Chapter 3
Incremental Learning Robot Task
Representation and Identification

Abstract  In this chapter, we present a novel method for incremental learning robot complex task representation, identifying repeated skills, and generalizing to new environment by heuristically segmenting the unstructured demonstrations into movement primitives that modelled with a dynamical system. The proposed method combines the advantages of recent task representation methods for learning from demonstration in into an integrated framework. In particular, we use the combination of finite state machine and dynamical movement primitives for complex task representation, and investigate the Bayesian nonparametric hidden Markov model for repeated skill identification. To this end, a robot should be able to identify its actions not only when failure or novelty occurs, but also as it executes any number of skills, which helps a robot understand what it is doing at all times. Two complex, multi-step robot tasks are designed to evaluate the feasibility and effectiveness of proposed methods. We not only present the results in task representation, but also analyzing the performance of skill identification by various nonparametric models with various modality combinations.

3.1 Introduction

As we all know, a sentence in a language is composed of words according to grammatical rules, and a word is composed of letters according to word formation rules. Then, a robot complex manipulation task corresponds to a sentence, the movement primitives of the robot coupling correspond to a word, and the movement primitives of the robot’s respective degrees can be expressed as letters. Therefore, the robot task can be represented by a set of learned movement primitives. Through the description of the movement primitives, the movement of the robot can be generalized, and the diversity and adaptability of the manipulation tasks can be improved. Based on the representation of the movement primitives of the robot tasks, the identification of the robot’s movement will facilitate the abnormal monitoring and recovery of the robot system.
### 3.2 Related Work

In recent years, as robots have been widely used in fields such as human interaction, home services, medical care, and industrial production, to enable non-robot professionals and people who lack specialized knowledge to quickly complete robot movement programming. Compared to the traditional robot control scheme [18, 23], robots Learning from human demonstrations has become mainstream for complex task descriptions [1, 3]. Generally, in order to effectively learn and generalize the robot’s manipulation tasks, the tasks are often divided into multiple segments of movement primitives for learning [4, 6], and thereafter the tasks are described by stitching or serializing different movement primitives [7–9]. Therefore, the learning methods of robot movement primitives mainly include the following four categories [12, 13]: Probabilistic movement primitives, dynamic movement primitives (DMPs), and probabilistic movements based on a Gaussian Mixture Model (GMM). Probabilistic Movement Primitive (ProMP) and Stable Estimator of Dynamical System (SEDS).

In particular, GMM has strong noise processing ability and strong robustness for human unstructured demonstration trajectories, and it is easy to deal with the problem of high-dimensional trajectories. The key implementation process is to first use the multivariate Gaussian distribution to model the demonstration trajectory, then use the Expectation-maximization (EM) algorithm and nonlinear regression technology to learn the model, and finally adjust the input new task configuration to achieve a generalized description of the trajectory, as DA Duque et al. presented in [11].

This method is embodied in that an arbitrary continuous probability distribution can be approximated by a weighted average of a mixture of multiple Gaussian distributions, and has been widely used in robot motion primitive learning and motion generation [14]. Elena et al. [15] learned the trajectories of human demonstrations through GMM, and estimated the stability of the non-linear multi-dimensional dynamic system on the generated trajectories, so that they can not only generate trajectories for unknown tasks but also perform online in the presence of external disturbances. Adjustment Calinon et al. [16, 17, 19] proposed a trajectory learning framework based on variance in the task space, modeled human trajectories through GMM, and learned EM algorithms for model unknown parameters. Finally, Gaussian Mixture regression model (Gaussian Mixture Regression (GMR) and optimization estimator. Peter et al. [20] used the Gaussian Process (GP) method to establish a regression model on the trajectory of human demonstration to express motion and used KLback–Leibler Divergence (KL) as an indicator of trajectory generalization performance. This learning through probability distribution guarantees the stability of motion generation.

Additionally, Schaal and Ijspeert et al. [13, 22, 24] to study the demonstration movement with a dynamic system of nonlinear differential equations. The most advanced advantage of this method is that only one demonstration trajectory can be used to complete the learning and generate the Adapted movement. At the same time, DMP inherits many advantages of dynamic systems, such as conditional con-
vergence, robustness to external disturbances, and time independence. Khansari et al. [21, 25, 26] aimed at the problem that the motion primitives learned in the GMM method could not guarantee its motion stability, and proposed a Stable Estimator of Dynamical Systems (SEDS) for nonlinear dynamic systems. The motion primitive learning method combines a dynamic system with a probabilistic statistical model to obtain its globally stable constraints. Finally, the parameter estimation problem is transformed into an optimization problem.

The robot’s manipulation tasks can be divided into multiple parameterized motion primitive, so that the combination and serialization of multiple different motion primitives can effectively improve the diversity and adaptability of robot tasks. Therefore, how to correctly select the next motion primitive after the execution of the current motion primitive is another research difficulty of the complex task description of the robot. In response to the serialization of motion primitives, Pastor et al. [27] proposed a motion description based on the Associative Skill Memories (ASM) of DMP motion primitives for motion selection, which differ from the control scheme [31]. Statistical data such as mean and variance are saved at all times. After completing 90% of the current motion primitives, use the remaining 10% of the sensing data and the first 10% of the data of the candidate primitives to calculate the distance of several miles, and the nearest one is executed as the next step.

