Pettersson O. Execution monitoring in robotics: a survey. Robot Auton Syst. 2005;53(2):73–88
32. Ahmidi N, Gao Y, Béjar B, Vedula SS, Khudanpur S, Vidal R, Hager GD. String motif-based description of tool motion for detecting skill and gestures in robotic surgery. In: International conference on medical image computing and computer-assisted intervention. Berlin: Springer;2013. p. 26–33.

33. Cho S, Asfour S, Onar A, Kaundinya N. Tool breakage detection using support vector machine learning in a milling process. Int J Mach Tools Manuf. 2005;45(3):241–9.

34. Maslyczyk A, Roberge JP, Duchaine V. A highly sensitive multimodal capacitive tactile sensor. In: 2017 IEEE international conference on robotics and automation (ICRA). IEEE;2017. p. 407–12.

35. Luo S, Wu H, Lin H, Duan S, Guan Y, Rojas J. Fast, robust, and versatile event detection through HMM belief state gradient measures. In: 2018 27th IEEE international symposium on robot and human interactive communication (RO-MAN). IEEE;2018. p. 1–8.

Open Access  This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (http://creativecommons.org/licenses/by/4.0/), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter’s Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter’s Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.