

Xuefeng Zhou · Hongmin Wu ·
Juan Rojas · Zhihao Xu · Shuai Li

Nonparametric Bayesian Learning for Collaborative Robot Multimodal Introspection

OPEN ACCESS

 Springer

Nonparametric Bayesian Learning for Collaborative Robot Multimodal Introspection

Xuefeng Zhou · Hongmin Wu ·
Juan Rojas · Zhihao Xu · Shuai Li

Nonparametric Bayesian Learning for Collaborative Robot Multimodal Introspection



Springer

Xuefeng Zhou
Robotic Team Guangdong Institute
of Intelligent Manufacturing
Guangzhou, Guangdong, China

Juan Rojas
School of Electromechanical Engineering
Guangdong University of Technology
Guangzhou, China

Shuai Li
School of Engineering
Swansea University
Swansea, UK

Hongmin Wu
Robotic Team Guangdong Institute
of Intelligent Manufacturing
Guangzhou, Guangdong, China

Zhihao Xu
Robotic Team Guangdong Institute
of Intelligent Manufacturing
Guangzhou, Guangdong, China



ISBN 978-981-15-6262-4 ISBN 978-981-15-6263-1 (eBook)
<https://doi.org/10.1007/978-981-15-6263-1>

© The Editor(s) (if applicable) and The Author(s) 2020. This book is an open access publication.
Open Access This book 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 book are included in the book's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the book'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.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, expressed or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

This Springer imprint is published by the registered company Springer Nature Singapore Pte Ltd. The registered company address is: 152 Beach Road, #21-01/04 Gateway East, Singapore 189721, Singapore

To our ancestors and parents, as always

Preface

Improving collaborative robot safety by data-driven monitoring and protection is critical for robots to be actively useful in human daily life. While some success has been demonstrated in structured and static environments that robots are slow, clumsy, and not general or robust enough when performing dexterous manipulations. As humans, doubt and understanding our own limitations, failures, and shortcomings is a key for improvement and development. Such knowledge alters our behaviors e.g. to execute tasks in a more cautious way. The main reason is humans have years of experience to collect data and develop good internal models of what happens when interacting with their environment. Equipping robots with a set of skills that allows them to assess the quality of their sensory data, internal models, used methods, etc. is correspondingly believed to greatly improve the overall performance of an autonomous system.

Statistical models have shown powerful performance for signal processing in recent years. Particularly, the nonparametric Bayesian learning methods are widely applied to the study of underlying dynamical phenomena generated in fields as diverse as robotics, bioinformatics, econometrics, and autonomous systems and control. Within these fields, there has been an explosion of multimodal data of increasingly complex phenomena, resulting in a push toward building more intricate multivariate time series models for accessing the quality of those sensory data. In recent robotics, human-robot coexistence and cooperation have become an urgent demand in modern manufacturing and services industries, resulting in the concept of “Human-Robot Collaboration, HRC” and “Physical Human-Robot Interaction, pHRI”. Sharing intelligence, having a common behavior, and cooperating with people to accomplish common tasks (behavior, task, and intelligence) are their basic characteristics and elements, and also they have become the consensus of domestic and foreign academic circles. The prerequisite of HC is human-robot peaceful coexistence, that is, “safety collaboration”. Therefore, the aim of this book is to introduce the Nonparametric Bayesian learning methods for robot introspection that ensures the robot with longer-term autonomy and safer HRC environment.

In this book, the robot introspection relates to safety that has a direct impact on a variety of research areas, such as HRC, pHRI, long term autonomy, and many others. Long term autonomy can benefit from autonomous anomaly monitoring and diagnosis as well as the anomaly recovery strategies. The ability to reason and solve its own anomalies, and proactively enrich owned knowledge is a direct way to improve autonomous behaviors of a robot. To this end, we start by considering the underlying pattern of multimodal observation during robot manipulation, which can be well modeled as a parametric Hidden Markov Model (HMM). And then, we instead take a nonparametric Bayesian approach in defining a prior through the use of the Hierarchical Dirichlet Process (HDP) on the standard HMM parameters, named Hierarchical Dirichlet Process Hidden Markov Model (HDP-HMM). The HDP-HMM can examine an HMM with an unbounded number of possible states and allow for flexibility in the complexity of the learned model and for the development of reliable and scalable variational inference methods.

Therefore, although many stable and robust robot control algorithms exist, internal modeling errors or external disturbances still affect the robotic systems, such as human collision, object sliding, and tool collision. The pHRI scenarios are the unstructured and non-standardized dynamic environments that cannot be completely modeled and analyzed. To endow robots with longer-term autonomy and safer HRC environment, it is necessary to model the real-time multi-modal signals for implementing the accurate behavioral introspection and anomaly recovery policy learning. In this book, the robot multimodal introspection and learning are theoretically investigated, and the main contents and achievements are as follows:

Chapter 1—In this chapter, we concisely introduce the definition of robot introspection that is considered in this book, and explain those interesting topics to be addressed. Moreover, the detailed outline of this book is presented. To make the following contents clear and easy to follow, we take all the research contents and achievements of this book into consideration; a multi-modal fusion-based robot introspective and learning system framework named SPAIR (Sense-Plan-Act-Introspect-Recover) is proposed based on the traditional robot control framework Sense-Plan-Act (SPA), which adds the phases of Robot Introspection (behavior recognition, anomaly monitoring, anomaly diagnosis) and anomaly recovery. The framework intentionally endows the robot with longer-term autonomy and safer human-robot collaborative environment, which mainly includes four functional modules: (1) directed graph representation of complex robot manipulation tasks; (2) robot movement generation and generalization learning; (3) real-time anomaly monitoring and diagnosis during robot execution; (4) robot anomaly recovery policy learning.

Chapter 2—In this chapter, to address the problem of multi-modal fusion, we abstract its research object as to how to effectively establish and interpret the probabilistic model of multivariate time series. The uncertainty of the number of hidden states and the high transition frequency between hidden states are two critical problems of HMM in modeling multivariate time series, which will greatly weaken the modeling performance and time consistency. Based on this, several

nonparametric Bayesian models named Sticky Hierarchical Dirichlet Process Hidden Markov Models (sHDP-HMM) are investigated to jointly model the robot end-effector's velocity, force/torque, tactile signals of as well as their related statistical information (such as mean and variance) during robot manipulation task.

Chapter 3—In this chapter, to address the problem of learning and generalizing complex tasks of robots multimodal fusion and system framework construction, a method of combining Dynamic Movement Primitive (DMP) and Finite State Machine (FSM) is proposed in this paper, which divides the robot complex manipulation task into sequential motion primitives for improving the adaptability and diversity of the manipulation tasks, i.e. parameterized directed graph descriptions. The proposed method is based on the basic theory of robot learning from demonstration.

Chapter 4—In this chapter, nonparametric Bayesian models-based methods with the multi-modal fusion for robot real-time motion behavior recognition and anomaly monitoring are proposed. First, the normal multi-modal signals of each movement primitives are modeled using sHDP-HMM with the help of a parametric description of the robot's manipulation task. Then, the robot behavioral recognition is implemented by comparing the cumulative log-likelihood values of real-time observations. Finally, three types of abnormal thresholds for robot abnormal monitoring are proposed when the behavior is known, which include log-likelihood value, log-likelihood gradient value as well as mapping relationship between the latent state and log-likelihood value.

Chapter 5—In this chapter, nonparametric Bayesian models-based multi-objective classifiers for multi-modal anomaly diagnosis are also investigated when the multi-modal anomaly is triggered. Additionally, anomalous samples are extracted before and after the occurrence of anomaly events according to the size of the given window; sHDP-HMM model is learned for each anomaly type individually; the optimal model is selected by cross-validation method; anomaly diagnosis is tackled by comparing the cumulative log-likelihood values of testing samples across all learned models. Besides this, an anomaly diagnosis method by the multimodal sparse representation method of anomaly samples is presented.

Chapter 6—In this chapter, we mainly focus on learning the recovery policy based on anomaly monitoring and diagnosis. Typically, two task-level robot anomaly recovery policies are, respectively, proposed by learning the experience and intention of human treating accidental and persistent anomaly events, which include the re-enactment policy for accidental anomalies by using a multinomial distribution as well as the adaptation policy for persistent anomalies by the parametric representation of human's demonstrations.

In summary, this book mainly presents methods and algorithms for the collaborative robot multimodal introspection via nonparametric Bayesian methods, together with the corresponding theoretical analysis and experimental verification on two type of robots that perform three kinds of manipulation tasks, including HIRO-NX robot performing electronic assembly, Baxter robot performing pick-and-place task, and Baxter robot performing kitting experiment. This book is written for graduate students, academic, and industrial researchers in the field of

nonparametric Bayesian methods, robotics, physical human-robot interaction, multivariate time series modeling, and autonomous system design. This book covers the clear explanation of the importance of robot introspection in the tread the robotic techniques, included but not limited to the task representation of robot complex task, the multimodal time series modeling, the anomaly monitoring, and diagnosis as well as the anomaly recovery policy during robot manipulation task. It provides not only rigorous theoretical analysis on the nonparametric Bayesian based methods and algorithms for assessing the quality of multimodal sensory data but also many intuitive real-robot experiments with or without cooperating with a human. We do hope that this book will benefit the readers, in terms of knowing the basic ideas in the nonparametric Bayesian modeling of multivariate time series for robot introspection. We also hope that the presented new advancements in the field of anomaly monitoring, diagnosis, and recovery could explore new theoretical tools and practical applications.

At the end of this preface, it is worth pointing out that, in this book, some distributed methods for robot introspection and their applications are provided. The ideas in this book may trigger more studies and researches in next generation of robotics, especially nonparametric Bayesian methods based multivariate time series modeling. There is no doubt that this book can be extended. Any comments or suggestions are welcome, and the authors can be contacted via e-mail: hm.wu@giim.ac.cn (Hongmin Wu).

Guangzhou, China
Guangzhou, China
Hong Kong, China
Guangzhou, China
Swansea, UK
April 2020

Xuefeng Zhou
Hongmin Wu
Juan Rojas
Zhihao Xu
Shuai Li

Acknowledgements

During the work on this book, we have had the pleasure of discussing its various aspects and results with many cooperators and students. We highly appreciate their contributions, which particularly allowed us to improve our book manuscript.

The continuous support of our research is provided by the Foshan Key Technology Research Project (Grant No. 1920001001148), GDAS' Project of Thousand doctors (post-doctors) Introduction (2020GDASYL-20200103128), Basic and Applied Basic Research Project of Guangzhou (Grant No. 202002030237), Foshan Innovation and Entrepreneurship Team Project (Grant No. 2018IT100173), Guangdong Province Key Areas R&D Program (Grant No. 2019B090919002), Guangzhou Science Research Plan's Major Project (Grant No. 201804020095) Guangdong Innovative Talent Project of Young College (Grant No. 2016TQ03X463), Guangzhou Science and Technology Plan Project (Grant No. 201803010106).

Contents

1	Introduction to Robot Introspection	1
1.1	What is Robot Introspection?	1
1.2	Background and Significance	2
1.3	Issues to be Addressed in Robot Introspection	3
1.4	System Framework for Robot Multimodal Introspection	5
1.4.1	Introduction	5
1.4.2	Modules of Introspective System	6
1.5	Summary	8
	References	9
2	Nonparametric Bayesian Modeling of Multimodal Time Series	11
2.1	Introduction	11
2.2	Related Works	12
2.3	Theoretical Basis of Bayesian Model	14
2.3.1	Multimodal Time Series	14
2.3.2	Bayes' Rules	14
2.3.3	Conjugate Prior	15
2.4	Hidden Markov Model	19
2.4.1	Forward-Backward Algorithm	20
2.4.2	Viterbi Algorithm	21
2.5	Nonparametric Bayesian Hidden Markov Model	22
2.5.1	Bayesian Nonparametric Priors	23
2.5.2	The sHDP-VAR-HMM	25
2.6	Summary	26
	References	26
3	Incremental Learning Robot Task Representation and Identification	29
3.1	Introduction	29
3.2	Related Work	30

3.3	Graphical Representation of Robot Complex Task	32
3.3.1	Learning Graphical Representation from Unstructured Demonstration	34
3.3.2	Robot Movement Primitive Learning	35
3.4	Nonparametric Bayesian Method for Robot Movement Identification	37
3.4.1	Problem Statement	37
3.4.2	Gradient of Log-Likelihood for Movement Identification	37
3.5	Experiments and Results	39
3.5.1	Experiment 1: Baxter Robot Performs Pick-and-Place Task	40
3.5.2	Experiment 2: HIRO-NX Robot Performs Snap Assembly Task	43
3.6	Summary	47
	References	48
4	Nonparametric Bayesian Method for Robot Anomaly Monitoring	51
4.1	Introduction	51
4.2	Related Work	52
4.3	Problem Statement	53
4.4	Log-Likelihood for Anomaly Monitoring	54
4.4.1	Problem Formulation	54
4.4.2	Skill Identification	56
4.4.3	Anomaly Monitoring	57
4.4.4	Experiments and Results	58
4.4.5	Discussion	62
4.5	Gradient of Log-Likelihood for Anomaly Monitoring	63
4.5.1	The Hidden Markov Model Forward Gradient	64
4.5.2	Anomaly Monitoring Based on the Forward Gradient	68
4.5.3	Experiments and Results	69
4.5.4	Discussion	72
4.6	Mapping Latent State to Log-Likelihood for Anomaly Monitoring	73
4.6.1	Hierarchical Dirichlet Process Vector Auto-regressive Hidden Markov Models	73
4.6.2	Anomaly Monitoring Based on Hidden State	78
4.6.3	Experiments and Results	82
4.6.4	Discussion	89
4.7	Summary	90
	References	91

5 Nonparametric Bayesian Method for Robot Anomaly Diagnose	95
5.1 Introduction	95
5.2 Related Work	95
5.3 Problem Statement	97
5.4 Collection and Augmentation of Anomaly Sample	98
5.5 Anomaly Diagnose Based on Nonparametric Bayesian Model	100
5.5.1 Multiclass Classifier Modeling	101
5.5.2 Experiments and Results	103
5.5.3 Discussion	106
5.6 Anomaly Classifier Based on Feature Extraction	108
5.6.1 Anomaly Sample Collection	108
5.6.2 Anomaly Features Extraction	110
5.6.3 Experiments and Results	113
5.6.4 Discussion	115
5.7 Summary	115
References	116
6 Learning Policy for Robot Anomaly Recovery Based on Robot Introspection	119
6.1 Introduction	119
6.2 Related Work	120
6.3 Statement of Robot Anomaly Recovery	121
6.4 Learning and Application of Anomaly Recovery Policy	122
6.4.1 Learning for Unstructured Demonstrations	122
6.5 Reverse Execution Policy for Accidental Anomalies	124
6.6 Human Interaction Policy for Persistent Anomalies	125
6.7 Experiments and Results	127
6.7.1 Platform Setup	127
6.7.2 Parameter Settings for Anomaly Monitoring and Diagnose	130
6.7.3 Discussion	134
6.8 Summary	135
References	136

Acronyms

AR-HMM	Autoregressive Hidden Markov Model
BP-AR-HMM	Beta-process Prior on Autoregressive Hidden Markov Model
DMP	Dynamic Movement Primitive
DTW	Dynamic Time Wrapping
FSM	Finite State Machine
GD	Gradient-based Detector
GMM	Gaussian Mixture Model
HC	Human-Collisions
HCO	Human-Collision-with-Object
HCRF	Hidden State Conditional Random Field
HDP-HMM	Hierarchical Dirichlet Process Hidden Markov Model
HMM	Hidden Markov Model
HRC	Human-Robot Collaboration
HSD	Hidden-state Detector
HULM	Hidden Unit Logistic Model
KNN	K-nearest Neighbors
MAP	Maximum A Posteriori
NO	No-Object
OS	Object Slips
pHRI	Physical Human-Robot Interaction
sHDP-HMM	Sticky Hierarchical Dirichlet Process Hidden Markov Model
sHDP-VAR-HMM	Sticky Hierarchical Dirichlet Process Hidden Markov Model with Autoregressive Observation
SPA	Sense-Plan-Act
SPAIR	Sense-Plan-Act-Introspect-Recover
STAR-HMM	State-based Transition Autoregressive Hidden Markov Model
TC	Tool-Collisions
WC	Wall-Collision

Chapter 1

Introduction to Robot Introspection



Abstract In this chapter, we mainly introduce the definition, background, significance, and the start-of-the-art methods of collaborative robot multimodal introspection. The current issues of robot introspection are also introduced, which including the complex task representation, anomaly monitoring, diagnoses and recovery by assessing the quality of multimodal sensory data during robot manipulation. The overall content of this book is presented at the end.

1.1 What is Robot Introspection?

As humans, doubt and understanding our own limitations, failures and shortcomings is a key for improvement and development, such knowledge alters our behaviors e.g. to execute tasks in a more cautious way [1, 2]. Due to the humans can learn and recognize the environment for a long time through the five senses (visual, acoustical, gustatory, osphrethic, and tactile) and generate corresponding internal models in the daily execution of operational tasks [3–7]. Correspondingly, equipping robots with a set of skills that allows them to assess the quality of their sensory data, internal models, used methods etc. is correspondingly believed to greatly improve the autonomous operability and safety performance of an autonomous system, e.g. presented in literature [8, 9].

In recent years, the ability of robots to learn intrinsic introspective models like humans has been favored by robotics researchers [10, 11]. With respect to humans, introspection is the process of examining one's internal state by evaluating the potential patterns of multi-modal sensing data [12, 13, 15], as shown in Fig. 1.1. Its application lies in giving robots three capabilities: “What’s it doing?”, “How to do?”, and “How’s it doing?”. Among them, “What’s doing?” is to solve the problem of continuous state estimation during the robot execution; “How’s it doing?” is to monitor the real-time state during robot execution, such as normal or abnormal; “How to do?” including the robot task planning and decision making on future motion. The development and application of robot introspection can effectively

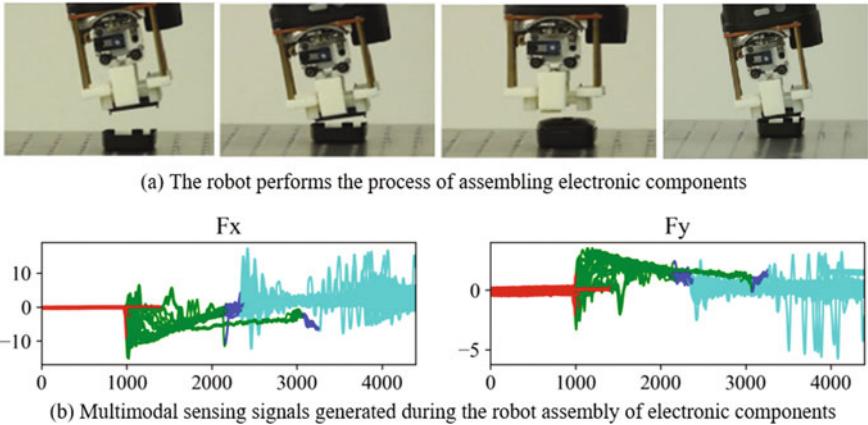


Fig. 1.1 Illustration of assessing the underlying dynamics of the multimodal signals that generated by a robot performing the electronic assembly at real-time

(1) estimate, monitor and prevent abnormal events; (2) evaluate the internal state of the robot; (3) strengthen the ability to recovery abnormalities; (4) optimize control and motion decisions.

However, robots do not have thoughts neither feelings, only data, hardware and algorithms. Therefore, robots can only assess the quality of sensor data, internal models, representations, information, perception input etc. Such knowledge can later lead to a modification of robots behavior by including the assessed quality score in the planning process. Consequently, robot introspection relates to safety, active perception, mapping and many topics. These topics have a direct impact on a variety of research areas, such as long term autonomy, search and rescue, and many others. Long term autonomy can benefit from autonomous failure recovery and active learning. For search and rescue, estimation of the confidence of the sensor input and used maps is essential for overall risk assessment. Moreover, for a large variety of tasks assessing the quality of sensor data, internal models, representation, information will directly affect mission success. The ability to reason and solve its own failures, and proactively enrich owned knowledge is a direct way to improve autonomous behaviours of a robot.