Niekum et al. [28, 29] proposed the serialization of robot manipulation tasks by the finite state machine method, and described the target points of each motion primitive in different coordinate systems, and then described by different coordinate systems. The classifier constituted by the target, and finally selects the motion primitive to be executed next in the way of classification. Manschitz et al. [4, 9, 30, 32] proposed a method for learning task manipulation graphs from Kinesthetic Teaching, and learned the conversion probability between various motion primitives from the experience of multiple teachings. The method is successfully applied to the task of unscrewing a light bulb by a multi-step robot.

Zhe Su et al. [33] introduced and evaluated a framework to autonomously construct manipulation graphs from manipulation demonstrations. Our manipulation graphs include sequences of motor primitives for performing a manipulation task as well as corresponding con-tact state information. The sensory models for the contact states allow the robot to verify the goal of each motor primitive as well as detect erroneous contact changes. The proposed framework was experimentally evaluated on grasping, unscrewing, and insertion tasks on a Barrett arm and hand equipped with two BioTacs.

In summary, due to the teaching programming and offline programming methods of traditional industrial robots can no longer satisfy the representation of multi-step and complex robot manipulation tasks. In this chapter, through comparison and analysis, methods and theories for describing complex tasks of robots have been developed. The research combined the methods of dynamic movement primitives (DMP) and finite state machines (FSM) to achieve task representation and serialization selection among different movement primitives, an simple pick-and-place task was set in [34].
3.3 Graphical Representation of Robot Complex Task

In the case of unstructured and dynamic environments, the task of the robot is difficult to complete from beginning to end with fixed and pre-programmed movements, because the uncertainty of the environment and the unpredictability of the state of the robot in actual manipulation will cause the task The failure or exception occurred. For example, during the movement of the Baxter robot during the pick-and-place task, the robot will not be able to adjust the movement to adapt to changes in the environment due to changes in the pose of the object. Based on this problem, this paper proposes a way to artificially segment a complex task of a robot by describing multiple movement primitives, and uses a finite state machine mechanism to construct the task to implement a directed graph and motion primitive description of the task.

A finite state machine is a mathematical model that represents a finite number of states and the transition between these states. It is usually represented by a five-tuple: $M = (Q, q_0, \sum, \delta, F)$, where $Q$ is a non-empty finite set of states; $q_i \in Q$ represents a state; $q_0$ is the initial state; $\sum$ is the input condition Instruction; $\delta$ is a transfer function between states. Each state in the state machine stores the related information of the past, reflecting the changes before and after the input of the state machine; and the state transition indicates that the current state has changed. State transition rules are used to describe the necessary conditions for state transition. Generally, a directed state transition diagram can be used to describe the working condition of a finite state machine, which can clearly indicate the transition relationship and transition conditions between different states. As shown in Fig. 3.1, a system includes a finite state machine with four states, where each node represents a state $q_i$, and

![Fig. 3.1 Illustrates the graphical representation of FSM state transition](image-url)
each edge represents a transition relationship between states. If the input of state $q_0$ is $a$, the movement proceeds to state $q_1$, if the input is $b$, the movement proceeds to state $q_2$, and the other states transition. The relationship is the same. The final node $q_3$ represents the end state of the system.

According to the understanding of the finite state machine, the robot’s execution task is represented by a multi-segment movement primitive, and its primitive is derived from the human’s definition and understanding of the task. Generally, the following five factors need to be considered when segmenting tasks:

1. Each movement behavior should maintain independent and complete movement manipulation;
2. The smooth motion trajectory of the robot should be maintained in each movement behavior;
3. The stability of modal information should be maintained in each movement behavior;
4. A certain length of exercise should be maintained in each exercise behavior;
5. If the requirements are met, minimize the number of sports activities.

Based on the above considerations, the finite state machine description of the autonomous pick-and-place task of the Baxter robot in Sect. 4.5.3 is implemented, as shown in Fig. 3.2. Finally, in order to improve the adaptability and generalization ability of operating tasks in an unstructured dynamic environment, this paper uses the model learning method to model motion primitives between states on the premise of finite state machine description of tasks. In order to be able to adapt to the needs of the task and changes in the environment. In particular, in this paper, the motion primitive learning method DMP based on dynamic systems is used to model the movement

![Fig. 3.2 Illustrates the FSM implementation of Baxter robot performing pick-and-place task](image-url)
primitive. Thanks to this method, it only needs to learn from a human demonstration trajectory and good generalization. After learning the motion primitives, when a new starting amount and target amount are given to each movement behavior, a new movement behavior can be generated by referring to the demonstration movement, and a new manipulation task is generated. The specific parameterization and generalization process will be explained in the next subsection.

3.3.1 Learning Graphical Representation from Unstructured Demonstration

- Demonstration Collection

Generally, capturing the demonstrations by receiving the multimodal input from the end-user, such as a kinesthetic demonstration. The only restriction on the multimodal data is that all the signals must be able to be synchronously recorded at the same frequency, i.e. temporal alignment. Additionally, the multimodal data at each time step should include the Cartesian pose and velocity of the robot end-effector (in case of object-grasping, will along with any other relevant data about the end-effector, e.g. the open or closed status of the gripper and the relative distance between the end-effector and object) as well as the signals from F/T sensor and tactile sensor. Subsequently, the recorded Cartesian pose and velocity trajectory of the end-effector will be referred to as the kinematic demonstration trajectory for controlling the robot motion, and the recorded signals of F/T sensor and tactile sensor are applied for learning the introspective capacities.

- Finite State Machine

Defining the set \( \mathcal{M} \) as the union of all the unique states \( z \in \{1, 2, \ldots, K\} \) from the derived hidden state sequences \( \mathcal{Z} = \{\mathcal{Z}_1, \mathcal{Z}_2, \ldots, \mathcal{Z}_N\} \), a Finite State Machine (FSM) that represents the task can be designed to be constructed by creating nodes \( N_1, N_2, \ldots, N_m \), where \( K \leq m \) and each element should correspond to the same hidden state \( z \). For \( k \in \{1, \ldots, m\} \), each node \( N_k \) is assigned a set of \( \mathcal{M}_k \) of all exemplars that have the same hidden state and executing phase, and the starting and goal pose are also recorded with \( N_s, N_e \), respectively. A \( m \times m \) transition matrix \( T \) can then be constructed, where each element \( T_{i,j} \) is originally set to 1 if there exists a directed transition from \( N_i \) to \( N_j \), and 0 otherwise. We will implement the incremental task exploration by modulating this transition value as described in Chap. 6 in two ways. Consequently, we hopefully learn the robot complex task representation by formulating each state of constructed finite state machine using a state-specific movement primitive technique.

---

\(^1\)Here, the term “node” is used instead of the default “state” in FSM for avoiding confusing hidden state of HDP-VAR-HMM or state of the robot.
3.3 Graphical Representation of Robot Complex Task

3.3.2 Robot Movement Primitive Learning

Learning movement primitive has been commonly applied to solve different robot tasks in isolation. This usually requires either careful feature engineering, or a significant number of demonstrations. This is far from what we desire: ideally, robots should be able to learn from very few demonstrations of any given complex task, and instantly generalize to new situations of the same task, without requiring task-specific engineering. In this paper, we explore the Dynamical Movement Primitives (DMPs) for achieving such capability for equipping robot with the aforementioned generalization capability of movement primitive, which provides a framework for describing the dynamical systems as set of nonlinear differential equations. DMPs guarantee the stability and convergence by introducing an additional canonical system and also provide simple mechanisms for LfD. Particularly, a discrete movement DMP can be formulated by the transformation system, that is

\[
\begin{align*}
\tau \dot{v} &= K(g - x) - Dv - K(g - x_0)s + Kf(s), \\
\tau \dot{x} &= v.
\end{align*}
\] (3.1)

where, the Eq. 3.1 is an extended PD control signal with spring and damping constants \(K\) and \(D\) respectively, position and velocity \(x\) and \(v\), goal \(g\), scaling \(s\), and temporal scaling factor \(\tau\). The scaling term originates from an additional system that controls the system’s phase execution by

\[
\tau \dot{s} = -\alpha s,
\] (3.2)

where, the scalar \(\alpha\) can be an arbitrary constant.

Additionally, the forcing term \(f(s)\) in Eq. 3.1 is used to alter attractor point dynamics and achieve an arbitrary trajectory, which commonly learned from an one-shot demonstration [1]. Thus, along with the Eq.3.1, the forcing term \(f(s)\) can be formulated as a phase-dependent linear combination of a set of basis functions \(\psi_i(s)\) (often use the Gaussian basis function), that is,

\[
\tau f(s) = \frac{\sum_{i=1}^{N} w_i \psi_i(s)s}{\sum_i \psi_i(s)}
\]

\[
\psi(s) = (-h_i(s - c_i)^2)
\] (3.3)

where, the basis function \(\psi(s)\) is formulated with mean \(c_i\) and variance \(h_i\). Then, the forcing item is represented the linear combination of basis functions with variable weights \(w_i\) and normalization constant \(\sum_i \psi_i(s)\). Phase \(s\) monotonically varies from 1 to 0 and controls phase progress by activating Gaussian centered at \(c_i\). The decreasing phase value ensures that the contribution of the forced term disappears and returns to the simpler point attractor dynamics to converge to the target. Thus, the \((g - x)\) term in Eq. 3.1 that with respect to spatio-temporal scaling, which performs spatial scaling for guaranteeing the system can adjust to varying goals immediately. While the \(\tau\) variable in Eq. 3.1 is designed for allowing the system to speed up or slow
Fig. 3.3 Illustrates the learning and generalization of dynamical movement primitive down the execution of a movement. Generally, the forcing term weights are learned from a kinesthetic demonstration with pose $x(t)$, velocity $\dot{x}$ as well as acceleration $\ddot{x}$ with duration $T$ are extracted as in [29]. After learning movement primitive from One-shot demonstration, the target forcing turn of an unseen scenario is computed by adjusting Eq. 3.1, integrating Eq. 3.2 to convert from time to phase, and substituting appropriate values to yield,

$$f_{\text{target}}(s) = -K(g - x(s)) + D\dot{x}(s) + \tau\ddot{x}(s).$$

(3.4)

Next, the goal is set to $g = x(T)$ and $\tau$ is selected such that a DMP reaches 95% convergence at $t = T$ before using standard linear regression to compute the weights $w_i$.