1.2 Background and Significance

Traditional industrial robots have played a significant role in standardization and automated production by isolating from human, which is difficult to meet unstructured, personalized, and flexible non-standard complex task requirements in a dynamic environment such as the human robot interactive scenarios that shown in Fig. 1.2. Due to they lack of multi-functional and intelligent capabilities that per-

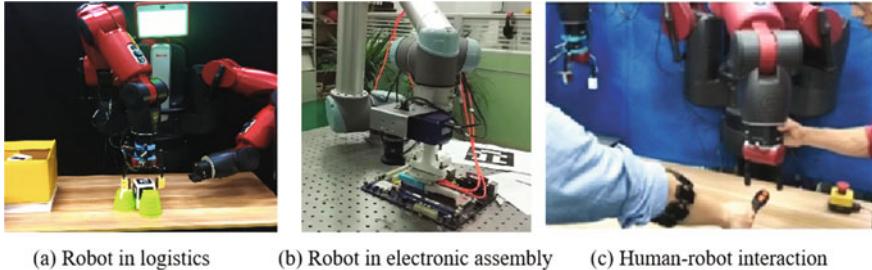


Fig. 1.2 Robots are applied in diverse, individualized and flexible non-standard tasks in unstructured dynamic environments

formed and without sufficient external sensing information. In addition, with the widespread application and development of collaborative robots [16, 17], robots will gradually move from a traditional closed manufacturing environment to a shared space that coexists and interacts with human [18], from semi-automatic operation tasks to autonomously perform, which inevitably leads to various unpredictable anomalies, such as objects slipping, collisions between end-effector and the environment, human collisions, and system abnormalities. Therefore, in order to give the robot a longer-term autonomy and a safer human-machine collaborative environment, the robot should perform real-time multi-modal fusion modeling to achieve accurate introspection of its own movement behavior (identification of movement behaviors, abnormal monitoring, and abnormal diagnoses), and anomaly recovery are essential in the next generation of intelligent collaborative robots.

Generally speaking, the robot follows the control framework of the Sense-Plan-Act (SPA) in a structured environment. First, the robot observes the surrounding environment and builds an internal model [14]. Then, it formulates a task execution plan, and finally executes this plan. However, the control framework is difficult to meet the increasingly complex robot operation tasks in unstructured dynamic environments [20, 23]. In the recent decades, increasing the robot multi-modal introspection and anomaly recovery policy after the robot execution would be an effective way to address the aforementioned issues. To this end, a novel Sense-Plan-Act-Introspect-Recover (SPAIR) control framework is proposed in this book, which can provide a theoretical framework and solutions for autonomous robot operation, human-computer interaction, and human-machine integration in the future.

1.3 Issues to be Addressed in Robot Introspection

This book intends to provide the reader with a comprehensive overview of the newly developed methods for improving robot introspection. In recent years, with the rapid development of collaborative robots, scholars in the field of robotics and artificial intelligence have carried out explorations on learning from human demonstration, integration with humans, multimodal sensing, and learning methods and theories

of robots. Although lots of research results and research ideas within those topics are presented, the difficulties and key issues of multi-modal modeling and its application of robots are not fully considered, and effective theoretical frameworks and systematic integration have not yet been formed. The application research of robot multi-modal sensing makes the research in this book more valuable. Therefore, this book develops a theoretical study of robot multi-modal introspection and learning based on non-parametric Bayesian models, aiming to endow robots a longer-term autonomy and a safer human-robot collaboration environment. With respect to the state-of-the-art status of robot introspection, the following five issues need to be addressed urgently:

- **(1) How to assess the quality of internal models, methods and sensor data, used by robots and how to alter their behavior upon this information?**

Humans can accurately perceive the state of the outside world or the objects they need to operate through the five senses based on past experience and internal models, and can self-recognize the advantages and disadvantages to learn and adjust existing experiences and models. However, robots can only sensing the world through sensory data such that how to model and analyze of multi-modal time series is a difficult problem to realize robot introspection.

- **(2) How to represent the complex robot task and generalize to the environmental changes during unstructured scenarios?**

It's difficult to complete tasks from the beginning to the end with fixed and pre-programmed movements in the case of an unstructured dynamic human-robot collaboration environment. Resulting in encountering fault or abnormality of the task because of the unstructured environment, the changing pose of the operating object, and the unpredictable robot state. Therefore, how to learn and generalize the robot complex tasks from human demonstration for adapting to the changes in the environment is an valuable question.

- **(3) How to implement the anomaly monitoring and diagnoses of multimodal time series during robot execution?**

The robot anomalies usually derived from joint encoders, the environmental changes, and human interference in the human robot interaction scenarios. Assuming that the robot can monitor and classify 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 the types of anomalies, it isn't possible to directly enumerate and model all anomalies, resulting the anomaly monitoring cannot be achieved through supervised learning methods. Generally, the robot anomaly monitoring were implemented by learning the outstanding difference between the normal ones. In other words, anomalies are identified from models of normal robot behavior. Making the robot anomaly monitoring and diagnoses should be critical factor for realizing the robot longer-term autonomy.

- **(4) How to learn the recovery policy from human demonstration for specific anomalous event?**

Robot anomaly recovery cannot be performed by the traditional robot's own motion planning algorithm under the human robot interaction scenarios. Whereas human expectations for robot motion should be taken into consideration. The main challenge is how to incrementally learn the robot recovery policy when encounter various abnormal events from unstructured human demonstrations.

- **(5) How to develop a stable and extendable software framework for integrating the introspective abilities by modeling the multimodal sensory data?**

How to develop a software for robot multimodal introspection that including multi-functional modules for complex task representation, multi-modal fusion, anomaly monitoring, anomaly diagnoses, and anomaly recovery. It's another common issue to be solved for realizing long-term autonomy and safe human robot collaborative environment.

1.4 System Framework for Robot Multimodal Introspection

1.4.1 Introduction

With the rapid development of collaborative robots, robots are moving from the traditional structured environment of industrial manufacturing to unstructured dynamic shared workspaces that coexist with, collaborate with, and integrate with people. Although the robot has a more robust and stable control algorithm, the algorithm cannot precisely model the unstructured environment, and unexpected events will occur. In order to endow the robot with a longer-term autonomy and a safer human-robot collaborative environment, the robot must implement self-monitoring and recovery capabilities by evaluating multi-modal sensory sensor in real time.

This book aim to propose a theoretical framework and extendable system platform for robot multimodal introspection and learning based on non-parametric Bayesian models. It mainly includes the following four aspects:

- Modeling of multimodal time series, details in Chap. 2;
- Learning and representation of robot complex task, details in Chap. 3;
- Multimodal anomaly monitoring, details in Chap. 4;
- Multimodal anomaly diagnose, details in Chap. 5;
- Heuristic incremental learning robot recovery policies, details in Chap. 6.

The proposed framework divides the complex tasks of the robot into the directed graph, and learns the motion primitive model and realizes the recognition of the motion behavior for the motion behavior between the nodes in the directed graph; The non-parametric Bayesian time series model learns models for anomaly monitoring

and anomaly diagnoses models, and compares and analyzes the performance and results of anomaly monitoring under three different anomaly threshold conditions. A robot anomaly recovery policy is designed at the top of the framework to handle anomalies encountered by external disturbances based on the representation of robot complex task and introspection. We assume the considered recovery policy should be learned incrementally from the context of task. Therefore, the proposed Re-enactment and Adaptation recovery policies correspond to accidental anomalies and persistent anomalies, respectively. Among them, the reenact policy uses a probability model of a polynomial distribution to count the humans decision-making at different anomaly situations for the accidental anomalies. Additionally, the robot learns and stores its recovery policy, while the motion adaptation uses human intuition to teach the recovery of persistent anomalies.

The anomalous event of robot is unpredictable, so that the recovery policy of the robot under different abnormal conditions cannot be planned offline. Therefore, learning the recovery policy incrementally is the key to achieving human-robot safety collaboration. The framework has fast and robust anomaly monitoring and diagnoses capabilities for both initially planned motion primitives and newly learned recovery motion primitives. In addition, by using non-parametric Bayesian models to incrementally learn models from multi-modal sensing data, the stability, reliability and fault tolerance of the framework are well improved.

1.4.2 *Modules of Introspective System*

To endow robots a longer-term autonomy and a safer human-robot collaboration environment, the robot's Introspection phase and the Recovery phase were added to the traditional robot control framework SPA. The system framework of the multi-modal sensing and learning is shown in Fig. 1.3 and named SPAIR. The framework mainly includes four modules: (1) directed graph representation of complex robot tasks; (2) generalization and identification of robot motion behavior; (3) real-time abnormal monitoring and diagnoses during robot execution; (4) learning recovery policy for recovering from robot abnormal events.

In recent years, the directed graph representation of complex tasks of robots has been investigated by a large number of scholars, such as Kappler et al. [19], Fox et al. [12], Kroemer et al. [21], and Niekum et al. [22]. To address the problem of robot task representation, many scholars tried to represent the complex task by a series of movement primitives, and realized the contact state estimation and task representation of the robot [13, 24–26]. This book assumes that the robot's tasks are represented by a directed manipulation graph that include a set of movement primitives, and each primitive is composed of nodes: the start node and the target node, and along with the transition between different movement primitives is expressed too. In order to accurately represent the robot complex tasks and the subsequent recovery behavior, the movement primitive is formulated as follows: (1) The behavior of the robot is used to represent a complete movement of the robot, as shown in Fig. 1.3.

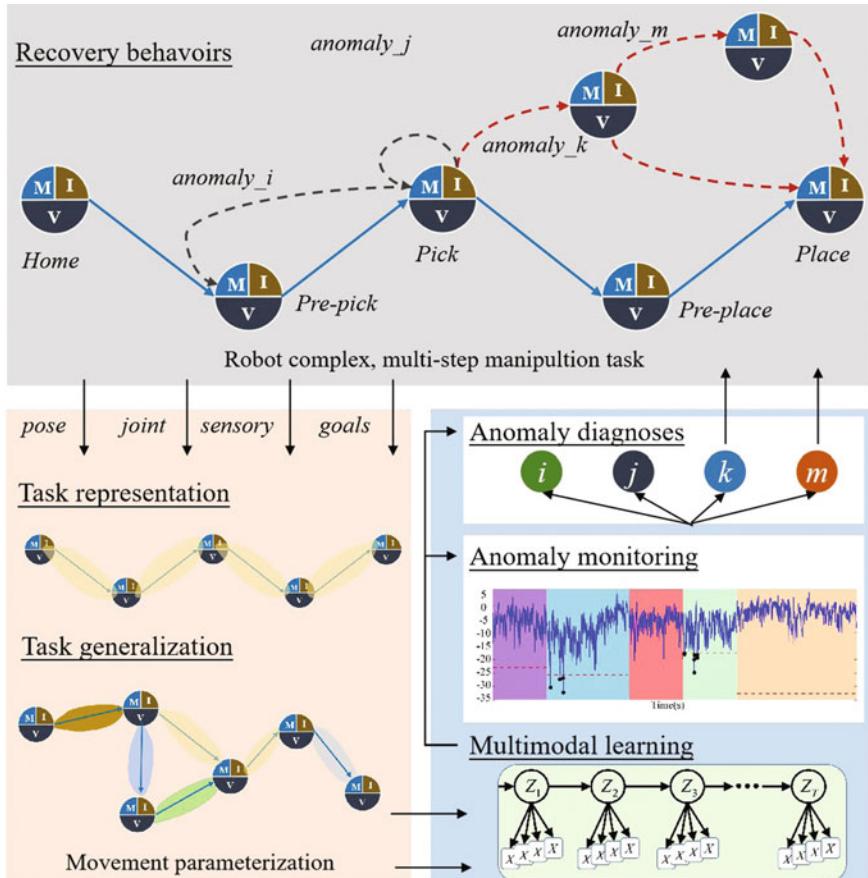


Fig. 1.3 A systematic framework for robot introspection and learning

The complete process from the “Home” node to the “Pre-pick” node; (2) The robot task is usually composed of one or more movement primitives, as shown in Fig. 1.3. The “Pick-and-Place” task of the robot consists of four motions. The behavior is composed of five action nodes, among which the movement primitives are “Home → Pre-pick”, “Pre-pick → Pick”, “Pick → Pre-place” and “Pre-place → Place”.

In particular, the directed manipulation graph will be updated when the robot encounter an abnormal event, which including: \mathcal{N}_{ij} indicates a recovery node between nodes \mathcal{N}_i and \mathcal{N}_j and node \mathcal{N}_{ijk} denotes a recovery node between nodes \mathcal{N}_{ij} and \mathcal{N}_k . By analogy, nodes for recovery will be generated in the original graph structure, and the connection between the nodes (including the recovery node and the node of the original task) will generate new motion behavior. From the recovery module at the top of Fig. 1.3, it can be known that the manipulation graph \mathcal{G} of the robot tasks will gradually robust and stable with the increasing recovery behaviors.

The nodes in the SPAIR framework will include multiple functions performed in parallel, including: the robot movement properties \mathcal{M} , the robot introspective properties \mathcal{I} , and the robot sensing external environment properties \mathcal{V} . The \mathcal{M} in this book refer to the robot movement primitives; (\mathcal{I}) includes the robot's movement identification, anomaly monitoring, and anomaly diagnoses; (\mathcal{V}) represent the visual estimation of the object's pose in the environment. Among them, the robot's introspective attributes are mainly responsible for real-time monitoring of robot anomalies. Once anomalies are detected, the system will classify the anomalies, and the system will recovery according to a limited recovery strategy for different types of anomalies. If the recovery strategy does not exist, The system will prompt the need for human recovery demonstration, and learn and update its recovery behavior. According to the diagnoses result, if it is a accidental anomaly, the robot will perform the motion redo recovery behavior, as shown in “*anomaly – i*” and “*anomaly – j*” shown in Fig. 1.3; if it is a persistent anomaly, the robot will execute the motion adaptation recovery behavior, as shown in “*anomaly – k*” and “*anomaly – m*” in Fig. 1.3.

The proposed framework is built on the *ROS-Indigo*¹ platform, allowing signals of different modalities to communicate in the form of topics, and the sampling frequency of different sensors is fixed to a specific value by synchronous sampling (this book uses 10Hz), and combines multiple open source software and code, such as the description of the robotic task by the finite state machine *SMACH*² and the *ar_track_alvar*³ for pose estimation of the object. In addition, the system will automatically collect multi-modal sensor signal data for each motion primitive and the subsequent proposed recovery behaviors during the robot's movement. So that the robot can learn experience and generalize movement from the generated data. To this end, the relevant code, videos, pictures and usage documents of this book will be all open source,⁴ so that others researchers can improve and share it.

1.5 Summary

In this chapter, we mainly introduce the definition and significance of robot introspection in the background of human-robot collaboration. We inspired from human that doubt and understanding our own limitations, failures and shortcomings is a key for improvement and development, such knowledge alters our behaviors e.g.to execute tasks in a more cautious way. To endow robots a longer-term autonomy and a safer human-robot collaboration environment, we referred to the state-of-the-art status of robot introspection, five issues are needed to be addressed urgently. To this end, we proposed an introspective system framework with four modules, named Sense-Plan-

¹<http://wiki.ros.org/indigo>.

²<http://wiki.ros.org/smach>.

³http://wiki.ros.org/ar_track_alvar.

⁴https://github.com/birlrobotics/smach_based_introspection_framework.

Act-Introspect-Recover (SPAIR) for address the problems including robot complex task graphical representation and execution (skill) identification from unstructured demonstrations, multimodel time series modelling for robot anomaly monitoring and diagnose as well as the recovery policies learning for different anomalies.

References

1. LaViers A. Make robot motions natural. *Nature*. 2019;565:422–4.
2. Les Z, Les M. Human perception. Machine understanding. Cham: Springer; 2020. p. 1–8.
3. Teixeira T, Dublon G, Savvides A. A survey of human-sensing: methods for detecting presence, count, location, track, and identity. *ACM Comput Surv*. 2010;5(1):59–69.
4. Sharma R, Pavlović VI, Huang TS. Toward multimodal human-computer interface. Advances in image processing and understanding: a festschrift for Thomas S Huang. Singapore: World Scientific; 2002. p. 349–365.
5. Pan B, Zheng Y, Wilkie D, et al. Crowd sensing of traffic anomalies based on human mobility and social media. In: Proceedings of the 21st ACM SIGSPATIAL international conference on advances in geographic information systems. ACM; 2013. p. 344–353.
6. Fong T, Nourbakhsh I, Dautenhahn K. A survey of socially interactive robots. *Robot Auton Syst*. 2003;42(3–4):143–66.
7. Jaimes A, Sebe N. Multimodal human-computer interaction: a survey. *Comput Vis Image Underst*. 2007;108(1–2):116–34.
8. Cifuentes CA, Rodriguez C, Frizeraneto A, et al. Multimodal human-robot interaction for walker-assisted gait; 2017 10(3):933–943.
9. de Gea Fernández J, Mronga D, Günther M, et al. Multimodal sensor-based whole-body control for human-robot collaboration in industrial settings. *Robot Auton Syst*. 2017; 94:102–119.
10. Do HM, Pham M, Sheng W, et al. RiSH: a robot-integrated smart home for elderly care. *Robot Auton Syst*. 2017; 101:74–92.
11. Gao ZD, Su JB, Zhou W. Robot autonomous perception model for internet-based intelligent robotic system. In: IEEE proceedings of international conference on machine learning and cybernetics; 2005.
12. Fox M, Ghallab M, Infantes G, et al. Robot introspection through learned hidden markov models. *Artif Intell*. 2006;170(2):59–113.
13. Rojas J, Luo S, Zhu D, et al. Online robot introspection via wrench-based action grammars. 2017 IEEE-RAS international conference on robotics and system. IEEE; 2017. p. 113–119.
14. 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.
15. Wu H, Lin H, Guan Y, et al. Robot introspection with Bayesian nonparametric vector autoregressive hidden Markov models. 2017 IEEE-RAS 17th international conference on humanoid robotics (Humanoids). IEEE; 2017. p. 882–888.
16. Bragana S, Costa E, Castellucci I, Arezes PM. A brief overview of the use of collaborative robots in industry 4.0: human role and safety. In: Occupational and environmental safety and health. Cham: Springer; 2019. p. 641–650
17. Goldberg K. Robots and the return to collaborative intelligence. *Nat Mach Intell*. 2019; 1(1):2.
18. Cherubini A, Passama R, Crozier A, Lasnier A, Fraisse P. Collaborative manufacturing with physical human-robot interaction. *Robot Comput-Integr Manuf*. 2016; 1(40):1–3.
19. Kappler D, Pastor P, Kalakrishnan M, et al. Data-driven online decision making for autonomous manipulation. In: Robotics: science and systems; 2015.
20. 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.

21. Kroemer O, Daniel C, Neumann G, et al. Towards learning hierarchical skills for multi-phase manipulation tasks. In: 2015 IEEE international conference on robotics and automation (ICRA). IEEE, 2015. p. 1503–1510.
22. Scott N, Sarah O, George K, et al. Learning grounded finite state representations from unstructured demonstrations. *Int J Robot Res.* 2015; 34(2):131–57.
23. 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>.
24. Rojas J, Huang Z, Harada K. Robot contact task state estimation via position-based action grammars. In: IEEE-RAS 16th international conference on humanoid robots; 2016. p. 249–255.
25. Rojas J, Newman W, Jie H, et al. A steering wheel manipulation scheme by an anthropomorphic humanoid robot in a constrained vehicle environment. In: IEEE international conference on robotics and biomimetics; 2015. p. 1566–71.
26. Rojas J, Harada K, Onda H, et al. Contextualized early failure characterization of cantilever snap assemblies. In: The 14th IEEE-RAS international conference on humanoid robots; 2014. p. 380–387.