In summary, when a new starting pose and target pose are given after learning the motion primitives, a new movement trajectory can be generated with reference to the demonstration motion, thereby improving the diversity and adaptability of complex tasks. As shown in Fig. 3.3, the DMP model is learned from a human demonstration trajectory (blue curve in the figure). When different starting points and target points are given, the corresponding motion can be generated by the learned DMP model. Track (gray curve in the picture). In addition, the DMP model has a good generalization ability, which is conducive to improving the diversity and adaptability of robot manipulation tasks in an unstructured dynamic environment. In particular, in the generalization process of the SPAIR system proposed in this paper, the starting position of each motion primitive is determined by the human demonstration movement or the external vision system.
3.4 Nonparametric Bayesian Method for Robot Movement Identification

3.4.1 Problem Statement

As those aforementioned related works of robot complex task representation, the task should be composed of multiple segments of movement primitives, and new manipulation tasks can be generated through the learning and generalization of learned primitives. As the planned robot manipulation tasks become more and more complex, the movement primitive library will increase accordingly, and there will be multiple feasible primitives candidate during the transition in the task. At this time, the manipulation task described by the finite state machine will select the primitive with the highest probability value for execution according to the current observation value. By real-time monitoring of the robot’s current behavior, that is, the identification that the robot is in or subsequently selecting that primitive, will help estimate the current execution status of the robot, and provide a basis for subsequent chapters for robot anomaly monitoring and recovery.

3.4.2 Gradient of Log-Likelihood for Movement Identification

Assume that \( N \) normal multimodal trajectories of each movement are collected for learning the common dynamics among them via kinesthetic demonstration or movement generation. To jointly model those trajectories using the Bayesian nonparametric hidden Markov model, particularly, the HDP-VAR-HMM (see details in Chap. 2) with linear Gaussian observation model is investigated, where each trajectory \( n \in N \) is represented as a multivariate time series that including multidimensional observation data \( y_n = [y_1, y_2, \ldots, y_T] \) at time \( t \). Take a robot common manipulation task as example, \( y_t \in \mathbb{R}^d \) could be an instant of wrench force, torque, and end effector velocity, where \( d \) is the total dimensionality of observed features. In HDP-VAR-HMM, each observation \( y_t \) is interpreted by a positive integer, named hidden state \( z_t = k \in \{1, 2, 3, \ldots\} \). A countably infinite set of consecutive integers \( z_t \) that should be satisfied with the requirements that an initial probabilistic state distributions \( \pi_0 \) and a infinite transition distribution \( \{\pi\}_{k=1}^{\infty} \) are generated using Markovian method. Consequently, the state sequence for all \( t > 1 \) with Markovian structure can be intuitively formulated as

\[
\begin{align*}
    z_t &\sim \pi_{z_{t-1}}, \\
p(z_1 = k) &= \pi_0, \\
p(z_t = k' | z_{t-1} = k) &= \pi_{kk'}. 
\end{align*}
\]
According to Eq. 3.5, the hidden state $z_{t-1}$ indexes all observations $y_t$ are assigned with $z_{t-1}$ at time step $t$. Since that, the observation $y_t$ given specific hidden state $z_t = k$ can be drew from its corresponding observation likelihood functions.

Since we assign the emission model of HDP-VAR-HMM as the first-order auto-regressive Gaussian likelihood. As such, each observation $y_t$ are considered to be a noisy linear combination of the previous observation $y_{t-1}$ plus additive Gaussian white noise $\epsilon$ and can be expressed as

$$y_t = A_k y_{t-1} + \epsilon_t(z_t = k), \quad \epsilon_t(k) \sim \mathcal{N}(0, \Sigma_k). \quad (3.6)$$

Since the $\epsilon$ is assumed with zero mean, each state $k$ should described with two dynamic parameters $\{A_k, \Sigma_k\}$, where, $A_k$ is a $d \times d$ matrix of regression coefficients that defines the expected value of each successive observation as $E[y_t \mid y_{t-1}] = A_k y_{t-1}$, and $\Sigma_k$ is a $d \times d$ symmetric positive definite matrix that defines the covariance matrix of state $k$. Then, the problem turns into how to calculate the VAR observation likelihood?

Therefore, the Gaussian regression observation model explains a observed data pairs of input $y_t$ and $y_{t-1}$, and output $z_t$, which is different from the HMM case. Resulting that each input is a vector of length $y \in \mathbb{R}^d$, while each output is a scalar. With this dimension reduction, the task segmentation by grouping the unique hidden states. In particular, we focus on a generative model for learning the output data that depends on $y_t$. As presented in [5], there are various generative models can be considered, such as full-covariance Gaussian, diagonal-covariance Gaussian, are possible for the observed input data. Consequently, the VAR observation likelihood can be directly define by the multivariate Gaussian log-likelihood function at specific state $z_k$,

$$\log p(y_t \mid \theta_k, y_{t-1}) = \log p(y_t \mid A_k, \Sigma_k, y_{t-1})$$
$$= \log \mathcal{N}(y_t \mid A_k y_{t-1}, \Sigma_k)$$
$$= -\frac{d}{2} \log(2\pi) - \frac{1}{2} \log|\Sigma_k| -$$
$$\frac{1}{2} (y_t - A_k y_{t-1})^T \Sigma_k^{-1} (y_t - A_k y_{t-1}). \quad (3.7)$$

We define the parameter space $\Theta = \{\theta_1, \theta_2, \ldots, \theta_K\}$ for all the hidden states, where $\theta_k = \{A_k, \Sigma_k\}$ for denoting the parameters of $k$-th state. Where $\theta_{z_t=k}$ 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. Therefore, given $M$ trained models for $S$ skills in the robot manipulation task ($M$ is equal to $S$ when using $k$-fold cross validation for optimal model selection), the optimal model of a skill is used along with the standard forward-backward algorithm to compute the expected cumulative likelihood of a sequence of observations.