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.



Chapter 2

Nonparametric Bayesian Modeling of Multimodal Time Series



Abstract In this chapter, we take a Bayesian nonparametric approach in defining a prior on the hidden Markov model that allows for flexibility in addressing the problem of modeling the complex dynamics during robot manipulation task. At first, considering the underlying dynamics that can be well-modeled as a hidden discrete Markov process, but in which there is uncertainty about the cardinality of the state space. Through the use of the hierarchical Dirichlet process (HDP), one can examine an HMM with an unbounded number of possible states. Subsequently, the sticky HDP-HMM is investigated for allowing more robust learning of the complex dynamics through a learned bias by increasing the probability of self-transitions. Additionally, although the HDP-HMM and its sticky extension are very flexible time series models, they make a strong Markovian assumption that observations are conditionally independent given the discrete HMM state. This assumption is often insufficient for capturing the temporal dependencies of the observations in real data. To address this issue, we consider extensions of the sticky HDP-HMM for learning the switching dynamical processes with switching linear dynamical system. In the later chapters of this book, we will verify the performances in modeling multimodal time series and present the results of robot movement identification, anomaly monitoring, and anomaly diagnose.

2.1 Introduction

Human can perceive the state of the outside world or the objects that need to be operated through the five perspectives based on past experience and internal models, and can recognize the advantages and disadvantages of self to learn and adjust existing experiences and models. However, robots can only understand the outside world through sensing data, so modeling and analysis of multi-modal time series is a difficult and hot research topic to realize robot perception. The multi-modal information during the execution of the robot is a highly complex and dynamic process that incorporates non-linearities such as motion encoders, vision, forces/torques, haptics, and sounds. Modeling based on the traditional hidden Markov model has the problems of uncertainty of the number of recessive states and rapid conversion of recessive states,

and it is difficult to accurately capture the potential modes of multi-modal data. In this chapter, based on the Bayesian basic theory and the theoretical derivation of the HMM model, combined with the inherent characteristics of the robot's multi-modal sensor information, it proposes that it is more efficient and complicated in terms of complex dynamic phenomena, uncertainty, and model parameter learning. The robust non-parameterized Bayesian model implements a solution in which the number of recessive states of the HMM model is determined by the complexity of the data and a high self-transition expected probability of the recessive state. To a certain extent, the non-parameterized Bayesian model is Research on robot multi-modal information sensing and fusion provides theoretical framework and application guidance.

2.2 Related Works

In neurosciences, studies have shown that the different senses of humans can be closely combined [1], so that they can implement the complex manipulation tasks and environmental perception in daily life. In recent years, inspired by the results of this research in neurosciences, scholars have carried out research on the role of multimodal sensing information fusion for implementing practical manipulation tasks in robotics [2, 3]. Fitzpatrick et al. [4] introduced a method of cross-modal verification, which enhances the robot's perception of multi-modal events by repeating and redundancy of sensing information, so that the robot can learn the underlying correlation between vision and auditory. Wu et al. [5] studied the use of a combination of acceleration and sound measurements to detect structural defects in aircraft parts. Su et al. [6] introduced a multi-modal event detection framework based on the robot's axis hole assembly task, including the robot's end pose and haptic sensing signals. It can be known from aforementioned instances that multimodal fusion refers to the synthesis of sensory data with a certain temporal sequence and spatial relationship from multiple different sensors, modeling at different levels and learning its potential patterns [7]. Then, the essence of multimodal fusion is to realize the modeling and analysis of Multivariate Time Series (MTS). For example, in order to allow the robot can perform flexible operations in an uncertain environment, a force/torque sensor is mounted on the end-effector [8], a tactile sensor and a acoustical sensor are installed on the robot claw, and a visual sensor is placed for monitoring the robot workspace [9–11], as shown in Fig. 2.1. In this way, the multi-sensor fusion method can be used to more comprehensively and accurately monitor environmental characteristics, eliminate signal variability and information uncertainty, and improve system reliability and stability [7].

Time series modeling is pervasive in fields as diverse as speech recognition, robotics, economics, medicine, multimedia, bioinformatics, and system control [12]. In recent years, related modeling methods have emerged endlessly, among which the Hidden Markov Model (HMM) [13] is the most popular method that were used in speech recognition [14–16], computational biology [17, 18], Machine translation [19, 20], cryptanalysis [21, 22], economics [23, 24], and human behavior recogni-

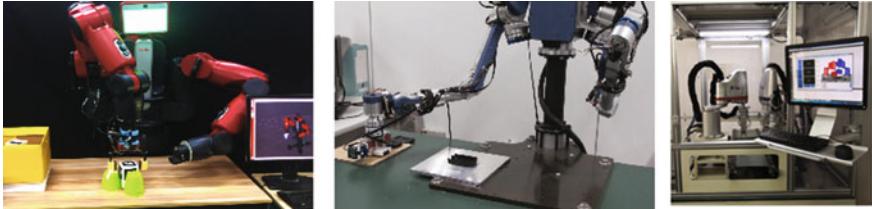


Fig. 2.1 Robotic manipulating platforms integrated with multiple sensors

tion [25, 26], etc. In particular, HMM has been widely used in the field of robotics, mainly including process supervision [27, 28], robot state estimation [29, 30], decision making [31, 32], and learning robot motion primitive transformation [33, 34], etc. In literature [27] indicated that HMM can be used for anomaly monitoring, providing a solution for the follow-up work of this book in anomaly implementation. Although there have been a lot of successful application cases of HMM, this model has two inherent shortcomings [35]: (1) It's difficult to determine the size of the HMM hidden state space (the number of hidden states); (2) HMM follows the Markov chain hypothesis (in a given hidden state, the observation data are distributed independently of each other) and the traditional Expectation Maximization (EM) algorithm is used to infer unknown parameters [36, 37]. This inherent disadvantage greatly weakens the accuracy and efficiency of HMM modeling in time series data.

To address the problems of HMM, non-parameterized Bayesian model [38] and HMM of infinite hidden state space [39] were proposed. Yee Whye The et al. [40] proposed a Hierarchical Dirichlet Process Hidden Markov Model (HDP-HMM). HDP-HMM uses the hidden state transition probability distribution and observation distribution of HMM. The methods of introducing hyper-parameters to Bayesian a priori are added separately, so that the number of hidden states can be learned from the complexity of the training data, and is calculated by the Markov Chain Monte Carlo (MCMC) method. The Gibbs Sampling (GS) method infers the unknown parameters of the model [41, 42]. HDP-HMM uses the full Bayesian method to learn the number of hidden states by assuming an infinite state space, it effectively avoids overfitting the data and repeated trial and error model selection. From the process of the model implementation and experimental results, it can be known that HDP-HMM's modeling of hidden states still retains the properties of the original Markov chain, which will cause rapid transitions between hidden states, resulting in a lack of the principle of time continuity. The so-called time continuity means that the occurrence of events can always maintain a certain consistency, which makes it difficult to analyze events in the real world. To address the problems of HDP-HMM, Fox et al. [43–45] proposed a Sticky Hierarchical Dirichlet Process Hidden Markov Model (sHDP-HMM). The sHDP-HMM tackle the rapid transition of hidden states in HMM is to increase the self-transition of hidden states by adding the bias of the probability of each hidden state transition, so as to meet the requirement of time continuity. The experimental

results show that the “sticky” hidden state enhances the consistency modeling of time series data, and can achieve more accurate results than HDP-HMM in speech recognition and multiple speaker recognition experiments [43].

2.3 Theoretical Basis of Bayesian Model

2.3.1 Multimodal Time Series

With the development of multi-modal sensing fusion technology and the diversity of the environment, except the joint encoders, robots often need to add force/torque sensors and tactile sensors to sense the surrounding environment, so that the robot’s observation is often multimodal. In general, a multidimensional time series with dimension D and length T is represented as follows:

$$Y_{T \times D} = [y_1(t), y_2(t), \dots, y_d(t), \dots, y_D(t)], t = 1, 2, \dots, T; 1 \leq d \leq D \quad (2.1)$$

where, t represents a time frame, d represents a variable, and $y_d(t)$ denotes the observation of the d -dimensional variable at the t -th time step. For ease of writing, define the observation vectors for all dimensions at time t as $y_t \in \mathbb{R}^D$. Hence, the Eq. 2.1 is expressed as a matrix with $T \times D$, and the matrix is set as an example of a multidimensional time series, that is,

$$Y_{T \times D} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_T \end{bmatrix} = \begin{bmatrix} y_1(1) & y_2(1) & \dots & y_d(1) & \dots & y_D(1) \\ y_1(2) & y_2(2) & \dots & y_d(2) & \dots & y_D(2) \\ \vdots & \vdots & \dots & \vdots & \dots & \vdots \\ y_1(T) & y_2(T) & \dots & y_d(T) & \dots & y_D(T) \end{bmatrix} \quad (2.2)$$

Each row in Eq. 2.2 represents the observation at time t , and each column represents the data of a signal. This book uses a non-parametric Bayesian hidden Markov model to learn and analyze the multi-dimensional time series [45, 46].

2.3.2 Bayes’ Rules

Assuming two random variables are y and θ respectively, then according to the definition of Bayes’ rule as follows:

$$p(\theta|y) = \frac{p(y|\theta)p(\theta)}{p(y)} \quad (2.3)$$

Where $p(\theta|y)$ is the posterior probability, $p(y|\theta)$ is the likelihood probability, $p(\theta)$ is the prior probability, and $p(y)$ is the standardized constant. From Eq. 2.3, the posterior probability is proportional to the product of the likelihood probability and the prior probability. Bayesian law summarizes the process of Bayesian inference: given the likelihood probability and the standardized constant, the posterior probability can be inferred from the prior probability. The Bayesian time series model mentioned in this book means that the variables in the model are time-dependent, but due to the endless time correlation, the model cannot be established, so it is proposed that the variables require certain constraints are time-independent such as Markov chain constraints. So that two adjacent variables y_{t+1} and y_t can be predicted by Eq. 2.3

$$p(y_{t+1}|y_t) = \frac{p(y_t|y_{t+1})p(y_{t+1})}{p(y_t)} \quad (2.4)$$

Therefore, if y in Eq. 2.3 is expressed as the random variable and θ as parameters of corresponding probability model, where unknown parameter is accompanied by a hyper-parameter, Bayesian theory can be extended to

$$p(\theta|y, \lambda) = \frac{p(y|\theta)p(\theta|\lambda)}{p(y)} \quad (2.5)$$

From Eq. 2.5, we can know that the parameters of the model are estimated by maximizing the posterior probability, that is, the maximum posterior probability estimation method (Maximum a Posteriori, MAP), i.e.

$$\theta_{MAP} = \arg \max_{\theta} p(\theta|y, \lambda) \quad (2.6)$$

If the prior probability $p(\theta|\lambda)$ is a constant, then the maximum posterior estimate is equivalent to the Maximum Likelihood (ML), that is,

$$\theta_{ML} = \arg \max_{\theta} p(y|\theta) \quad (2.7)$$

Equations 2.6 and 2.7 are both point estimation methods for model parameters, which will widely used to the later chapters.

2.3.3 Conjugate Prior

As we all know, the posterior distribution and the prior distribution belong to the same type in Bayes' rule such that the prior distribution and the posterior distribution are called conjugate distributions. Where the prior distribution is called the conjugate prior of the likelihood function. Therefore, the used of conjugate priors is often motivated by practical considerations that the parameters θ can't be directly

learned from the observations. Namely, the conjugate priors allow for introducing a computationally tractable mechanism for incorporating new observations into the posterior distribution of the parameters θ . With the Bayesian framework, we are interested in incorporating a prior distribution on the latent model parameter θ for making predication about the new observations. To this end, assuming the associated conditional exist, and given N i.i.d. observations, this predictive likelihood is written by

$$p(y|y_1, \dots, y_N, \lambda) = \int_{\Theta} p(y|\vartheta a)p(\vartheta|y_1, \dots, y_N, \lambda)d\vartheta \quad (2.8)$$

To simplify the computational complexity, we only discusses conjugate priors that require the use of three similarity functions in subsequent chapters. In particular, three common observation model are considered in the hidden Markov model, including Multinomial Distribution, Multivariate Gaussian Distribution as well as Linear Gaussian Regression Model.

• Multinomial Observations

Multinomial Distribution [130, 131] considers a random variable y on a finite sample space $\mathcal{Y} = \{1, \dots, K\}$, its probability mass function can be denoted by $\pi = [\pi_1, \dots, \pi_K]$, and describes the probability of a string of N observations of y taking on values y_1, \dots, y_n by

$$p(y_1, y_2, \dots, y_N | \pi) = \frac{N!}{\prod_k N_k!} \prod_k \pi_k^{N_k}, N_k \triangleq \sum_n \delta(y_n, k) \quad (2.9)$$

Where, the notation $\delta(j, k)$ indicate the discrete Kronecker delta. When $K = 2$, this distribution is referred to as the binomial distribution.

The Dirichlet prior distribution is used for formulating the conjugate prior of the multinomial distribution. A K -dimentional Dirichlet distribution is the conjugate prior for the class of K -dimensional multinominal distribution and is uniquely defined by a set of hyperparameters $\alpha = [\alpha_1, \dots, \alpha_K]$, which has the following form:

$$\begin{aligned} \text{Dir}(\alpha_1, \alpha_2, \dots, \alpha_K) &= p(\pi | \alpha) \\ &= \frac{\Gamma(\sum_k \alpha_k)}{\prod_k \Gamma(\alpha_k)} \prod_k \pi_k^{\alpha_k - 1}, \alpha_k > 0 \end{aligned} \quad (2.10)$$

Where, $\Gamma(\cdot)$ represents the standard Gamma function. When $K = 2$, this distribution is referred to as the Beta distribution, which can be denoted by $Beta(\alpha_1, \alpha_2)$. The initial value of the distribution in Eq. 2.10 is given by

$$[\pi_i] = \frac{\alpha_i}{\sum_k \alpha_k} \quad (2.11)$$

Then, refer to the conjugacy of the Dirichlet distribution, conditioned on N multinomial observations y_1, \dots, y_N , the posterior distribution on π is also Dirichlet that

formulated by

$$p(\pi|y_1, y_2, \dots, y_N, \alpha) \propto p(\pi|\alpha)p(y_1, y_2, \dots, y_N|\pi) \propto Dir(\alpha_1 + N_1, \dots, \alpha_K + N_K) \quad (2.12)$$

Where, the N_k represents the observed times of observation $y_n = k$. Subsequently, using the normalizing constant of the Dirichlet distribution, and substituting to Eq. 2.8, one can derive the predictive likelihood to be

$$p(y = k|y_1, \dots, y_N, \alpha) = \frac{N_k + \alpha_k}{N + \sum_{k=1}^K \alpha_k} \quad (2.13)$$

• Multivariate Gaussian Observations

Multivariate Gaussian distribution is parameterized by a mean vector μ and covariance matrix Σ , which provides a useful description of continuous-valued random variables that concentrate about a given value and have constrained variability [130]. This distribution over a sample space $y \in \mathcal{Y} = \mathbb{R}^D$ that each observation y is a D -dimensional vector can be derived by

$$\mathcal{N}(y; \mu, \Sigma) = p(y|\mu, \Sigma) = \frac{1}{(2\pi)^{D/2} |\Sigma|^{1/2}} \exp \left\{ -\frac{1}{2} (y - \mu)^T \Sigma^{-1} (y - \mu) \right\} \quad (2.14)$$

We conveniently denote this multivariate Gaussian distribution by $\mathcal{N}(\mu, \Sigma)$. Three categories of priors about this distribution in terms of the parameters are considered in this book, including known covariance, known mean, and both the mean and covariance are uncertain.

Known Covariance

For known covariance Σ , we use the normal prior distribution as the conjugate prior on the mean vector μ , and represented by $\mathcal{N}(\mu_0, \Sigma_0)$.

Known Mean

For known mean and only the covariance Σ is uncertain, the conjugate prior is formulated by the inverse-Wishart distribution [130]. The D -dimensional inverse-Wishart distribution with two parameters: covariance parameter Δ and v degrees of freedom, and given by

$$p(\Sigma|\Delta, v) = \frac{|\nu\Delta|^{\frac{v}{2}} |\Sigma|^{\frac{v+D+1}{2}}}{2^{\frac{vD}{2}} \pi^{\frac{D(D-1)}{4}} \prod_{i=1}^k \Gamma(\frac{v+1-i}{2})} \exp \left\{ -\frac{1}{2} \text{tr}(\nu\Delta\Sigma^{-1}) \right\} \quad (2.15)$$

Where, $\text{tr}(\cdot)$ denotes the trace of matrix. We write this distribution by $\mathcal{IW}(v, \Delta)$, and the first moment is given by calculating the expectation value:

$$E[\Sigma] = \frac{\nu\Delta}{v - D - 1} \quad (2.16)$$

Both Covariance and Mean are Uncertain

We use the normal-inverse-Wishart distribution as the conjugate prior when both the mean and covariance are uncertain. This distribution defines a conditionally normal prior on the mean $\mu|\Sigma \sim \mathcal{N}(\vartheta, \Sigma/\kappa)$, and an inverse-Wishart distribution on the covariance, $\Sigma \sim \mathcal{IW}(v, \Delta)$. Therefore, the joint prior for uncertain mean and covariance is then given by

$$p(\mu, \Sigma|\kappa, \vartheta, v, \Delta) \propto |\Sigma|^{\frac{v+d+1}{2}} \exp\left\{\frac{1}{2} \text{tr}(\nu \Delta \Sigma^{-1}) - \frac{\kappa}{2} (\mu - \vartheta)^T \Sigma^{-1} (\mu - \vartheta)\right\} \quad (2.17)$$

Where, κ represents the degree of trust in the mean generated by this prior distribution, ϑ represents the mean of this prior distribution, v represents the degree of trust in the variance generated by this prior distribution, and Δ represents the mean of the variance Σ matrix. We will use the notation $\mathcal{NIW}(\kappa, \vartheta, v, \Delta)$ to represent the normal-inverse-Wishart distribution with four parameters.

• Multivariate Linear Gaussian Regression Observations

The normal multivariate linear regression model is one in which the observations $y_i \in \mathbb{R}^d$ can be described as a linear combination of a set of known regressors $x_i \in \mathbb{R}^n$ with errors accounted for by additive Gaussian noise, written as

$$y_i = x_{i1}a_1 + \cdots + x_{in}a_n + e_i, \quad e_i \sim \mathcal{N}(0, \Sigma) \quad (2.18)$$

For considering the temporal correlation in a multimodal system, we may combine a set of N observation vectors into a matrix $Y = [y_1 \cdots y_N]$, the regressors also into a matrix $X = [x_1 \cdots x_N]$, and the noise terms into $E = [e_1 \cdots e_N]$ and formulated by

$$Y = AX + E \quad (2.19)$$

where the notation $A = [a_1 \cdots a_N]$ is referred to as the design matrix.

Assuming the noise covariance Σ is known, the conjugate prior on the design matrix A is the matrix-normal distribution [132]. A matrix $A \in \mathbb{R}^{d \times m}$ has a matrix-normal distribution $\mathcal{MN}(A; M, V, K)$ that given by

$$p(A) = \frac{|K|^{\frac{d}{2}}}{|2\pi V|^{\frac{m}{2}}} e^{\frac{1}{2}\text{tr}((A-M)^T V^{-1} (A-M)^K)} \quad (2.20)$$

where, the M is the mean matrix, and V and K^{-1} are related to the covariance along the rows and the columns of A .