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
3.4 Nonparametric Bayesian Method for Robot Movement Identification

state at any given moment, but this would neglect uncertainty. Instead, we use a
marginal probabilistic representation over hidden states at each time step. Thus,
the log-probability at time $t$ is derived by computing the logarithm of the sum of
exponentials over all hidden states

$$\mathcal{L}_t = \log \sum_{k=1}^{K} \exp \left( \frac{\alpha_t(k) \cdot \beta_t(k)}{p(y_{0:T})} \right),$$

(3.8)

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 mes-
sage passing and backward message in the standard forward-backward algorithm,
respectively.

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

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

(3.9)

The expected cumulative likelihood is $E[\log P(y_{1:T} | \Theta)]$ for each trained model
$m \in M$ using Eq. 3.8, where $\Theta_m$ represents the parameters of the trained model. That
is, given a test trial $\xi$, the cumulative log-likelihood is computed for test trial observa-
tions conditioned on all available trained skill model parameters $log P(y_{\xi_1:\xi_T} | \Theta)_m$
at a rate of 100 Hz. The process is repeated when a new skill $s \in S$ is started. Given the
observable position in the FSM $\xi_c$, we can index the correct log-likelihood $I(\xi_c \in s)$
and see if the probability density of the test trial given the correct model is greater
than the rest:

$$log P(y_{\xi_1:\xi_T} | \Theta_s) > log P(y_{\xi_1:\xi_T} | \Theta_m) \forall m (m \in M \land m \neq s).$$

(3.10)

If so, the skill identification is deemed correct and we record the time at which the
correct classification was flagged. We compute both a classification accuracy matrix
and the mean time threshold value by cross-validation period.

3.5 Experiments and Results

Results for nominal and anomalous skill identification are reported independently
for clarity. In the case of anomaly monitoring, the trials within the test folds for
skill identification were randomly intermixed. A 12-dimensional observation vec-
tor consisting of 6D wrench measurements and 6D pose values at each time step
$t$: $y_t = (f_x, f_y, f_z, m_x, m_y, m_z, x, y, z, r, p, y)$. All the parameters of the nonpara-
metric methods presented in this paper are set to the same values as those found in [5], when used for speech recognition classification.

An autoregressive order \( r = 2 \) is used. The parameter mean matrices \( M \) and \( K \) are set such that the mass of the matrix-normal distribution is centered around stable dynamic matrices while allowing variability in the matrix values. We start by assuming the mean matrix \( M = 0 \) and setting \( K = 10 \times I_m \). The inverse-Wishart portion of the prior is given by \( \nu_0 = m + 2 \text{DoF} \) (the smallest integer setting to maintain a proper prior). The scale matrix \( S_0 = 0.75 \times \) is the empirical covariance of the data set. Also, setting the prior from the data can move the distribution mass to reasonable parameter space values. A Gamma\((a, b)\) prior is set on the sHDP concentration parameters \( \alpha + \kappa \) and \( \gamma \). A Beta\((c, d)\) prior is set on the self-transition parameter \( \rho \). We choose a weekly informative setting and choose: \( a = 1, b = 0.01, c = 10, d = 1 \). Finally, the Gibbs sampling parameters truncation level and maximum iterations are set to 20 and 500 respectively.

In order to assess the robot introspection ability of the sHDP-VAR\((r)\)-HMM, two real-robot platforms were tested: (i) a Baxter robot for a pick-and-place task, as shown in Fig. 3.4 and (ii) a HIRO-NX robot for the snap assembly task, as shown in Fig. 3.9. Both robots used F/T sensors (Robotiq FT300 and a JR3 respectively) that were placed on the robot wrists to measure forces and moments at 100Hz. End-effector pose signals were computed using forward kinematics. Our workstation consisted of an 8-core Intel Xeon processor, 16GB RAM, running Linux Ubuntu 14.04 and ROS-Indigo.

### 3.5.1 Experiment 1: Baxter Robot Performs Pick-and-Place Task

The dual-arm Baxter robot is used for the pick-and-place task. The latter is bootstrapped via a finite state machine composed of four skills (see Fig. 3.4) and the goal is to use the sensed wrench and pose signals in building a statistical model for each skill through the proposed sHDP-VAR\((r)\)-HMM method. Figure 3.5 shows an example of the wrench signals for each skill (shown in different colors) of this task. The signals were segmented into four skills according to the states shown in the
3.5 Experiments and Results

Fig. 3.5 Experiment 1: recorded wrench signals in the pick-and-place task. Different colors encode different skills

finite state machine in Fig. 3.4. Notice that as the task is being executed, significant patterns emerge within and in-between skills. For instance, if we examine the signals for Go-to-Pick-Hover and Go-to-Pick shown in Fig. 3.4, we can identify patterns that can be encoded with statistical significance using our model. To evaluate the performance of the sHDP-VAR(r)-HMM models, we tested 24 samples for each skill through leave-one-out cross validation [35]. The optimal model for each skill was selected via maximum likelihood estimation.