Along with Eqs. 2.15 and 2.20, the conjugate prior on the set of parameters A and Σ is the matrix-normal inverse-Wishart prior that places a conditionally matrix-normal prior on matrix A given Σ , that is

$$\begin{cases} \Sigma \sim \mathcal{IW}(v, \Delta) \\ A|\Sigma \sim \mathcal{MN}(A; M, \Sigma, K) \end{cases} \quad (2.21)$$

2.4 Hidden Markov Model

HMM has been a workhorse in pattern recognition able to encode probabilistic state-space model. The HMM is a stochastic process where a finite number of latent states have Markovian state transitions. Conditioned on the mode sequence, the model assumes (discrete or continuous) conditionally independent observations given the latent state. Let z_t denote the latent state of the Markov chain at time $t \in T$, and π_j the state-specific transition distribution for state $j \in K$. Given the state z_t , the observation y_t is conditionally independent of observations and states at other time steps. Then, the Markovian structure on the state sequence and the observation are simply described as:

$$p(z_t | z_{t-1}) = \pi_{z_{t-1}}, \quad p(y_t | z_t) = F(\theta_{z_t}) \quad t > 1. \quad (2.22)$$

where, the state at the first time step is distributed according to an initial transition distributions π_0 ; $F(\cdot)$ represents a family of distributions (e.g., the multinomial for discrete data, or the multivariate Gaussian for real-vector-valued data); θ_{z_t} are the state-specific emission parameters and the $z_{1:T}$ is a state path over the hidden state space. Therefore, the resulting joint density for T observations is given by:

$$p(z_{1:T}, y_{1:T}) = \pi_0(z_1) p(y_1 | z_1) \prod_{t=2}^T p(z_t | z_{t-1}) p(y_t | z_t). \quad (2.23)$$

Then, to calculate the log-likelihood value of an observation vector $y_{1:T} = (y_1, y_2, y_3, \dots, y_T)$, we first write the joint distribution with the Markov property $p(y_t | y_{t-1}, \dots, y_1) = p(y_t | y_{t-1})$ and derive the joint probability of observation $y_{1:T}$ as:

$$\begin{aligned} p(y_{1:T}) &= \sum_{t=1}^T p(y_t | y_{1:t-1}) \\ p(y_t | y_{1:t-1}) &= \sum_{k=1}^K p(z_t = k | y_{1:t-1}) p(y_t | z_t) \end{aligned} \quad (2.24)$$

As illustrated in Eq. 2.23, if the state path $z_{1:T}$ is estimated, the maximum probability of the observation sequence can be obtained. However, the true state-path is hidden from the observations. There are different approaches to compute it: (i) the maximum likelihood state at any given moment; however, this approach does not consider uncertainty; (ii) a marginal probabilistic representation over hidden states at each time step. Here, the log-probability at time t is derived by computing the natural logarithm of the sum of exponentials over all hidden states:

$$\log p(y_{1:T}) = \log \sum_{k=1}^K \exp \left(\frac{\alpha_t(k) \cdot \beta_t(k)}{p(y_{1:T})} \right) \quad (2.25)$$

where $\alpha_t(k) = p(y_{1:t}, z_t)$, $\beta_t(k) = p(y_{t+1:T} | z_t)$, represent the forward message passing and backward message respectively in the standard forward-backward algorithm [37, 47].

$$\begin{aligned} p(z_t = k | y_{1:t-1}) &= \sum_j p(z_t | z_{t-1} = j) p(z_{t-1} | y_{1:t-1}) \\ \alpha_t(k) &= p(z_t = k | y_{0:t}) \\ &= \frac{1}{p(y_t | y_{1:t-1})} p(y_t | z_t = k) p(z_t = k | y_{1:t-1}). \end{aligned} \quad (2.26)$$

The denominator represents the probability of a sequence of observations that can be calculated by Eq. 2.24.

2.4.1 Forward-Backward Algorithm

To learn unknown parameters, the forward-backward algorithm [48] provides an efficient message passing procedure on computing node marginals of interest problems including *filtering* $p(z_n | y_1, \dots, y_n)$, *prediction* $p(z_{n+m} | y_1, \dots, y_n)$, and *smoothing* $p(z_n | y_1, \dots, y_N)$, $N > n$, where, z_n denotes the hidden state and y_n represents the observation. According to the implementation of the belief propagation algorithm in Bayesian graphical model, we define a set of forward and backward messages by

$$\begin{aligned} \text{forward : } \alpha_n(z_n) &\triangleq p(y_1, \dots, y_n, z_n) \\ \text{backward : } \beta_n(z_n) &\triangleq p(y_{n+1}, \dots, y_N | z_n) \end{aligned} \quad (2.27)$$

To address the problem of filtering, we formulate the filtering with the forward messages defined in Eq. 2.27. The filtering is simply generated as

$$\text{filtering : } p(z_n | y_1, \dots, y_n) = \frac{\alpha_n(z_n)}{\sum_z \alpha_n(z)} \quad (2.28)$$

To address the problem of prediction, we can utilize the Markov structure for the underlying chain to derive, that is

$$p(z_{n+m} | y_1, \dots, y_n) = \frac{\sum_{z_{n+m-1}} p(z_{n+m} | z_{n+m-1}) \cdots \sum_{z_n} p(z_{n+1} | z_n) \alpha_n(z_n)}{\sum_z \alpha_n(z)} \quad (2.29)$$

From Eq. 2.29, it shows that the prediction is equivalent to propagating the forward message without incorporating the missing observations y_{n+1}, \dots, y_{n+m} . However, we utilize both of the forward and backward messages for addressing the problem of smoothing for any m by

$$\begin{aligned}
p(z_n | y_1, \dots, y_N) &= \frac{p(y_1, \dots, y_N | z_n) p(z_n)}{p(y_1, \dots, y_N)} \\
&= \frac{p(y_1, \dots, y_n | z_n) p(y_{n+1}, \dots, y_N | z_n) p(z_n)}{p(y_1, \dots, y_N)} \\
&= \frac{\alpha_n(z_n) \beta_n(z_n)}{\sum_z \alpha_m(z) \beta_m(z)}
\end{aligned} \tag{2.30}$$

Hence, we can referred to the belief propagation algorithm for addressing the problems of filtering, predication, and smoothing of HMM. Among of them, we can use the forward messages for calculating the marginal likelihood of observations.

2.4.2 Viterbi Algorithm

The Viterbi algorithm is proposed for addressing the problem of the most likely state sequence to have generated an observation sequence y_1, \dots, y_N when given a set of HMM parameters, which provides an efficient dynamic programming approach to computing this Maximum A Posteriori (MAP) hidden state sequence by

$$\begin{aligned}
\hat{z} &= \max_{z_1, \dots, z_N} \pi^0(z_1) p(y_1 | z_1) \prod_{n=2}^N p(z_n | z_{n-1}) p(y_n | z_n) \\
&= \min_{z_1, \dots, z_N} \left[-\log \pi^0(z_1) - \log p(y_1 | z_1) + \sum_{n=1}^N -\log p(z_n | z_{n-1}) - \log p(y_n | z_n) \right]
\end{aligned} \tag{2.31}$$

From the above, we can note that choosing the MAP sequence is not necessarily equivalent to choosing the maximum marginal independently at each node by $\hat{z}_n = \max p(z_n | y_1, \dots, y_N)$. The marginal sequence also may not even be a feasible sequence for the HMM.

The Viterbi algorithm can be used to compute the most probable sequence of states in HMM based on the dynamic programming principle that the minimum cost path to $z_n = k$ is equivalent to the minimum cost path to node z_{n-1} plus the cost of a transition from z_{n-1} to $z_n = k$. Therefore, the MAP hidden Markov model state sequence $\hat{z}_1, \dots, \hat{z}_N$ can be computed in the following four steps that originally presented in literature [45]. Here, we first initialize the minimum path sum to state $z_1 = k$ for each $k \in \{1, \dots, K\}$ by

$$n = 1 : \quad \mathcal{S}_1(z_1 = k) = \log \pi^0(z_1 = k) - \log p(y_1 | z_k = k) \tag{2.32}$$

For $n = 2, \dots, N$, and for each $k \in \{1, \dots, K\}$, calculate the minimum path sum th state $z_n = k$ by

$$n = \{2, \dots, N\} : \quad \mathcal{S}_n(z_n = k) = -\log p(y_n | z_n = k) + \min_{z_{n-1}} \{\mathcal{S}_{n-1}(z_{n-1}) \\ - \log p(z_n = k | z_{n-1})\} \quad (2.33)$$

and simplifies to

$$z_{n-1}^*(z_n) = \arg \min_{z_{n-1}} \{\mathcal{S}_{n-1}(z_{n-1}) - \log p(z_n = k | z_{n-1})\} \quad (2.34)$$

thus, computing the probability of hidden state sequence given observation sequence by

$$\min_{z_1, \dots, z_N} -\log p(z_1, \dots, z_N | y_1, \dots, y_N) = \mathcal{S}_N(z_N) \quad (2.35)$$

and then set the notation $\hat{z}_N = \arg \min_{z_N} \mathcal{S}_N(z_N)$. Finally, the most likely hidden state sequence given observations can be iteratively calculated by

$$\hat{z}_n = z_n^*(\hat{z}_{n+1}), \quad n \in \{N-1, \dots, 1\} \quad (2.36)$$

We will propose methods that have straightforward connections with the Viterbi algorithm for anomaly monitoring during robot manipulation in Chap. 4.

2.5 Nonparametric Bayesian Hidden Markov Model

Bayesian methods are widely applied to various fields such as biomedicine, clinical trials, social science, and economics. The collected data are more and more complicated with the rapid development of modern science and technology. These complicated data include high/ultrahigh dimensional data, big data, discrete and continuous data. The aim of this section is to introduce the newly developed Bayesian methods, including Bayesian variable selection (e.g., fixed-dimensional data analysis and high/ultrahigh dimensional data analysis), Bayesian influence analysis (e.g., case deletion method and local influence analysis), Bayesian estimation and clustering methods (e.g., fixed dimensional data, high/ultrahigh dimensional data analysis, Bayesian network, and Bayesian clustering for big data), Bayesian hypothesis test including discrete and continuous random variables, variational Bayesian analysis and Bayesian clinical trials including design and dose-finding algorithm, and their applications in various fields.

Moreover, Bayesian nonparametric methods avoid the often restrictive assumptions of parametric models by defining distributions on function spaces, especially, for the prior function in Bayesian formulation. If effectively designed, these methods allow for data-driven posterior inference. The overview of Bayesian nonparametric inference in recent decades, see [49–51], and we briefly describe two aspects of Bayesian nonparametric methods on the hidden Markov model, including the

Dirichlet process and its hierarchical extension, such that formulated the Hierarchical Dirichlet Process Hidden Markov Model (HDP-HMM) for modeling multimodal observations in robotics.

2.5.1 Bayesian Nonparametric Priors

HMMs restrictively assumes a fixed model complexity. In process monitoring, latent states can represent robot primitives. It's clear that not all robot skills have the same number of primitives and that even the same skill might have variation when conducted under different conditions. To introduce flexibility into the number of computed hidden states, we leverage priors as probability measures.

HMMs can also be represented though a set of transition probability measures G_j . Probability measures yield strictly positive probabilities and sum to 1. Consider if instead of using a transition distribution on latent states, we use it across emission parameters $\theta_j \in \Theta$. Then,

$$G_j = \sum_{k=1}^K \pi_{jk} \delta_{\theta_k}, \quad (2.37)$$

where δ_{θ_k} is the unit mass for mode k at θ . The emission parameter θ_j , can be evaluated at time index $t - 1$, such that:

$$p(\theta'_t | \theta'_{t-1}) = G_{j_{t-1}}, \quad p(y_t | \theta'_t) = F(\theta'_t) \quad (2.38)$$

So, given θ_j , different probability weights are assigned to possible successor candidates θ_k . We can also assign a prior to the categorical probability measure G_j . The Dirichlet distribution is a natural selection due to conjugacy. Thus, the transition probabilities $\pi_j = [\pi_{j1} \cdots \pi_{jK}]$ are independent draws from a K -dimensional Dirichlet distribution:

$$\begin{aligned} p(\pi_j) &= Dir(\alpha_1, \dots, \alpha_K), \quad j = 1, \dots, K \\ \sum_k^K \pi_{jk} &= 1 \end{aligned} \quad (2.39)$$

Additionally, emission parameters are drawn from a base measure H such that, $p(\theta_j) = H$. The Dirichlet process (DP) was used as the base measured instead of the Dirichlet distribution. The DP is a distribution over countably infinite probability measures $G_0 : \Theta \Rightarrow R^+$, where $G_0(\theta) \geq 0$ and $\int_{\Theta} G_0(\theta) d\theta = 1$. The DP has a joint Dirichlet distribution:

$$p(G_0(\theta_1), \dots, G_0(\theta_K)) = Dir(\gamma H(\theta_1), \dots, \gamma H(\theta_K)). \quad (2.40)$$

We can further summarize the probability measure as $p(G_0) = DP(\gamma, H)$ as:

$$G_0 = \sum_{k=1}^{\infty} \beta_k \delta_{\theta_k} \quad p(\theta_k) = H, \quad (2.41)$$

where γ is the concentration parameter and H is the base measure over parameter space Θ . The weights β_k are sampled via a stick-breaking construction:

$$\beta_k = v_k \prod_{l=1}^{k-1} (1 - v_l), \quad (2.42)$$

where $p(v_k) = \beta(1, \gamma)$. For succinctness, the stick-breaking process is defined as: $p(\beta) = GEM(\gamma)$. The DP is used to define a prior on the set of HMM transition probability measures G_j . However, if each transition measure G_j is an independent draw from $DP(\gamma, H)$, where H is continuous, like a Gaussian distribution, transition measures lead to non-overlapping support. This means that previously seen modes (robotic primitives) cannot be selected again. To deal with this limitation, a Hierarchical Dirichlet Process (HDP) is used. The latter constructs transition measures G_j on the same support points $(\theta_1, \theta_2, \dots, \theta_K)$. This can be done when G_j is only a variation on a global discrete measure G_0 , such that:

$$\begin{aligned} G_0 &= \sum_{k=1}^{\infty} \beta_k \delta_{\theta_k}, \quad p(\beta | \gamma) = GEM(\gamma) \\ p(\theta_k | G_0) &= G_0 \\ G_j &= \sum_{k=1}^{\infty} \pi_{jk} \delta_{\theta_k}, \quad p(\pi_j | \alpha, \beta) = DP(\alpha, \beta). \end{aligned} \quad (2.43)$$

This HDP is used as a prior on the HMM. The implications are a mode complexity learned from the data and a sparse state representation. The $DP(\alpha, \beta)$ distribution encourages modes with similar transition distributions, but does not distinguish between self- and cross-mode transitions. This is problematic for dynamical data. The HDP-HMM yields large posterior probabilities for mode sequences with unrealistically fast dynamics. Emily B. Fox et al. [52] introduced the sticky parameter into the HDP, yielding the sHDP-HMM, thus the transition probability π_j is defined as:

$$p(\pi_j | \alpha, \kappa, \beta) = DP \left(\alpha + \kappa, \frac{\alpha \beta + \kappa \delta_i}{\alpha + \kappa} \right) \quad (2.44)$$

The sticky HDP increases the expected probability of self-transitions by an amount proportional to κ and leads to posterior distributions with smoothly varying dynamics. Finally, priors are placed on the concentration parameters α and γ , and the sticky parameter κ . Latent state creation is influenced by α and γ , while self-transition probabilities are biased by κ . These priors allow to integrate expert knowledge for better modeling than the expectation-maximization (EM) algorithm traditionally used in HMMs.

2.5.2 The sHDP-VAR-HMM

Many complex dynamical phenomena are not adequately described as conditionally independent of the observations given the state z_t . That is, the observations are a noisy linear combination of some finite set of past observations plus additive white noise. In such cases, the dynamical processes can be modeled by an autoregressive (AR) process. Particularly, an order r vector AR process, denoted by VAR(r), with observations $y_t \in \mathbb{R}^d$. The VAR(r) system can be considered an extension of the HMM; instead of having conditionally independent observations given the hidden state sequence, the system has conditionally linear dynamics. Figure 4.9 shows a graphical model representation and compares it to the sHDP-HMM described in previous section.

The VAR has simplifying assumptions that make it a practical choice in applications [52]. The switching regime can be combined with the sHDP-HMM from Sect. 2.5.1 to leverage the expressiveness of the VAR system with the ability of non-parametric priors to learn the mode complexity of the model. The VAR(r) process, with autoregressive order r consists of a latent state $z_t \in R^n$ with linear dynamics observed through $y_t \in R^d$. The observations have mode specific coefficients and process noise as:

$$\begin{aligned} p(z_t) &= \pi_{z_{t-1}}, \quad t > 1 \\ y_t &= \sum_{i=1}^r A_i^{z_t} y_{t-i} + e_t(z_t), \quad p(e_t) = \mathcal{N}(0, \Sigma). \end{aligned} \tag{2.45}$$

We see a generative model for a time-series $\{y_1, y_2, \dots, y_T\}$ of observed multimodal data, a matrix of regression coefficients $\mathbf{A}^{(k)} = [A_1^{(k)} \dots A_r^{(k)}] \in \mathbb{R}^{dx(d*r)}$, and a measurement noise Σ , with a symmetric positive-definite covariance matrix. Given the observation data, we are interested in learning the “ r^{th} ” model order, for which we need to infer $\{\mathbf{A}^{(k)}, \Sigma^{(k)}\}$. We leverage the Bayesian approach through the placement of conjugate priors on both parameters for posterior inference. As the mean and covariance are uncertain, the Matrix-Normal Inverse-Wishart (MNIW) serves as an appropriate prior on the multivariate AR distribution. The MNIW places a conditionally matrix-normal prior on $\mathbf{A}^{(k)}$ given Σ :

$$p(\mathbf{A}^{(k)} | \Sigma) = \mathcal{MN}(\mathbf{A}; M, \Sigma, K), \tag{2.46}$$

and an inverse-Wishart prior on Σ :

$$p(\Sigma) = IW(v_0, S_0). \tag{2.47}$$

See [45] Appendix F.1 for a detailed derivation of resulting posterior and definitions for variables M and K which are parameters that define the matrix-normal portion of the MNIW prior.

2.6 Summary

In this chapter, a Bayesian nonparametric approach in defining a prior on the hidden Markov model that allows for flexibility in addressing the problem of modeling the complex dynamics during robot manipulation task is proposed. Considering the underlying dynamics that can be well-modeled as a hidden discrete Markov process, but in which there is uncertainty about the cardinality of the state space. We use the hierarchical Dirichlet process (HDP) for examining an HMM with an unbounded number of possible states. Subsequently, the sticky HDP-HMM is investigated for allowing more robust learning of the complex dynamics through a learned bias by increasing the probability of self-transitions. Although the HDP-HMM and its sticky extension are very flexible time series models, they make a strong Markovian assumption that observations are conditionally independent given the discrete HMM state. This assumption is often insufficient for capturing the temporal dependencies of the observations in real data. Consequently, an extension of the sticky HDP-HMM for learning the switching dynamical processes with switching linear dynamical system is investigated. As the basis, the detailed formulations and theoretical process of those models are explained step-by-step in each sections, and widely used in the later chapters of this book.

References

1. Alho K, Kujala T, Paavilainen P, et al. Auditory processing in visual brain areas of the early blind: evidence from event-related potentials. *Electroencephalogr Clin Neurophysiol*. 1993;86(6):418–427.
2. 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.
3. 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.
4. Fitzpatrick P, Arsenio A, Torres-Jara ER. Reinforcing robot perception of multi-modal events through repetition and redundancy and repetition and redundancy. *Interact Stud*. 2006;7(2):171–196.
5. Huadong W, Mel S. Correlation of accelerometer and microphone data in the “coin tap test”. *IEEE Trans Instrum Meas*. 2000;49(3):493–7.
6. Su Z, Kroemer O, Loeb GE, et al. Learning to switch between sensorimotor primitives using multimodal haptic signals. In: International conference on simulation of adaptive behavior. Springer; 2016. p. 170–182.
7. Kirchner EA, Fairclough SH, Kirchner F. Embedded multimodal interfaces in robotics: applications, future trends, and societal implications. *The handbook of multimodal-multisensor interfaces: language processing, software, commercialization, and emerging directions-volume*. 2019;1(3):523–76.
8. 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>.
9. Kappler D, Pastor P, Kalakrishnan M, et al. Data-driven online decision making for autonomous manipulation. In: *Robotics: science and systems*; 2015.

10. Pieskä S, Kaarela J, Mäkelä J. Simulation and programming experiences of collaborative robots for small-scale manufacturing. In: The 2nd international symposium on small-scale intelligent manufacturing systems (SIMS); 2018. p. 1–4.
11. Li Y, Yue Y, Xu D, et al. Folding deformable objects using predictive simulation and trajectory optimization. In: IEEE/RSJ international conference on intelligent robots and systems (IROS); 2015. p. 6000–6006.
12. Brockwell PJ, Davis RA, Calder MV. Introduction to time series and forecasting. Berlin: Springer; 2002.
13. Eddy SR. Hidden markov models. *Curr Opin Struct Biol.* 1996;6(3):361–5.
14. Huang XD, Ariki Y, Jack MA. Hidden Markov models for speech recognition; 1990.
15. Rabiner LR. A tutorial on hidden Markov models and selected applications in speech recognition. *Proc IEEE.* 2009;77(2):257–286.
16. Gales M, Young S. The application of hidden Markov models in speech recognition. *Found Trends Signal Process.* 2008;1(3):195–304.
17. Krogh A, Brown M, Mian IS, et al. Hidden Markov models in computational biology: applications to protein modeling. *J Mol Biol.* 1994;235(5):1501–1531.
18. Krogh A, Larsson B, Von Heijne G, et al. Predicting transmembrane protein topology with a hidden Markov model: application to complete genomes. *J Mol Biol.* 2001;305(3):567–80.
19. Och FJ, Ney H. The alignment template approach to statistical machine translation. *Comput Linguist.* 2004;30(4):417–449.
20. Och FJ, Ney H. A comparison of alignment models for statistical machine translation. In: Proceedings of the 18th conference on computational linguistics-volume 2. Association for Computational Linguistics; 2000. p. 1086–1090.
21. Karlof C, Wagner D. Hidden Markov model cryptanalysis. In: International workshop on cryptographic hardware and embedded systems. Springer; 2003. p. 17–34.
22. Green PJ, Noad R, Smart NP. Further hidden Markov model cryptanalysis. In: International workshop on cryptographic hardware and embedded systems. Springer, 2005. p. 61–74.
23. Bhar R, Hamori S. Hidden Markov models: applications to financial economics. Springer Science & Business Media; 2004.
24. Mamon RS, Elliott RJ. Hidden Markov models in finance. Berlin: Springer; 2007.
25. Anne V, Elliot D, Eddie J, et al. Sensor tracked points and HMM based classifier for human action recognition. *Comput Sci Emerg Res J.* 2016;5:4–8.
26. Ren L, Patrick A, Efros AA, et al. A data-driven approach to quantifying natural human motion. *ACM Trans Graph (TOG) ACM.* 2005;24:1090–1097.
27. Hovland GE, McCarragher BJ. Hidden Markov models as a process monitor in robotic assembly. *Int J Robot Res.* 1998;17(2):153–168.
28. Hussam A, George R. Process monitoring using hidden Markov models. *Qual Reliab Eng Int.* 2014;30(8):1379–87.
29. Fox M, Ghallab M, Infantes G, et al. Robot introspection through learned hidden markov models. *Artif Intell.* 2006;170(2):59–113.
30. Wu H, Lin H, Guan Y, et al. Robot introspection with Bayesian nonparametric vector autoregressive hidden Markov models. In: 2017 IEEE-RAS 17th international conference on humanoid robotics (Humanoids). IEEE; 2017. p. 882–888
31. Chuk TY, Chan AB, Shimojo S, et al. Mind reading: discovering individual preferences from eye movements using switching hidden Markov models. In: Proceedings of the 38th annual conference of the cognitive science society, CogSci; 2016.
32. Manogaran G, Vijayakumar V, Varatharajan R, et al. Machine learning based big data processing framework for cancer diagnosis using hidden Markov model and GM clustering. *Wirel Pers Commun.* 2018;102(3):2099–2116.
33. Kroemer O, Van Hoof H, Neumann G, et al. Learning to predict phases of manipulation tasks as hidden states. In: 2014 IEEE international conference on robotics and automation (ICRA). IEEE; 2014. p. 4009–4014.
34. Kroemer O, Daniel C, Neumann G, et al. Towards learning hierarchical skills for multi-phase manipulation tasks. In: 2015 IEEE international conference on robotics and automation (ICRA). IEEE; 2015. p. 1503–1510.

35. Eddy SR. What is a hidden Markov model?. *Nat Biotechnol.* 2004;22(10):1315.
36. McLachlan G, Krishnan T. The EM algorithm and extensions. Hoboken: Wiley; 2007.
37. Leroux BG. Maximum-likelihood estimation for hidden Markov models. *Stoch Process Their Appl.* 1992;40(1):127–143.
38. Hjort NL, Holmes C, Müller P, et al. Bayesian nonparametrics. Cambridge: Cambridge University Press; 2010.
39. Gael JV, Teh YW, Ghahramani Z. The infinite factorial hidden Markov model. *Adv Neural Inf Process Syst.* 2009:1697–1704
40. Teh YW, Jordan MI. Hierarchical Bayesian nonparametric models with applications. *Bayesian Nonparametrics.* 2010;1:158–207.
41. Carlo CM. Markov chain monte carlo and gibbs sampling. Lecture notes for EEB; 2004. p. 581.
42. Brooks S, Gelman A, Jones G, et al. Handbook of markov chain monte carlo. Boca Raton: CRC Press; 2011.
43. Fox EB, Sudderth EB, Jordan MI, et al. A sticky HDP-HMM with application to speaker diarization. *Ann Appl Stat.* 2011;1020–1056.
44. Fox EB, Sudderth EB, Jordan MI, et al. The sticky HDP-HMM: bayesian nonparametric hidden Markov models with persistent states; 2007.
45. Fox EB. Bayesian nonparametric learning of complex dynamical phenomena. Massachusetts Institute of Technology; 2009.
46. Fox E, Sudderth EB, Jordan MI, et al. Nonparametric Bayesian learning of switching linear dynamical systems. *Adv Neural Inf Process Syst.* 2009:457–464.
47. Yu SZ, Kobayashi H. An efficient forward-backward algorithm for an explicit-duration hidden Markov model. *IEEE Signal Process Lett.* 2003;10(1):11–4.
48. Rabiner LR. A tutorial on hidden Markov models and selected applications in speech recognition. *Proc IEEE.* 1989;77.
49. Müller Peter, Quintana FA. Nonparametric bayesian data analysis. *Stat Sci.* 19(1):95–110.
50. Sudderth EB. Graphical models for visual object recognition and tracking. Ph.D. thesis, MIT, Cambridge; 2006.
51. Walker SG, Damien PL, Purushottam W, Smith AFM. Bayesian non parametric inference for random distributions and related functions. *J R Stat Soc, Ser B.* 1999;61(3):485–527.
52. Fox EB, Sudderth EB, Jordan MI, Willsky AS. Bayesian nonparametric methods for learning Markov switching processes. *IEEE Signal Process Mag.* 2010;1827(6):43–54.

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.



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 contact 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 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: $M = (Q, q_0, \Sigma, \delta, F)$, where Q is a non-empty finite set of states; $q_i \in Q$ represents a state; q_0 is the initial state; Σ is the input condition Instruction; δ 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

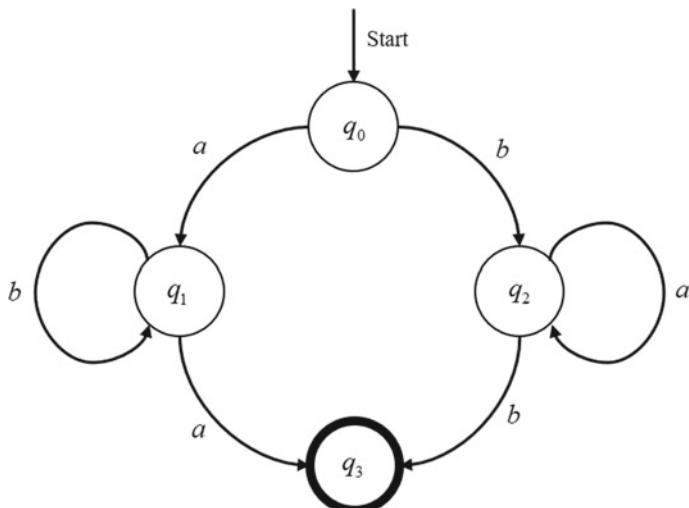


Fig. 3.1 Illustrates the graphical representation of FSM state transition

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

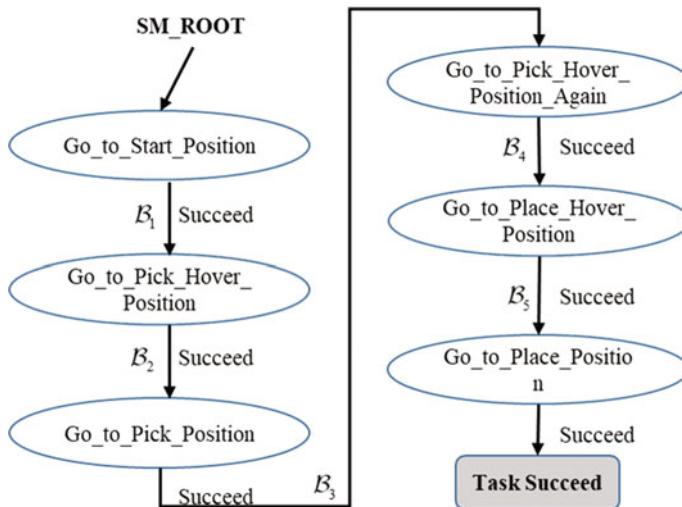


Fig. 3.2 Illustrates the FSM implementation of Baxter robot performing pick-and-place task

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 $z \in \{1, 2, \dots, K\}$ from the derived hidden state sequences $\mathcal{Z} = \{\mathcal{X}_1, \mathcal{X}_2, \dots, \mathcal{X}_N, \}$, a Finite State Machine (FSM) that represents the task can designed to be constructed by creating nodes¹ N_1, N_2, \dots, N_m , where $K \leq m$ and each element should correspond to the same hidden state z . For $k \in \{1, \dots, m\}$, 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.

¹Here, the term “node” is used to instead of the default “state” in FSM for avoiding confusing hidden state of HDP-VAR-HMM or state of the robot.

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

$$\begin{aligned}\tau \dot{v} &= K(g - x) - Dv - K(g - x_0)s + Kf(s), \\ \tau \dot{x} &= v.\end{aligned}\tag{3.1}$$

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 τ . The scaling term originates from an additional system that controls the system's phase execution by

$$\tau \dot{s} = -\alpha s,\tag{3.2}$$

where, the scalar α 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-sot 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,

$$\begin{aligned}\tau f(s) &= \frac{\sum_{i=1}^N w_i \psi_i(s)s}{\sum_i \psi_i(s)} \\ \psi(s) &= (-h_i(s - c_i)^2)\end{aligned}\tag{3.3}$$

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 τ variable in Eq. 3.1 is designed for allowing the system to speed up or slow

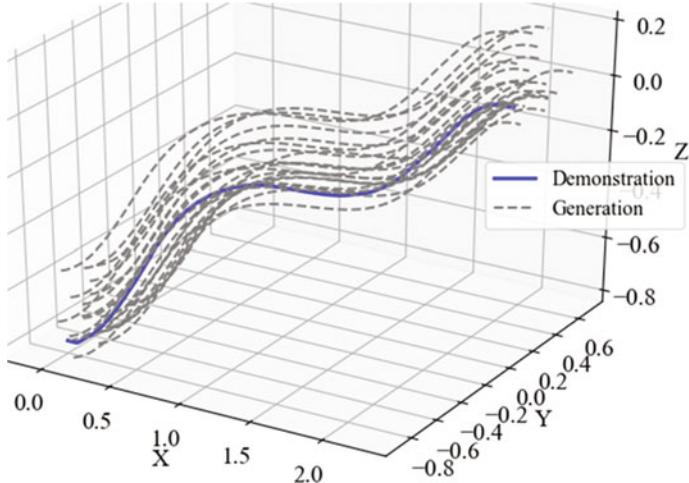


Fig. 3.3 Illustrates the learning and generalization of dynamical movement primitive

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,

$$f_{target}(s) = \frac{-K(g - x(s)) + D\dot{x}(s) + \tau\ddot{x}(s)}{g - x_0}. \quad (3.4)$$

Next, the goal is set to $g = x(T)$ and τ 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 $y_n = [y_1, y_2, \dots, y_T]$ 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 $z_t = k \in \{1, 2, 3, \dots\}$. A countably infinite set of consecutive integers z_t that should be satisfied with the requirements that an initial probabilistic state distributions π_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

$$\begin{aligned} z_t &\sim \pi_{z_{t-1}}, \\ p(z_1 = k) &= \pi_0, \\ p(z_t = k' | z_{t-1} = k) &= \pi_{kk'}. \end{aligned} \tag{3.5}$$

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 ϵ 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 ϵ 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 $\mathbb{E}[y_t | y_{t-1}] = A_k y_{t-1}$, and Σ_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 ,

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

We define the parameter space $\Theta = \{\theta_1, \theta_2, \dots, \theta_K\}$ 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

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

$$\mathcal{L}_t = \log \sum_{k=1}^K \exp \left(\frac{\alpha_t(k) \cdot \beta_t(k)}{p(y_{0:T})} \right), \quad (3.8)$$

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

$$\begin{aligned} \alpha_t(k) &= \frac{1}{p(y_t | y_{0:t-1})} p(y_t | z_t = k) p(z_t = k | y_{1:t-1}), \\ p(z_t = k | y_{0:t-1}) &= \sum_j p(z_t | z_{t-1} = j) p(z_{t-1} | y_{0:t-1}). \end{aligned} \quad (3.9)$$

The *expected cumulative likelihood* is $\mathbb{E} [\log P(y_{1:T} | \Theta_m)]$ for each trained model $m \in M$ using Eq. 3.8, where Θ_m represents the parameters of the trained model. That is, given a test trial ξ , the cumulative log-likelihood is computed for test trial observations conditioned on all available trained skill model parameters $\log P(y_{\xi_1:\xi_t} | \Theta)_m^M$ 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 ξ_c , we can index the correct log-likelihood $\mathbb{I}(\xi_c \in s)$ and see if the probability density of the test trial given the correct model is greater than the rest:

$$\begin{aligned} \log P(y_{\xi_1:\xi_c} | \Theta_s) &> \log P(y_{\xi_1:\xi_c} | \Theta_m) \\ \forall m (m \in M \wedge m \neq s). \end{aligned} \quad (3.10)$$

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 vector 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-

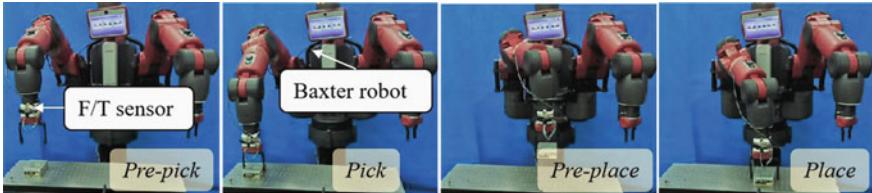


Fig. 3.4 Experiment 1: the skills of a traditional pick-and-place task by FSM: **i** Pre-pick, **ii** Pick, **iii** Pre-place, and **iv** Place

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 $M = \mathbf{0}$ and setting $K = 10 * I_m$. The inverse-Wishart portion of the prior is given by $\nu_0 = m + 2$ DoF (the smallest integer setting to maintain a proper prior). The scale matrix $S_0 = 0.75*$ 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 $\text{Gamma}(a, b)$ prior is set on the sHDP concentration parameters $\alpha + \kappa$ and γ . A $\text{Beta}(c, d)$ prior is set on the self-transition parameter ρ . 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

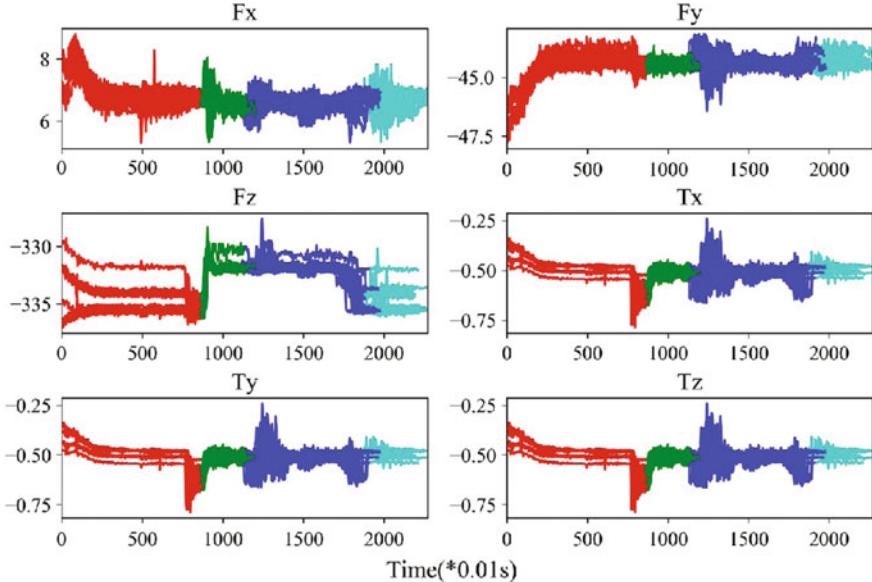


Fig. 3.5 Experiment 1: recorded wrench signals in the pick-and-place task. Different colors encode different skills

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 i th 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-

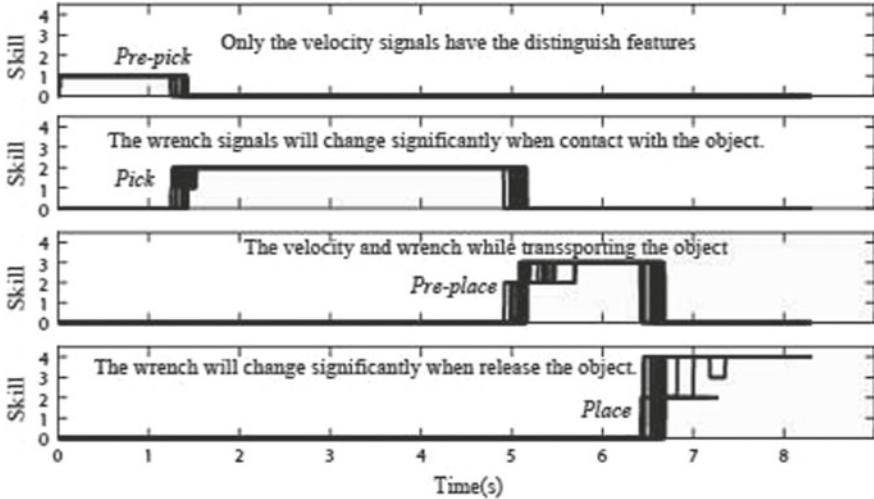


Fig. 3.6 Experiment 1: sHDP-VAR(2)-HMM model outputs for the pick-and-place task

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.