For nominal skill identification, we test normally executing skills. Figure 3.6 shows corresponding signals generated on the basis that for $ith$ skill model output is 1 if its cumulative likelihood is greater than the rest of the models at each time step, else 0. If a model has a maximum output cumulative likelihood value during the last time step in a sample, then the value is set to 1, implying this observation belongs to that model class (the correct skill identification); otherwise, the firing would be 0. From the skill intervals shown in Fig. 3.5 and the models outputs of in Fig. 3.6, the monitoring accuracy for the sHDP-VAR(2)-HMM based method can be readily proposed. The same scheme can be inferred for the other skills.

We assessed the performance of the model by comparing with other similar techniques. Namely, the sticky HDP-HMM with Gaussian observation model and the sHDP-VAR(1)-HMM with a first order autoregressive model. We also show the performance by using different multimodal signal combinations. A confusion matrix is used for accuracy evaluation of skill identification for the different methods and shown in Fig. 3.8. Notice from Fig. 3.6, that at the beginning of each skill, the identifi-
cation suffers from low accuracies. This is due to the way in which we cumulatively compute the log-likelihood (not unlike humans that often need more information before confidently classifying). We compute the average test time to perform correct classification and report it as a percentage of total duration for a given skill (see Table 3.1). Low percentages imply quick identification, while large ones imply slow decisions.

For anomaly monitoring, we manually induced three types of external perturbations to collect anomaly data. First we introduce human collisions. Human collisions cause a pose displacement of a held object and leads to failures in the placement/assembly of that object. Five failure trials were induced for each perturbation, for a total of 15 anomalous trials (and 24 nominal trials). Receiver Operating Characteristic (ROC) curves were used to measure the discriminative ability of the system across wrench and pose sensor-modalities (i.e. wrench and wrench-pose modalities). Constant $k$ determines our classification threshold and is optimized to obtain the best monitoring performance. Comparisons between unimodal wrench signals and multimodal pose-wrench signals are conducted. Figure 3.7 shows ROC curves evaluate the relationship between the false positive rate (FPR) and true positive rate (TPR). For any given true-positive rate, multimodal tests resulted in lower false-positive rates when compared to unimodal versions. Our method also enjoyed high true-positive rate when compared to other introspective methods.

Fig. 3.6 Experiment 1: sHDP-VAR(2)-HMM model outputs for the pick-and-place task
3.5 Experiments and Results

Fig. 3.7  Experiment 1: Receiver operating characteristic (ROC) curves for the pick-and-place task. The figure shows ROC curves that compare the performance of multimodal and unimodal sensing for anomaly monitoring and the performance of our anomaly monitoring method versus other baseline methods.

Fig. 3.8  The identification accuracy of different methods with different multimodal signals in a pick-and-place experiment.

3.5.2  Experiment 2: HIRO-NX Robot Performs Snap Assembly Task

In this experiment, so as to show the applicability and superiority of the suggested sHDP-VAR-HMM monitoring scheme in a real-world assembly task of industrial relevance. A 6 DoF dual-arms HIRO robot with electric actuators and a Robotiq
Table 3.1  Average time for correct anomaly identification in a pick-and-place task. The notation “pPick” and “pPlace” are the abbreviation from “Pre-pick” and “Pre-place”, respectively

| Method                        | pPick (%) | Pik (%) | pPlace (%) | Place (%) |
|-------------------------------|-----------|---------|------------|-----------|
| sHDP-HMM (Wrench)             | 8.65      | 1.34    | 3.38       | 77.72     |
| sHDP-HMM (Wrench + Pose)      | 8.70      | 1.34    | 3.41       | 71.21     |
| sHDP-VAR(1)-HMM (Wrench)      | 3.62      | 1.33    | 3.45       | 61.10     |
| sHDP-VAR(1)-HMM (Wrench + Pose) | 3.60     | 1.40    | 3.41       | 36.28     |
| sHDP-VAR(2)-HMM (Wrench)      | 3.51      | 1.33    | 3.39       | 27.61     |
| sHDP-VAR(2)-HMM (Wrench + Pose) | 3.50     | 1.33    | 3.39       | 18.18     |

Fig. 3.9  Experiment 2: the skills in a snap assembly task FSM: i Approach; ii Rotation; iii Insertion; iv Mating

6DoF force-torque sensor attached on the wrist is used to perform a snap assembly task of camera parts as shown in Fig. 3.9. A custom end-effector holds a male part, while a female part is fixed to a table in front of the robot. The tool center point (TCP) is placed at the location where the male and female parts make contact. The world reference frame was located at the manipulator’s base. The TCP position and orientation were determined with reference to the world coordinate frame $To$. The force and moment reference frames were determined with respect to the wrist’s frame. OpenHRP [10] executes the FSM and modular hybrid pose-force-torque controllers execute the skills. Four nominal skills are connected by the FSM: (i) a guarded approach, (ii) an alignment procedure, (iii) a snap insertion with high elastic forces, and (iv) a mating procedure. Unexpected events occur during initial parts’ contact, (e.g. the wrong parts localization) or during the insertion stage where wedging is possible. An example of the recorded wrench signals is shown in Fig. 3.10.
As the procedure in experiment 1, forty-four real-robot nominal trials and 16 anomalous trials were conducted. We measured the skill identification and anomaly monitoring for the modeling schemes considered in experiment 2. The output of the proposed HDP-VAR(2)-HMM scheme is shown in Fig. 3.11. The identification accuracy per robot skill is presented in Fig. 3.13 and the average time for computing correct classifications is reported as a percentage of total skill duration (see Table 3.2). And
Fig. 3.12  Experiment 2: Receiver operating characteristic (ROC) curves for the snap assembly task across multimodal scenarios and baseline methods. Unimodal wrench signals are compared with multimodal pose and wrench signals.

Fig. 3.13  The identification accuracy of different method with variable sensor signals in snap assembly experiment.

as in experiment 1, we still compare performance with the same baseline methods. Meanwhile, the ROC curves confirm the anomaly monitoring performance which is illustrated in Fig. 3.12. It is thus seen, that the proposed process monitoring scheme has both accurate and efficient performance in contact tasks.
### 3.6 Summary

This chapter mainly focuses on the representation of robot complex, multi-step manipulation tasks and recognition of movement during online execution. Address the issue of task representation, we should give enough consideration to human understanding and intention in human-robot interactive manipulation tasks such that artificially segment complex tasks into multiple sub-tasks, and adopt dynamics that require only one demonstration movement of humans. The dynamical movement primitive (DMP) modeling method learns a parameterized primitive for each sub-task. When given a task, a combination of finite state machine and movement primitives is proposed to describe a complex manipulation task. The described sub-tasks can use different starting points and ending points than the demonstration as the motion basis. The input of the meta-model is used to generate new manipulation tasks, thereby improving the diversity and adaptability of manipulation tasks.

For the problem of robot movement recognition, a method of comparing the log-likelihood function gradient values of the observed values at the current moment under all the learned nonparametric Bayesian models is proposed. First, it is assumed that the manipulation task of the robot is decomposed into a set of movement primitives. Then, the multidimensional signals of each robot movement primitive under normal conditions is collected by repeatedly performing the manipulation task, and the probability model is established for each primitive by using the method of Chap. 2. Finally, the log-likelihood function gradient values of the observations under each model are evaluated in real time to realize the recognition of the movement behavior during the execution of the robot. Two evaluation indexes of recognition accuracy and recognition efficiency (response efficiency) of movement are applied to the proposed method on the two tasks of “HIRO-NX robot performing electronic component assembly tasks” and “Baxter robot performing autonomous pick and place tasks”. The comparisons of parametric and non-parametric models and various modalities combinations were performed, resulting in proofing the feasibility and effectiveness of the proposed movement recognition method in this chapter.

### Table 3.2 The average time of computing correct identification in snap assembly task

| Method                        | App (%) | Rot (%) | Ins (%) | Mat (%) |
|-------------------------------|---------|---------|---------|---------|
| sHDP-HMM (Wrench)            | 0.71    | 0.68    | 5.35    | 3.41    |
| sHDP-HMM (Wrench + Pose)     | 0.71    | 0.60    | 5.35    | 4.20    |
| sHDP-VAR(1)-HMM (Wrench)     | 0.71    | 0.71    | 5.35    | 0.89    |
| sHDP-VAR(1)-HMM (Wrench + Pose) | 0.71  | 0.60    | 5.33    | 0.81    |
| sHDP-VAR(2)-HMM (Wrench)     | 0.71    | 0.60    | 5.34    | 0.73    |
| sHDP-VAR(2)-HMM (Wrench + Pose) | 0.71  | 0.60    | 5.32    | 0.69    |
References