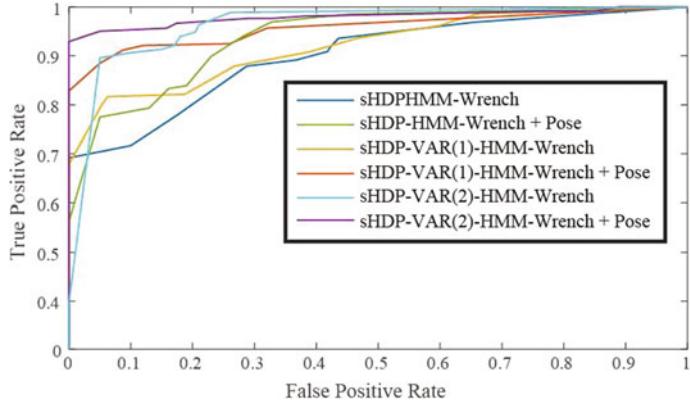


Fig. 3.7 Experiment 1: Receiver operating characteristic (ROC) curves for the pick-and-place task. The figure shows ROC curves that compare the performance of multimodal and unimodal sensing for anomaly monitoring and the performance of our anomaly monitoring method versus other baseline methods

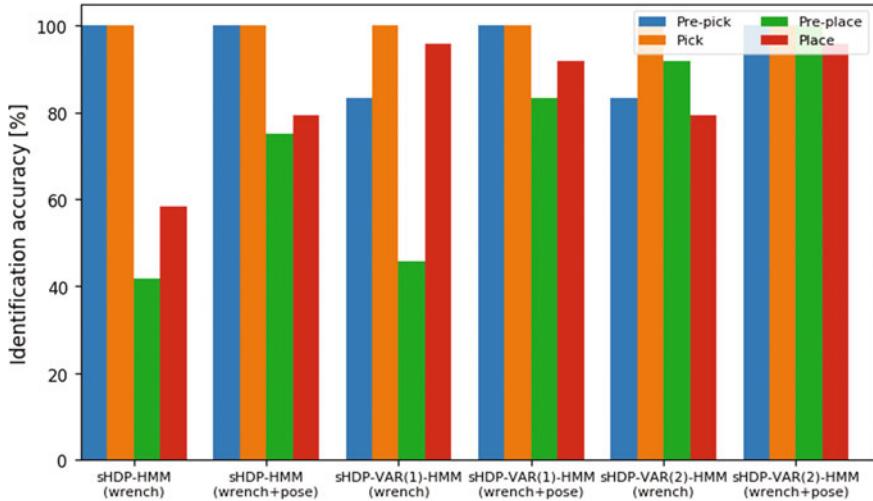


Fig. 3.8 The identification accuracy of different methods with different multimodal signals in a pick-and-place experiment

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

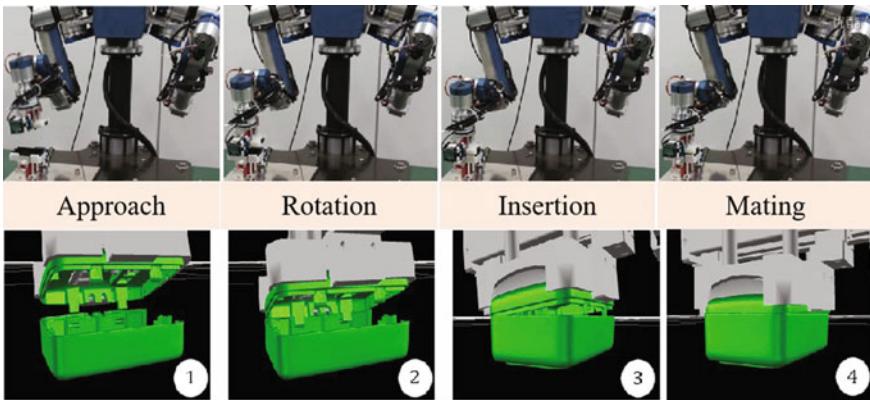


Fig. 3.9 Experiment 2: the skills in a snap assembly task FSM: **i** Approach; **ii** Rotation; **iii** Insertion; **iv** Mating

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.

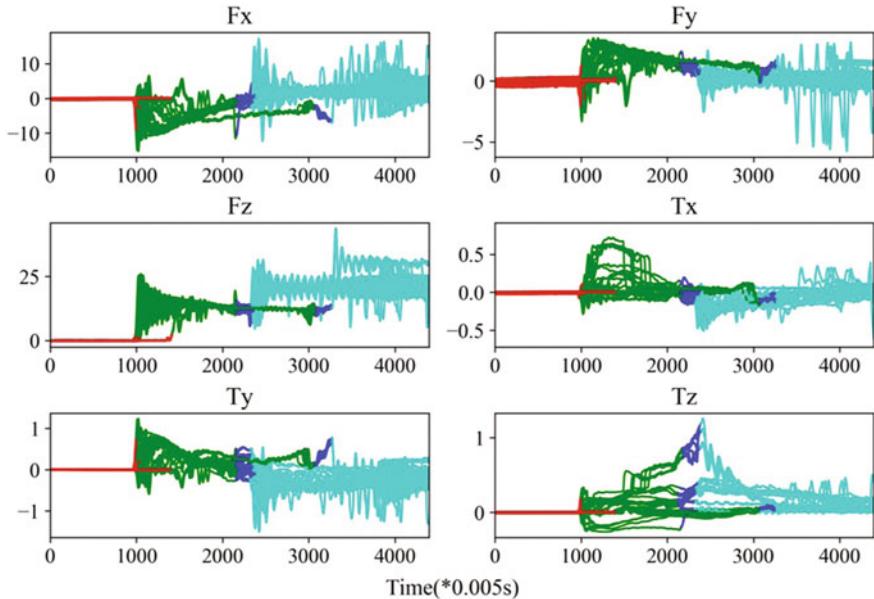


Fig. 3.10 Experiment 2: recorded wrench signals in the snap assembly task. Different colors encode different skill

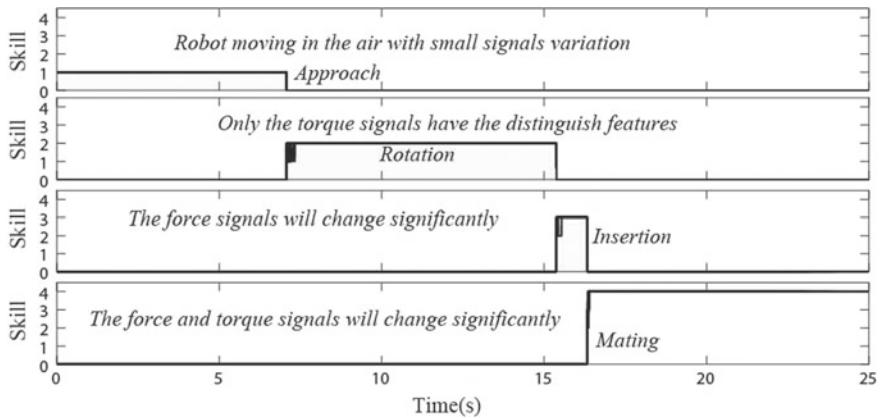


Fig. 3.11 Experiment 2: sHDP-VAR(2)-HMM model outputs in the snap assembly task

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

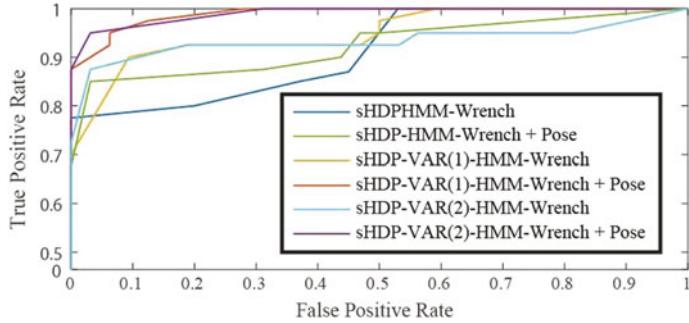


Fig. 3.12 Experiment 2: Receiver operating characteristic (ROC) curves for the snap assembly task across multimodal scenarios and baseline methods. Unimodal wrench signals are compared with multimodal pose and wrench signals

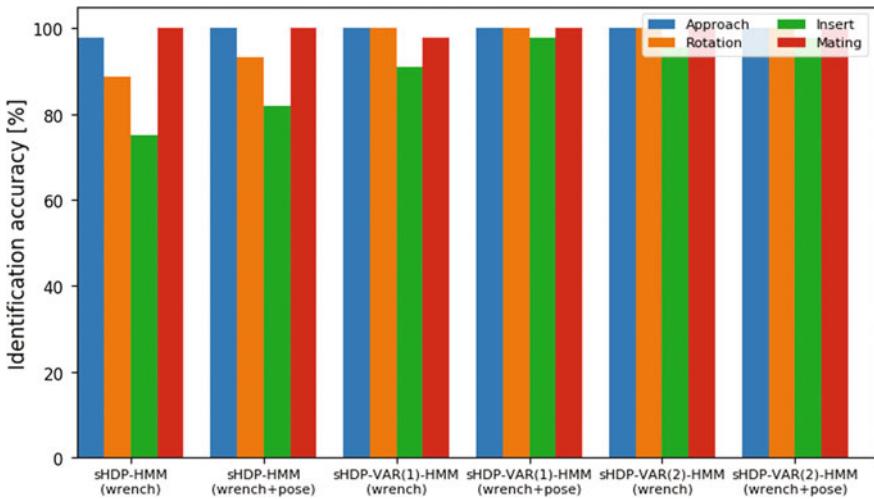


Fig. 3.13 The identification accuracy of different method with variable sensor signals in snap assembly experiment

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.

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

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.

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, Alakshendra 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.

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. Ijspeert 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 O. 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.



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-

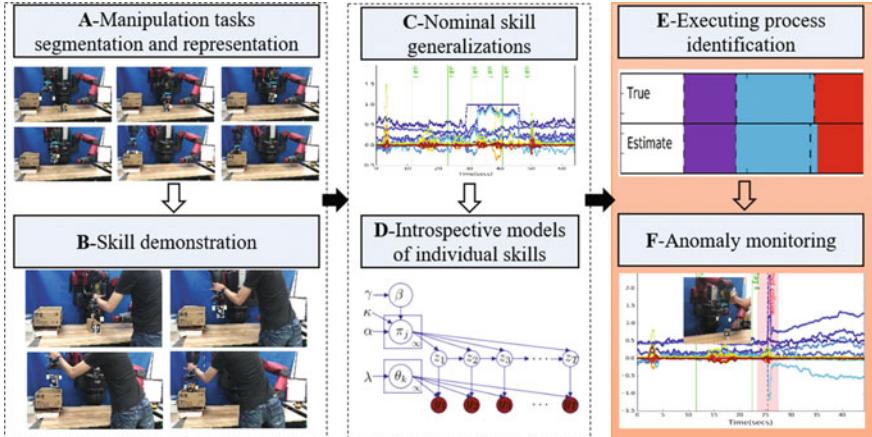


Fig. 4.1 Illustration of the anomaly monitoring procedure after skill identification

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 $Y = \{y_1, \dots, y_T\}$ 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 $t = \{x_1, \dots, x_T\}$ of observed multi-modal data τ consisting of Cartesian pose and wrench values. Then, a mode-dependent matrix of regression coefficients $\mathbf{A}^{(k)} = [A_1^{(k)} \dots A_r^{(k)}] \in \mathbb{R}^{d \times (d*r)}$ with an r th autoregressive order and d dimensions is used along with a measurement noise Σ , with a symmetric positive-definite covariance matrix. The generative model for the sHDP-AR-HMM is summarized as:

$$\begin{aligned} G_0 &= \sum_{k=1}^{\infty} \beta_k \delta_{\theta_k} \quad \beta | \gamma \sim GEM(\gamma). \\ \theta_k | G_0 &\sim G_0. \\ G_j &= \sum_{k=1}^{\infty} \pi_{jk} \delta_{\theta_k} \quad \pi_j | \alpha, \beta \sim DP(\alpha, \beta). \end{aligned} \tag{4.1}$$

The probability measure G_j , which models the transition distribution of the modes π_j determines the weights (probabilities) of transitioning between modes δ_{θ_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 $\theta_{z_t} = \mathcal{N}(\mu, \sigma^2)$. 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).

$$\text{Observation Dynamics: } y_t = \sum_{i=1}^r A_i^{(z_t)} y_{t-i} + e_t(z_t). \\ e_t \sim \mathcal{N}(0, \Sigma). \quad (4.2)$$

$$\text{Mode Dynamics: } z_t^{(i)} \sim \pi_{z_{t-1}}^{(i)}.$$

By using the model over a set of multi-modal exemplar data τ , 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 θ of a given skill, hence $\theta_{S_m} = \{\Pi, \mathbf{A}\}_{S_m}$.

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 τ_m . τ_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 Π of a given skill s . Hence $\Pi_s = \{\pi, \mathbf{A}\}$: 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 $\log P(y_{r_1:r_t}|\Pi)_s^S$ 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 $\mathbb{I}(\Pi_s = s_c)$ and see if its probability density of the test trial given the correct model is greater than the rest:

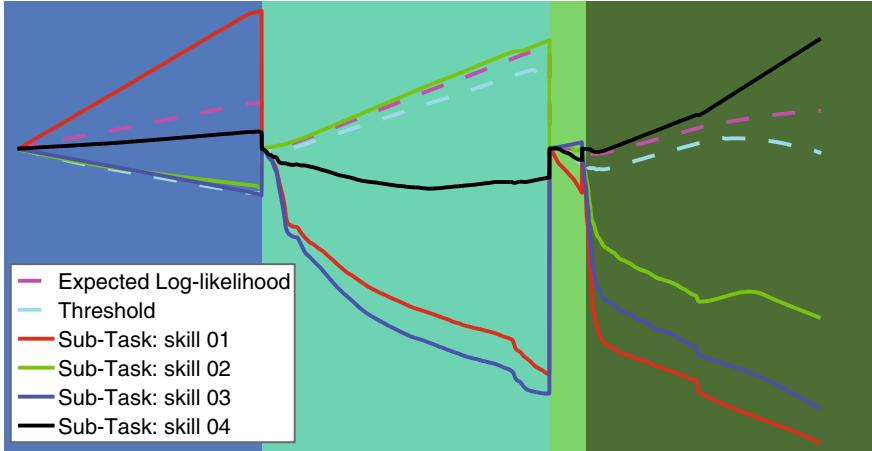


Fig. 4.2 Illustration of cumulative log-likelihood (L-) curves for four skills s in a task. As such, 4 probabilistic models Π_s are derived. Given an indexed position in graph N_i , one can generate four L-curves. The curve for the correct model, has a value increasing likelihood, while the other graphs 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

$$\log P(y_{r_1:r_t} | \Pi_{correct}) > \log P(y_{r_1:r_t} | \Pi_s) \forall s (s \in S \wedge s \neq s_c).$$

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 $F1$. 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

$$F1_{s_c} = \mu(L) - c * \sigma(L) \quad (4.3)$$

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 $\log P(y_{r_1:r_t} | \Pi_{correct}) < F1_{s_c}$: 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 $F1$. As the difference between L and $F1$ is minimal at first the new anomaly threshold $F2$ (for an indexed skill) is focused on computing the derivative of the difference:

$$F2_{s_c} = \frac{d | L - F1_{s_c} |}{dt} \quad (4.4)$$

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-

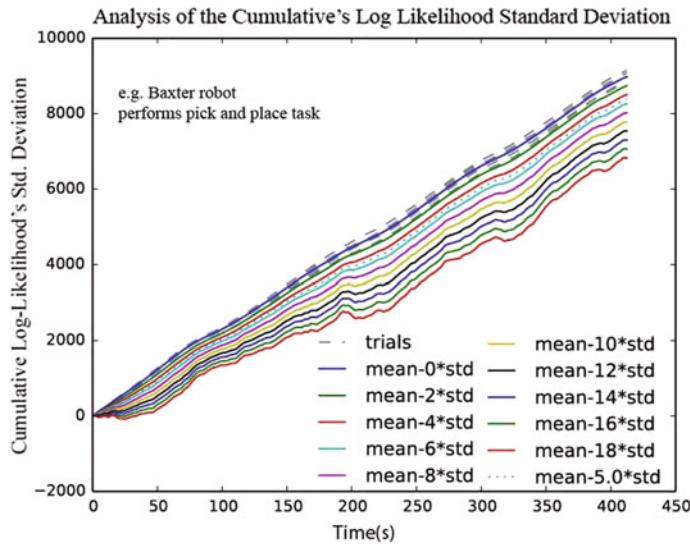


Fig. 4.3 This plot shows how the standard deviation of the cumulative log-likelihoods computed during training grows over time. The standard deviation grows as observations show greater variance due to the accumulation of error

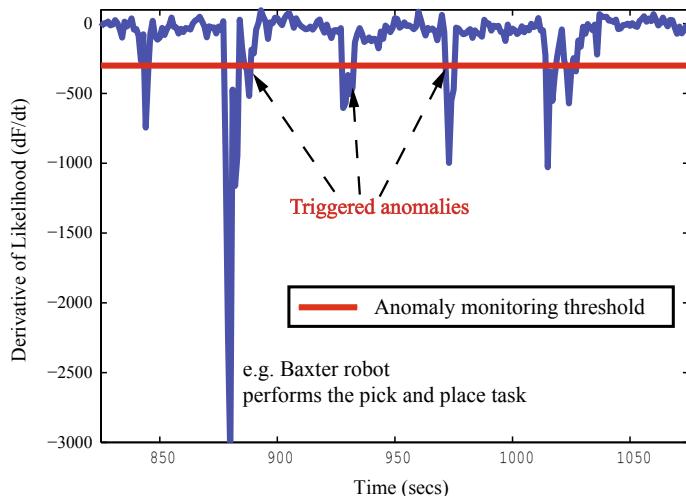


Fig. 4.4 Example of how the anomaly identification metric crosses the anomaly threshold as anomalies are triggered by external perturbations during a skill execution

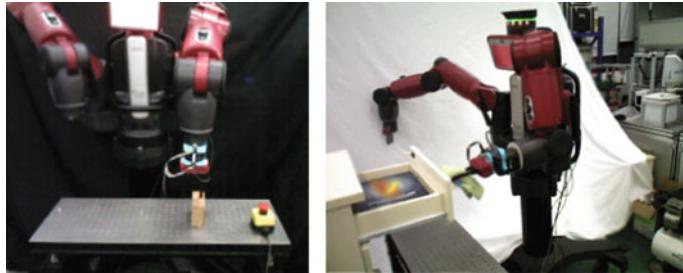


Fig. 4.5 Two examples (pick-and-place (UPPER) and open-and-close-drawer) in which a human collaborator accidentally collides the robot. The introspection system identifies an anomaly (see bottom left plots) and triggers a recovery behavior (see the fluorescent node in the graph on the right)

lisions for a pick-and-place task. For each of the three experiments, we test four separate conditions:

- (i) no anomalies
- (ii) anomalies without recovery
- (iii) one anomaly caused at each executed skill and
- (iv) multiple anomalies caused at each skill.

The pick-and-place task consists of 5 basic nodes (not counting the home node and the recovery node): pre-pick, pick, pre-pick (returns to an offset position), pre-place, and place. The open-and-close-drawer task consists of 5 nodes: pre-grip, grip, pull-to-open, push-to-close, and go-back-to-start. The direction and intensity of the human collisions was random but all executed under the sense that these are accidental contacts as a human user reaches into the workspace of the robot without noticing the robot's motion. We note that while the tasks are simple, the main focus of the work is placed on the robustness of the recovery systems given different external disturbance scenarios. A dual-armed humanoid robot -Baxter- was used. All code was run in ROS Indigo and Linux Ubuntu 14.04 on a mobile workstation with an Intel Xeon processor, 16GB RAM, and 8 cores.

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

Table 4.1 Results for accidental human and tool collisions. Recovery strategy success rates are presented by the F-score, and robustness of the robot introspection system for anomaly is shown in the recall and precision settings. We also compare the performance of the more expressive sHDP-HMM model with an HMM that serves as a baseline. Generally, the sHDP-HMM model allowed for better introspection and thus better recovery rates and better recall

	HMM					sHDP-HMM				
	Micro			Macro		Micro			Macro	
	F	Recall	Precision	Recall	Precision	F	Recall	Precision	Recall	Precision
Human collision										
<i>Pick & Place</i>										
(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
<i>Open drawer</i>										
(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										
<i>Pick & Place</i>	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 $\sim 82\%$ (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 $\sim 84.5\%$. 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 $\sim 95\%$ and a recall of $\sim 84.5\%$ (micro/macro). The drawer task recovered 93.3% of the time with a precision of $\sim 100\%$, and a recall of $\sim 93\%$ (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 $\sim 88.89\%$, 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 Θ . 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 Θ 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 $i \in \{1, 2, \dots, N\}$ can be represented according to Eq. (4.5). To compute \mathcal{L}_t , we first compute $\log \alpha_i(t)$. According to the forward-algorithm, we derive:

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

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

$$\log \alpha_i(t) = \log b_i(y_t) + \log \sum_{j=1}^N \exp(\log \alpha_j(t-1) + \log A_{ji}) \quad (4.7)$$

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 $y_{1:t}$:

$$\begin{aligned}
 \hat{z}_{1:t} &= \arg \max_{z_{1:t}} P(z_{1:t} \mid y_{1:t}) \\
 &= \arg \max_{z_t} (b_{z_t}(y_t) \\
 &\quad \arg \max_{z_{1:t-1}} (A_{z_{t-1} z_t} P(z_{1:t-1}, y_{1:t-1}))) \\
 &= \arg \max_{z_t} (b_{z_t}(y_t) \\
 &\quad \arg \max_{z_{t-1}} (A_{z_{t-1} z_t} b_{z_{t-1}}(y_{t-1}) \\
 &\quad \quad \ddots \\
 &\quad \quad \arg \max_{z_1} (A_{z_1 z_2} \pi_{z_1} b_{z_1}(y_1)) \dots))
 \end{aligned} \tag{4.8}$$

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 trace-back \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 achieved 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 tend to self-transition. Also at time $t-1$:

$$\delta_{t-1}(l) > \delta_{t-1}(k). \tag{4.9}$$

We also define the following symbol:

$$w_{ji}(t) = A_{ji} * b_i(y_t). \quad (4.10)$$

Due to our first assumption, we have: $\max_j w_{ji}(t) = w_{ii}(t)$. Then, at time t , the δ 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, ($\delta_{t-1}(l) > \delta_{t-1}(k)$ 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:

$$\delta_{t-1}(k) * w_{kk}(t) < \delta_{t-1}(l) * w_{lk}(t) \quad (4.15)$$

and

$$\delta_{t-1}(k) * w_{kk}(t) > \delta_{t-1}(l) * w_{ll}(t) \quad (4.16)$$

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 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 $\log \sum_{i=1}^N \exp(y_i) \approx \max_{i \in \{1, \dots, N\}} y_i$ 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:

$$\mathcal{L}_t \approx \max_{i \in \{1, \dots, N\}} (\log \alpha_i(t)) \quad (4.17)$$

$$\begin{aligned} \log \alpha_i(t) \approx & \log b_i(y_t) + \\ & \max_{j \in \{1, \dots, N\}} (\log \alpha_j(t-1) + \log A_{ji}) \end{aligned} \quad (4.18)$$

Substitute Eq. (4.18) into Eq. (4.17), rename i and j to z_t and z_{t-1} , then recursively decompose $\log \alpha$, we have:

$$\begin{aligned}
\mathcal{L}_t &= \max_{z_t \in \{1, \dots, N\}} (\log \alpha_{z_t}(t)) \\
&= \max_{z_t \in \{1, \dots, N\}} (\log b_{z_t}(y_t) + \\
&\quad \max_{z_{t-1} \in \{1, \dots, N\}} (\log A_{z_{t-1} z_t} + \log \alpha_{z_{t-1}}(t-1))) \\
&= \max_{z_t \in \{1, \dots, N\}} (\log b_{z_t}(y_t) + \\
&\quad \max_{z_{t-1} \in \{1, \dots, N\}} (\log A_{z_{t-1} z_t} + \log b_{z_{t-1}}(y_{t-1}) + \\
&\quad \ddots \\
&\quad \max_{z_1 \in \{1, \dots, N\}} (\log A_{z_1 z_2} + \log \pi_{z_1} b_{z_1}(y_1) \dots))
\end{aligned} \tag{4.19}$$

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:

$$\begin{aligned}
\mathcal{L}_t &= \max_{z_t \in \{1, \dots, N\}} (\log b_{z_t}(y_t) + \\
&\quad \max_{z_{t-1} \in \{1, \dots, N\}} (\log A_{z_{t-1} z_t} + \log \alpha_{z_{t-1}}(t-1))) \\
&= \log b_{\hat{z}_t}(y_t) + \log A_{\hat{z}_{t-1} \hat{z}_t} + L_{t-1}.
\end{aligned} \tag{4.20}$$

Then, the forward gradient can be derived from Eq. (4.20) as:

$$\nabla \mathcal{L}_t = \mathcal{L}_t - \mathcal{L}_{t-1} = \log b_{\hat{z}_t}(y_t) + \log A_{\hat{z}_{t-1} \hat{z}_t}. \tag{4.21}$$

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 $s \in \{1, \dots, p\}$ and an input time series Y , then the most probable skill \hat{s} generating $Y[t]$ is inferred as:

$$\hat{s} = \arg \max_{s \in \{1, \dots, n\}} (\nabla \mathcal{L}_t^{m_s}(Y)) \quad (4.22)$$

where, $\nabla L_t^{m_s}(Y)$ 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 $i \in \{1, \dots, n\}$ collected from successful executions of skills $s \in S$. To detect anomalies in a new time series Y we can first derive:

$$\nabla \mathcal{L}_{max} = \max_{i \in \{1, \dots, n\}} (\max_{t \in \{1, \dots, T_i\}} (\nabla \mathcal{L}_t^m(Y_i))),$$

$$\nabla \mathcal{L}_{min} = \min_{i \in \{1, \dots, n\}} (\min_{t \in \{1, \dots, T_i\}} (\nabla \mathcal{L}_t^m(Y_i))),$$

$$\nabla \mathcal{L}_{range} = \nabla \mathcal{L}_{max} - \nabla \mathcal{L}_{min},$$

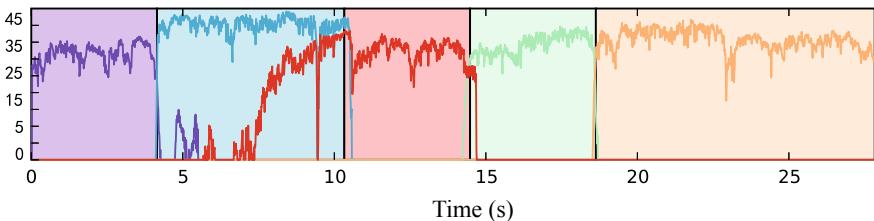


Fig. 4.6 For each skill (denoted in different background color), the log-likelihood gradient can be used for skill identification

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 :

$$\nabla \mathcal{L}_t^m(Y) < \nabla \mathcal{L}_{min} - \frac{\nabla \mathcal{L}_{range}}{2}. \quad (4.23)$$

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:

- (i) hover over the picking position,
- (ii) grasp the object,
- (iii) lift the object,
- (iv) hover to the placing position, and
- (v) 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%.

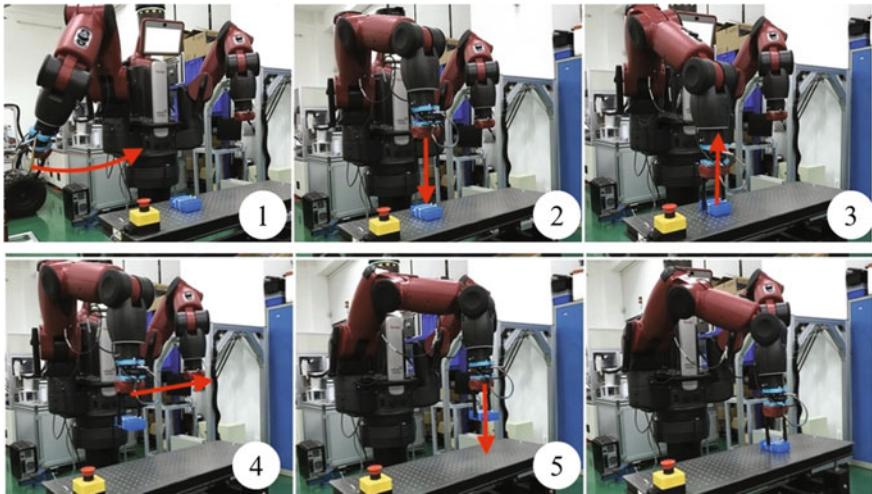


Fig. 4.7 The Baxter humanoid robot performing a pick-and-place task. 5 independent skills used to perform the task. Executed skill motions are sketched with red arrows

Fig. 4.8 Skill identification confusion matrix for pick-and-place

		Skill identification confusion matrix for pick and place task				
		Skill 1	Skill 2	Skill 3	Skill 4	Skill 5
True Label	Skill 1	1.00	0.00	0.00	0.00	0.00
	Skill 2	0.00	0.99	0.00	0.00	0.01
Skill 3	0.00	0.00	1.00	0.00	0.00	
Skill 4	0.00	0.00	0.01	0.99	0.00	
Skill 5	0.00	0.04	0.00	0.00	0.94	
		Skill 1	Skill 2	Skill 3	Skill 4	Skill 5
		Predicted Label				

- **(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: $offset = predicted-t_beginning$, $length=t_end-t_beginning$, and $reaction=offset/length$. 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

Table 4.4 Skill and anomaly identification, and reaction time comparison for state-of-the-art event-detection methods

Technique	ID accuracy (%)
AFF/DCC/CSM/SVM [32]	84.66 ^a
sHDP-HMM [16]	89.50
RCBHT w/multiclass SVM [22]	97.00
HMM w/GradientBased measure (current)	98.40
Tool breakage SVM [33]	99.38
Technique	anomalyID (%) Accuracy
HMM, varying threshold [11]	~80.00
MLP [12]	83.27
sHDP-VAR-HMM, mag metric [21]	~85.00
sHDP-HMM [16]	87.50
RCBHT w/multiclass SVM [22]	97.00
HMM, gradient metric (current)	100.00
Technique	Reaction time (%)
sHDP-VAR-HMM, mag metric [21]	3.70 ^b
HMM, gradient metric (current)	1.84

^a Average of Table 3 is cited from [32], k-fold: gesture classification performance (novice, inter, and expert and both macro/micro levels)

^b Average reaction time for their best multimodal setup

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 $y_n = [y_1, y_2, \dots, y_T]$ 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 $z_t = k \in \{1, 2, 3, \dots\}$, (ii) an initial probabilistic state distributions π_0 generated with Markovian dynamics, and (iii) a transition distribution $\{\pi\}_{k=1}^\infty$. Then, the Markovian structure on the state sequence dictates that for all $t > 1$:

$$\begin{aligned} z_t &\sim \pi_{z_{t-1}}, \\ p(z_1 = k) &= \pi_0, \\ p(z_t = k' | z_{t-1} = k) &= \pi_{kk'}. \end{aligned} \tag{4.24}$$

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 ε and can be expressed as:

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

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 Σ_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 ,

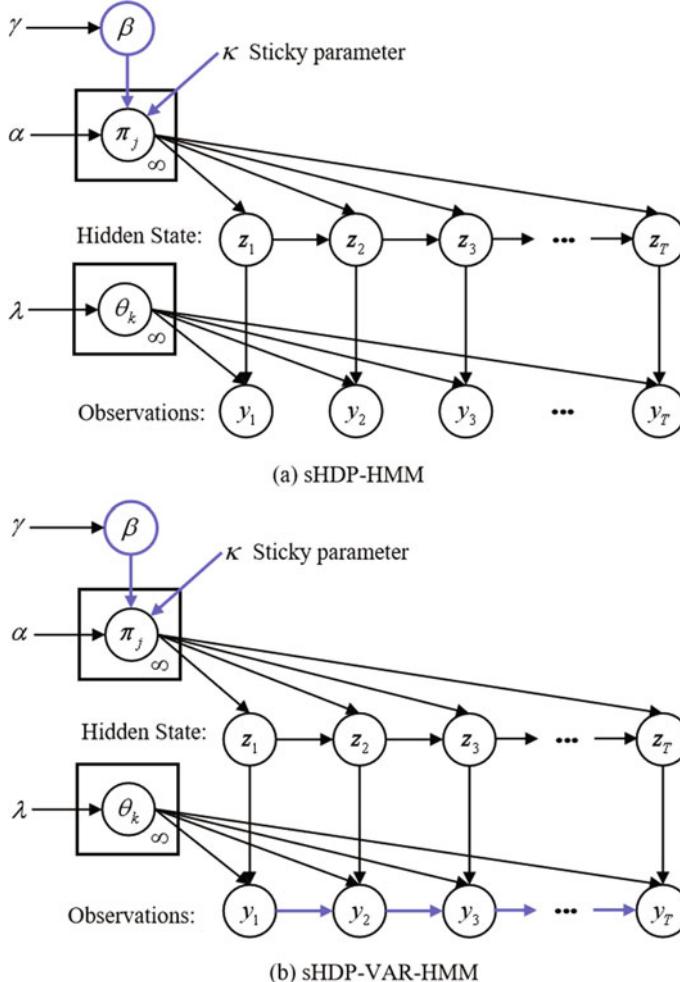


Fig. 4.9 Graphical representation of HDP-HMM and HDP-VAR-HMM. **a** The graphical representation of a sticky HDP-HMM over T time steps. The latent, discrete-valued Markov process z_t captures the temporal dependencies in the observations y_t . **b** The graphical model for a sticky HDP-VAR (r)-HMM over 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}

$$\begin{aligned} \log p(y_t | \theta_k, y_{t-1}) &= \log p(y_t | A_k, \Sigma_k, y_{t-1}) = \log \mathcal{N}(y_t | A_k y_{t-1}, \Sigma_k) \\ &= -\frac{d}{2} \log(2\pi) - \frac{1}{2} \log |\Sigma_k| - \frac{1}{2} (y_t - A_k y_{t-1})^T \Sigma_k^{-1} (y_t - A_k y_{t-1}). \end{aligned} \quad (4.26)$$

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 Σ_k are uncertain. This distribution assume that when only the covariance Σ_k is uncertain, the conjugate prior is defined as d -dimensional Inverse Wishart (IW) distribution with covariance parameter Δ , a symmetric, positive definite scale matrix, and v , the degrees of freedom, i.e.

$$\Sigma_k \sim \mathcal{IW}(v, \Delta), \quad (4.27)$$

where the first moment is given by the mean

$$\mathbb{E}[\Sigma_k] = \frac{v\Delta}{v-d-1}. \quad (4.28)$$

After that, we place a conditionally Matrix-Normal (MN) prior on A_k given Σ_k , which is given by

$$A_k \sim \mathcal{MN}(M, \Sigma_k, V), \quad (4.29)$$

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 Σ_k . Therefore, we derive the useful expectations

$$\begin{aligned} \mathbb{E}(A_k) &= M, \\ \text{Cov}[A_k] &= \frac{1}{v-d+1} (V^{-1} \times \Delta). \end{aligned} \quad (4.30)$$

The resulting log-likelihood function of joint distribution of A_k, Λ_k is defined as $\log p(A_k, \Sigma_k | v, \Delta, M, V^{-1})$

$$\log p(A_j, \Lambda_j | v, \Delta, M, V^{-1}) = F_{\mathcal{IW}}(v, \Delta) + F_{\mathcal{MN}}(M, V) + F, \quad (4.31)$$

where,

$$\begin{aligned} F_{WI}(v, \Delta) &= -\frac{d(d-1)}{4} \log \pi - \frac{d}{2} (\log 2)v - \sum_{i=1}^d \log \Gamma \left(\frac{v+1-d}{2} \right) + \frac{v}{2} \log |\Delta|, \\ F_{MN}(M, V) &= -\frac{d^2}{2} \log 2\pi + \frac{d}{2} \log |V|, \\ F &= \frac{v-1}{2} \log |\Lambda_k| - \frac{1}{2} \text{tr}(A_k^T \Lambda_k A_k V) + \text{tr}(\Lambda_k A_k V M^T) - \frac{1}{2} \text{tr}(\Lambda_k (\Delta + M V M^T)). \end{aligned} \quad (4.32)$$

where $\Gamma()$ is represented the gamma function. Therefore, all the optional hyperparameters of HDP-VAR-HMM are specified by v , Δ , 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 π_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 β over the infinite set of states (see Fig. 4.9). Consider the probability $\sum_{j=1}^{\infty} \beta_j = 1$ on a countably infinite set, where defines the sticking-breaking construction of the prior on β and first draws independent variable $\mu_j \sim Beta(1, \gamma)$ for each state j , and then sets $\beta_j = \mu_j \prod_{\ell=1}^{j-1} (1 - \mu_{\ell})$, where the concentration parameter γ controls the relative proportion of the weights β_j . Here, we interpret μ_j as the conditional probability of choosing state j amongs states $\{j, j + 1, j + 2, \dots\}$

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

$$[\beta_1, \beta_2, \dots, \beta_K, \beta_{>K}] \sim Dir(\alpha\beta_1, \alpha\beta_2, \dots, \alpha\beta_{>K}), \quad (4.33)$$

Generally, we define the starting probability vector π_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 κ of favors self-transition by assigning extra prior mass on the transition probability π_{jj} was introduced. Thus, the transition representation will in a temporally consistent way, that is

$$[\beta_1, \beta_2, \dots, \beta_K, \beta_{>K}] \sim Dir(\alpha\beta_1, \dots, \alpha\beta_j + \kappa, \dots, \alpha\beta_{>K}). \quad (4.34)$$

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-VAR-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 $z_t = k \in \{1, 2, 3, \dots, K\}$. 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 Θ_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.

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

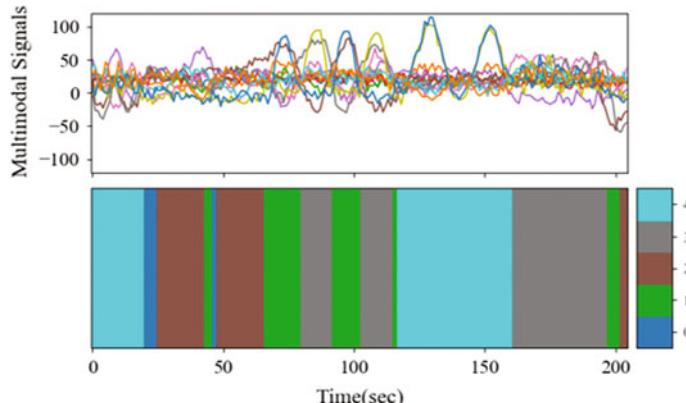


Fig. 4.10 An illustration for the representation of multimodal time series in low dimensional hidden state space

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 $\log p(y_t | \Theta_s) < \rho(z)$, 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 [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, μ and σ 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 $p(y_t | y_{t-1}, \dots, y_1) = p(y_t | y_{t-1})$, that is

$$\begin{aligned} p(y_{0:T}) &= \sum_{t=1}^T p(y_t | y_{1:t-1}), \\ p(y_t | y_{1:t-1}) &= \sum_{k=1}^K p(z_t = k | y_{1:t-1}) p(y_t | z_t). \end{aligned} \tag{4.36}$$

Where θ_{z_t} 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

$$\mathcal{L}_t = \log \sum_{k=1}^K \exp \left(\frac{\alpha_t(k) \cdot \beta_t(k)}{p(y_{0:T})} \right), \tag{4.37}$$

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.

$$\begin{aligned}\alpha_t(k) &= \frac{1}{p(y_t | y_{0:t-1})} p(y_t | z_t = k) p(z_t = k | y_{1:t-1}), \\ p(z_t = k | y_{0:t-1}) &= \sum_j p(z_t | z_{t-1} = j) p(z_{t-1} | y_{0:t-1}).\end{aligned}\quad (4.38)$$

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 $n \in N$ is T_n . Thus, we can get a set of log-probabilities vector of each skill

$$\mathcal{L}^{1:N} = \{\mathcal{L}_{T_1}^1, \mathcal{L}_{T_2}^2, \dots, \mathcal{L}_{T_N}^N\}. \quad (4.39)$$

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.