1. Pastor P, Hoffmann H, Asfour T, et al. Learning and generalization of motor skills by learning from demonstration[C]. In: 2009 ICRA’09 IEEE international conference on robotics and automation. IEEE;2009. p. 763—68.
2. Khansari-Zadeh SM, Billard A. Learning stable nonlinear dynamical systems with gaussian mixture models. IEEE Trans Robot. 2011;27(5):943–57.
3. Grollman DH, Jenkins OC. Incremental learning of subtasks from unsegmented demonstration[C]. In: 2010 IEEE/RSJ international conference on intelligent robots and systems (IROS). IEEE;2010. p. 261–66.
4. Manschitz S, Kober J, Gienger M, et al. Probabilistic progress prediction and sequencing of concurrent movement primitives[C]. In: 2015 IEEE/RSJ international conference on intelligent robots and systems (IROS). IEEE;2015. p. 449–55.
5. Fox EB. Bayesian nonparametric learning of complex dynamical phenomena[D]. Massachusetts Institute of Technology;2009.
6. Manschitz S, Gienger M, Kober J, et al. Probabilistic decomposition of sequential force interaction tasks into movement primitives[C]. 2016 IEEE/RSJ international conference on intelligent robots and systems (IROS). IEEE;2016. p. 3920-927.
7. Pastor P, Righetti L, Kalakrishnan M, et al. Online movement adaptation based on previous sensor experiences[C]. In: IEEE/RSJ international conference on intelligent robots and systems (IROS). IEEE;2011. p. 365-71.
8. Stulp F, Theodorou EA, Schaal S. Reinforcement learning with sequences of motion primitives for robust manipulation[J]. IEEE Trans Robot. 2012;28(6):1360–70.
9. Manschitz S, Kober J, Gienger M, et al. Learning to sequence movement primitives from demonstrations[C]. 2014 IEEE/RSJ international conference on intelligent robots and systems (IROS 2014). IEEE;2014. p. 4414–21.
10. Kanehiro F, Hirukawa H, Kajita S. Openhrp: Open architecture humanoid robotics platform. Int J Robot Res. 2004;23(2):155–65.
11. Duque DA, Prieto FA, Hoyos JG. Trajectory generation for robotic assembly operations using learning by demonstration. Robot Comput-Integr Manuf. 2019;1(57):292–302.
12. Calinon S. Robot programming by demonstration[M]. Lausanne: EPFL Press;2009.
13. Schaal S, Peters J, Nakanishi J, et al. Learning movement primitives[C]. Robotics research the eleventh international symposium. Berlin: Springer;2005. p. 561-72.
14. Jha A, Chiddarwar SS, Bhute RY, Alakashendra V, Nikhade G, Khandekar PM. Imitation Learning in Industrial Robots: A Kinematics based Trajectory Generation Framework. In: Proceedings of the advances in robotics 28 Jun 2017. p. 1–6.
15. Gribovskaya E, Khansari-Zadeh SM, Billard A. Learning non-linear multivariate dynamics of motion in robotic manipulators[J]. Int J Rob Res. 2011;30(1):80–117.
16. Calinon S, Billard A. A framework integrating statistical and social cues to teach a humanoid robot new skills[C]. In: Proceedings of the IEEE international conference on robotics and automation (ICRA), Workshop on social interaction with intelligent indoor robots;2008.
17. Muhlig M, Gienger M, Hellbach S, et al. Task-level imitation learning using variance-based movement optimization[C]. 2009 ICRA’09 IEEE international conference on robotics and automation. IEEE;2009. p. 1177-84.
18. Xu Z, Li S, Zhou X, Wu Y, Cheng T, Huang D. Dynamic neural networks based kinematic control for redundant manipulators with model uncertainties[J]. Neurocomputing. 2019;329(1):255–66.
19. Calinon S, Bruno D, Caldwell DG. A task-parameterized probabilistic model with minimal intervention control[C]. 2014 IEEE international conference on robotics and automation (ICRA). IEEE;2014. p. 3339–44.
20. Englert P, Paraschos A, Peters J, et al. Model-based imitation learning by probabilistic trajectory matching[C]. In: 2013 IEEE international conference on robotics and automation (ICRA). IEEE;2013. p. 1922–7.
References

21. Khansari-Zadeh SM, Billard A. Imitation learning of globally stable non-linear point-to-point robot motions using nonlinear programming[C]. In: 2010 IEEE/RSJ international conference on intelligent robots and systems (IROS). IEEE;2010. p. 2676–83.

22. Schaal S. Dynamic movement primitives-a framework for motor control in humans and humanoid robotics[M]. Adaptive motion of animals and machines. Springer;2006. p. 261–80.

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

24. Ispeert AJ, Nakanishi J, Hoffmann H, et al. Dynamical movement primitives: learning attractor models for motor behaviors[J]. Neural Comput. 2013;25(2):328–73.

25. Khansari M, Klingbeil E, Khatib Q. Adaptive human-inspired compliant contact primitives to perform surface-surface contact under uncertainty[J]. Int J Robot Res. 2016;35(13):1651–75.

26. Khansari-Zadeh SM, Khatib Q. Learning potential functions from human demonstrations with encapsulated dynamic and compliant behaviors[J]. Auton Robot. 2017;41(1):45–69.

27. Pastor P, Kalakrishnan M, Righetti L, et al. Towards associative skill memories[C]. In: 2012 12th IEEE-RAS international conference on humanoid robots (Humanoids). IEEE;2012. p. 309–15.

28. Niekum S, Osentoski S, Konidaris G, et al. Learning and generalization of complex tasks from unstructured demonstrations[C]. 2012 IEEE/RSJ international conference on intelligent robots and systems (IROS). IEEE;2012. p. 5239–46.

29. Niekum S, Osentoski S, Konidaris G, et al. Learning grounded finite-state representations from unstructured demonstrations[J]. Int J Robot Res. 2015;34(2):131–57.

30. Manschitz S, Gienger M, Kober J, et al. Mixture of attractors: a novel movement primitive representation for learning motor skills from demonstrations[J]. IEEE Robot Auto Lett. 2018;3(2):926–33.

31. 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 [J]. IEEE Trans Indus Electr Early Access;2020. https://doi.org/10.1109/TIE.2020.2970635.

32. Manschitz S, Kober J, Gienger M, et al. Learning movement primitive attractor goals and sequential skills from kinesthetic demonstrations[J]. Robot Auton Syst. 2015;74:97–107.

33. Su Z, Kroemer O, Loeb GE, Sukhatme GS, Schaal S. Learning manipulation graphs from demonstrations using multimodal sensory signals. In: 2018 IEEE international conference on robotics and automation (ICRA) 21 May 2018, IEEE. p. 2758–65.

34. Wu H, Guan Y, Rojas J. Analysis of multimodal Bayesian nonparametric autoregressive hidden Markov models for process monitoring in robotic contact tasks. Int J Adv Robot Syst. 2019;16(2):1729881419834840.

35. Andersen MR, Magnusson M, Jonasson J, Vehtari A, bayesian leave-one-out cross-validation for large data. In: Thirty-sixth international conference on machine learning;2019.

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.