- **(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

$$\hat{z}_i = \arg \max_{z_i} p(z_i | y_{0:T}). \quad (4.40)$$

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

$$\hat{z}_{1:T} = \arg \max_{z_{1:T}} p(z_{1:T}, y_{0:T}). \quad (4.41)$$

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_t .

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

$$\mathcal{L}_T(z_{1:T}) = s_1(z_1) + \sum_{t=2}^T s_t(z_t, \pi_{z_{t-1}}), \quad (4.42)$$

where

$$\begin{aligned} s_1(z_1) &= \log(\pi_{z_1} p(y_1 | \theta_{z_1}, y_0)), \quad t = 1 \\ s_t(z_t, z_{t-1}) &= \log(p(z_t | z_{t-1}) p(y_t | \theta_{z_t}, y_{t-1})), \quad t \geq 2 \end{aligned} \quad (4.43)$$

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

$$\begin{aligned} v_t(z_t) &= \max_{z_{1:t-1}} \mathcal{L}_t(z_{1:t}), \quad z_t \in \{1, 2, 3, \dots, K\} \\ v_t(z_{t-1}) &= \max_{z_t} \{v_{t-1}(z_t) + s_t(z_t, z_{t-1})\}, \end{aligned} \quad (4.44)$$

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

$$\hat{z}_T = \arg \max_{z_{t-1}} v_n(z_{t-1}), \quad (4.45)$$

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

$$\mathcal{Z}^{1:N} = \{\mathcal{Z}_{T_1}^1, \mathcal{Z}_{T_2}^2, \dots, \mathcal{Z}_{T_N}^N\}. \quad (4.46)$$

• (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 $\mathcal{L}_{z_k}^{\max}$ and minimum $\mathcal{L}_{z_k}^{\min}$. We cluster the log-likelihoods with the same state, which represents the clustering data by HMM belong to the same statistical distribution.

$$\begin{aligned} \mathcal{L}_{z_k}^{\max} &= \max_{i \in \{1, \dots, N\}} \mathcal{L}_i^n \mathbf{1}(z_k == k), \\ \mathcal{L}_{z_k}^{\min} &= \min_{i \in \{1, \dots, N\}} \mathcal{L}_i^n \mathbf{1}(z_k == k), \\ \mathcal{L}_{z_k}^{\text{range}} &= \mathcal{L}_{z_k}^{\max} - \mathcal{L}_{z_k}^{\min}, \end{aligned} \quad (4.47)$$

where the notation $\mathbf{1}(\eta)$ to represent an indicator random variable that is 1 if the event η 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:

$$\rho(z_k) = \mathcal{L}_{z_k}^{\min} - \frac{\mathcal{L}_{z_k}^{\text{range}}}{2}. \quad (4.48)$$

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.

$$z_{test} = \arg \max_{1, \dots, K} \{p(z_t = 1 | y_{0:t}), \dots, p(z_t = K | y_{0:t})\}. \quad (4.49)$$

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-VAR-HMM and streaming sensor data, the detector can recursively compute the z_t and \mathcal{L}_t at time t , and then perform the comparison, if $\mathcal{L}_t < \rho(z_{test})$, 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 → Pre-pick; (Skill 4) : Pre-pick → Pick; (Skill 5) : Pick → Pre-pick; (Skill 7) : Pre-pick → Pre-place; (Skill 8) : Pre-place → Place; (Skill 9) : Place → Pre-place that were effectively

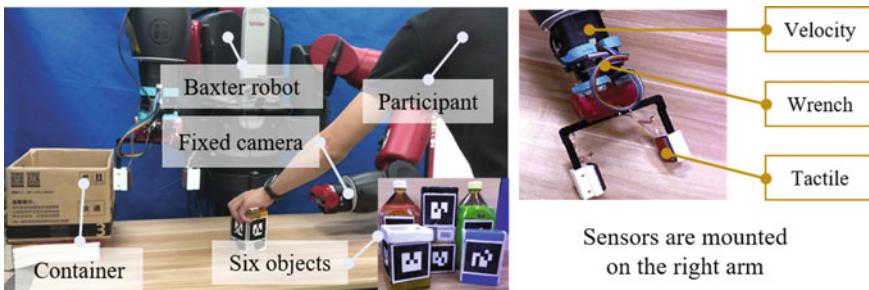


Fig. 4.11 A kitting experiment consists of 6 skills (in various color) that were modeled by DMP, respectively. Objects that need to be packaged are placed by a human collaborator before the robot in a collection bin. The shared workspace affords possibilities for accidental contact and unexpected alteration 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.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 4×7 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 1280×800 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 $3.2 \times 10^{-4} \text{ m}^3$ to $1 \times 10^{-3} \text{ m}^3$. 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 tags² 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.

¹<http://www.ros.org/SMACH>.

²http://wiki.ros.org/ar_track_alvar.

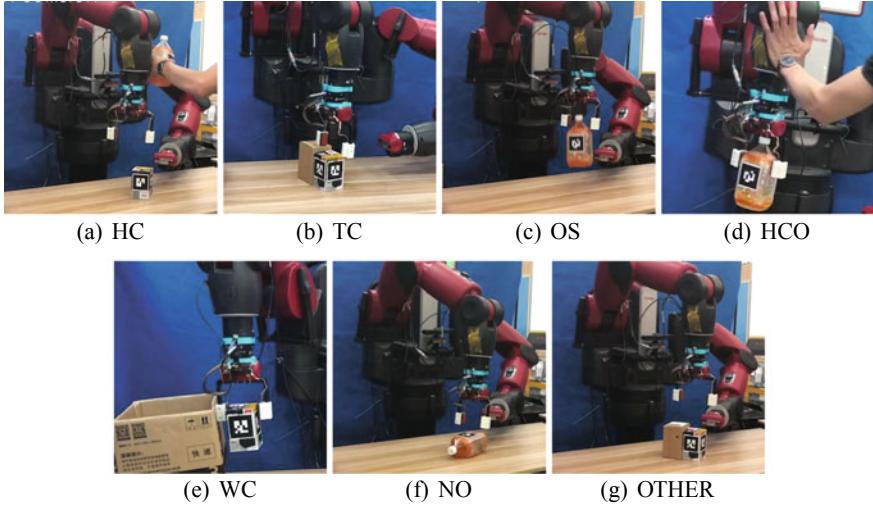


Fig. 4.12 Seven representative unexpected anomalies in the kitting experiment (left-to-right): Human Collision (HC), Tool Collision (TC), Object Slip (OS), Human Collision with Object (HCO), Wall Collision (WC), No Object (NO), Other (OTHER)

- (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:

$$n_f = \sqrt{f_x^2 + f_y^2 + f_z^2}, \quad n_t = \sqrt{t_x^2 + t_y^2 + t_z^2}. \quad (4.50)$$

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:

$$n_l = \sqrt{l_x^2 + l_y^2 + l_z^2}, \quad n_a = \sqrt{a_x^2 + a_y^2 + a_z^2}. \quad (4.51)$$

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:

$$s_l = \sqrt{\frac{1}{28} \sum_{i=1}^{28} (l_i - \mu_l)^2}, \quad s_r = \sqrt{\frac{1}{28} \sum_{i=1}^{28} (r_i - \mu_r)^2}. \quad (4.52)$$

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:

$$\mathbf{y}_t = \left[\frac{f_x}{10}, \frac{f_y}{10}, \frac{f_z}{10}, \frac{t_x}{10}, \frac{t_y}{10}, \frac{t_z}{10}, \frac{n_f}{10}, \frac{n_t}{10}, \frac{n_a}{0.4}, l_x, l_y, l_z, a_x, a_y, a_z, \frac{n_l}{0.4}, n_a, \frac{s_l}{250}, \frac{s_r}{250} \right]. \quad (4.53)$$

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 Λ_k . The conjugate prior is the MNIW for which four parameters v , Δ , 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 $v = d + 2$ and we set Δ by constraining the mean or expectation of the covariance matrix $\mathbb{E}[\Lambda_k^{-1}]$ under the Wishart prior in Eq. (4.30).

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

Assume that we record N sequential data for each skill and the length of sequence $n \in N$ is T_n . Thus, we can easily define the parameter Δ accordingly as

$$\Delta = (v - d - 1)\mathbb{E}[\Lambda_k^{-1}]. \quad (4.55)$$

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).

$$\begin{aligned} M &= \mathbf{0}_d, \\ V &= 1.0 * \mathbf{I}_d. \end{aligned} \quad (4.56)$$

where $\mathbf{0}_d$ and \mathbf{I}_d are the matrix of all zeros and the identity matrix, respectively, each of size $d \times d$. For the concentration parameters, a $Gamma(a, b)$ prior is set on HDP concentration parameters α . A $Beta(c, d)$ prior is set on the self-transition parameter μ and the degree of self-transition bias κ 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 $\mu \sim Beta(1, 10)$. 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 $zhat = 2$ 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

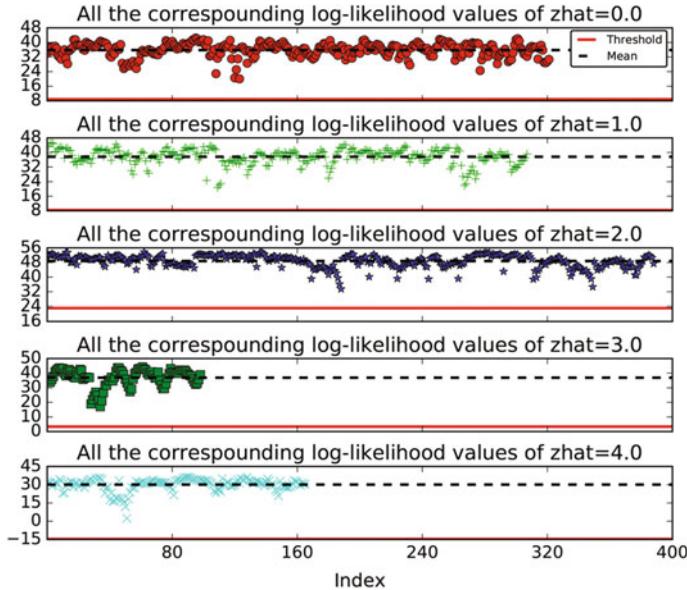


Fig. 4.13 All the log-likelihood values of individual hidden state of skill 3 when the truncation states number is set to $K = 5$. Each subplot shows the mean value (black dotted line) and the corresponding log-likelihood threshold value (red solid line) in Eq. (4.48)

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%.

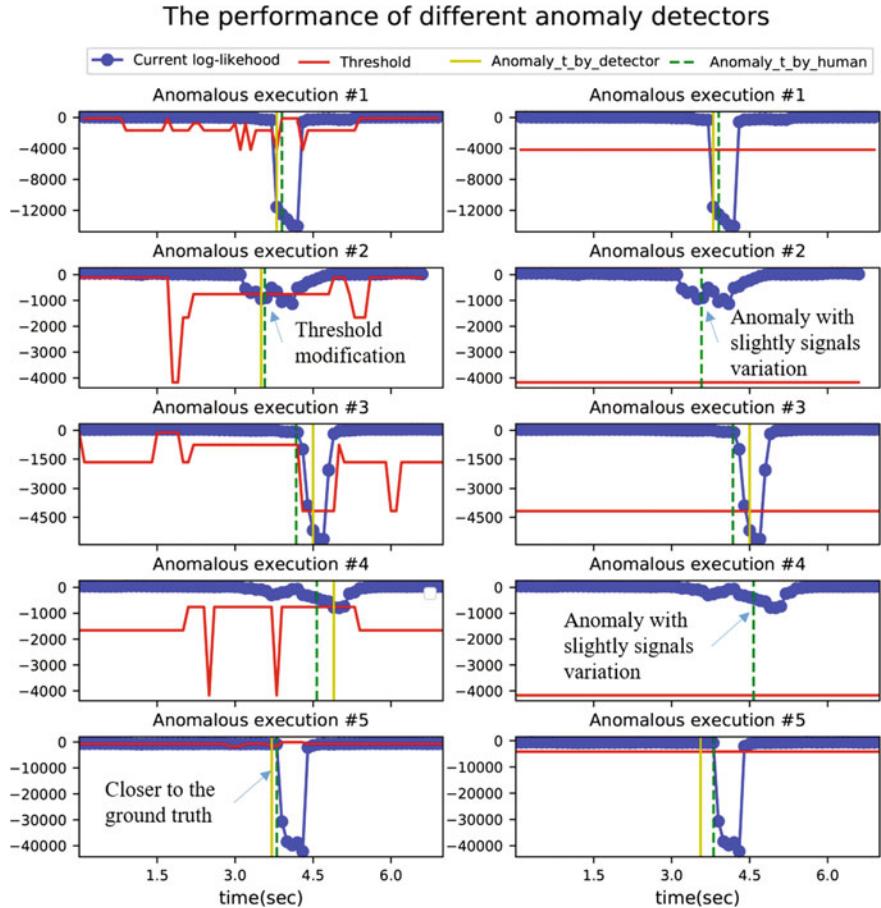


Fig. 4.14 The comparison of two anomaly detectors for five robot anomalous executions. Each row represents the same testing execution. *Left column*: results of our hidden state-based anomaly detector. *Right column*: results of the gradient-based anomaly detector [21]. The ground truth (*anomaly_t_by_human*) 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%).

Table 4.5 The comparison of anomaly monitoring performance measures between our proposed method and a baseline method

Kitting experiment	Testing information		Precision		Recall		F1-score		Accuracy	
	Success	Anomalies	GD	HSD	GD	HSD	GD	HSD	GD	HSD
Movements										
Home → Pre-prick	141	6	1.0	1.0	0.33	0.50	0.50	0.67	0.973	0.980
Pre-prick → Prick	112	23	1.0	1.0	0.78	0.78	0.878	0.88	0.963	0.963
Prick → Pre-prick	83	28	1.0	0.80	0.04	0.31	0.069	0.45	0.757	0.820
Pre-prick → Pre-place	13	70	1.0	0.84	0.54	0.97	0.704	0.90	0.614	0.843
Pre-place → Place	11	6	1.0	0.40	1.0	1.0	1.0	0.57	1.0	0.769
Place → Pre-Place	8	3	0.0	0.0	0.0	0.0	0.0	0.0	0.889	0.889
Across all skills	368	136	0.98	0.84	0.47	0.76	0.64	0.80	0.861	0.910

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. Sülch 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](https://arxiv.org/abs/1602.07109).
20. Milacski ZÁ, 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, Martini 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. Fox EB, Suderth EB, Jordan MI, Willsky AS. Bayesian nonparametric methods for learning Markov switching processes. *IEEE Signal Process Mag*. 2010;27(6):43–54.
27. Fox EB, Hughes MC, Suderth EB, Jordan MI. Joint modeling of multiple time series via the beta process with application to motion capture segmentation. *Ann Appl Stat*. 2014;8(3):1281–313.
28. Hughes MC, Stephenson WT, Suderth E. Scalable adaptation of state complexity for nonparametric hidden Markov models. In: Advances in neural information processing systems. 2015. p. 1198–206.
29. Murphy KP. Machine learning: a probabilistic perspective. Cambridge: MIT Press;2012.
30. Pettersson O. Execution monitoring in robotics: a survey. *Robot Auton Syst*. 2005;53(2):73–88.
31. Wu H, Luo S, Lin H, Duan S, Guan Y, Rojas J. Recovering from external disturbances in online manipulation through state-dependent revertive recovery policies. In: 2018 27th IEEE international symposium on robot and human interactive communication (RO-MAN). IEEE;2018. p. 166–73.

32. Ahmadi 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.



Chapter 5

Nonparametric Bayesian Method for Robot Anomaly Diagnose



Abstract In this chapter, we introduce two novel anomaly diagnose methods using the Bayesian nonparametric hidden Markov models when anomaly triggered, including i) multi-class classifier based on nonparametric models, ii) sparse representation by statistical feature extraction for anomaly diagnose. Additionally, the detail procedure for anomaly sample definition, the supervised learning dataset collection as well as the data augmentation of insufficient samples are also declared. We evaluated the proposed methods with a multi-step human-robot collaboration objects kitting task on Baxter robot, the performance and results are presented of each method respectively.

5.1 Introduction

In addition, anomaly diagnoses refers to the process of clearly classifying anomalies using supervised learning on the premise that the robot detects the occurrence of anomalies. Due to the uncertainty, diversity and unpredictability of anomalies, the robot's anomaly diagnoses can effectively improve the safety performance of the robot system and provide guarantee for subsequent anomaly repair behaviors. Therefore, the online abnormal monitoring and diagnoses of robots is the focus and difficulty of robots' long-term autonomous operation.

5.2 Related Work

The robot introspection not only needs to implement the movement identification and monitoring of anomalies, but also to diagnose the types of anomalies for providing sufficient support in subsequent recovery. The samples used for anomaly diagnoses in this chapter would have the same modal information as in anomaly monitoring and belong to multidimensional time series. Since the problem of multidimensional time series classification is a supervised learning problem [15–17], which aims to determine the labels of multidimensional time series of the same structure

and length, Orsenigo et al. [18] proposed a discrete support vector machine (Discrete Support Vector Machine) Machine (DSVM) classification method. This method benefits from the concepts of warping distance and softened variable margin and realizes the classification of multidimensional time series. Lee et al. [19] proposed a time series classification method based on K-Nearest Neighbor [20] (K-Nearest Neighbor, KNN), which was successfully used to evaluate the traffic prediction application of the mobile telecommunications industry. Seto et al. [21] combined dynamic time warping [8, 22] (DTW) and KNN to derive a multivariate time series classification based on dynamic time warping template selection, and applied it to human activity recognition. In addition to the above-mentioned distance-based multidimensional time series classification method, the method of establishing feature vectors through dimensionality reduction methods has also received attention [23–25]. Nanopoulos et al. [26] proposed a method based on statistical feature extraction to establish feature vectors, and then trained a multi-layer perceptron (MLP) from the feature vectors and target categories to achieve the classification of time series. Utomo et al. [27] proposed to classify the multidimensional data of the hospital intensive care unit by the method of multidimensional compression description (MultiCoRe), which extracted the features including time domain and frequency domain for classification. Jaakkola et al. [28] introduced a method combining HMM and SVM for classifying protein domains. This method can also be applied to other fields of biological sequence analysis. Raman et al. [29] proposed the use of a multi-layer HDP-HMM modeling method to achieve classification of human movements, training HDP-HMM from all movement motion samples, and building a multi-objective classifier from this model. Test the size of the log-likelihood function of the specimen to achieve classification. Dilello et al. [30] used sHDP-HMM to classify robots with multiple modal anomalies. The model assumes that the observations are independent of each other, which weakens the correlation between abnormal data to some extent. Although the training of neural networks with small samples is easy to cause overfitting and difficult to achieve online monitoring, but because of its better modeling capabilities and less data pre-processing process, multivariate time based on deep learning Sequence classification is currently being extensively studied by scholars [31–34].

Besides the aforementioned that multivariate time series classification has received significant interest in areas such as healthcare, object recognition, and human action recognition [35–37]. Meanwhile, most of the solutions to the classification problem of multidimensional time series are solved by supervised learning, that is, the multi-objective classifier learning is performed by manually labeled samples in advance by humans. In view of this, the anomaly diagnoses problem considered in this chapter is mainly for the classification of conventional anomaly samples with multimodel observation of robotic systems. In robotics, complex semi-autonomous systems that have provided extensive assistance to humans, anomalies are occur occasionally [39]. For this reason, enabling accurate, robust, and fast anomaly monitoring and diagnoses in robots has the potential to enable more effective, safer, and more autonomous systems [38, 49]. Execution monitoring, especially with the focus of detecting and classifying anomalous executions has been well studied in robotics [40]. Daehyung Park et al.[39] introduced a multimodal execution monitoring system to detect and

classify anomalous executions for robot-assisted feeding. Specifically, the classifier labelled not only the anomaly type but also the cause of the anomaly by using an artificial neural network system. The neural net successfully integrated the anomaly monitoring and diagnoses for assisting a person with severe quadriplegia. However, due to the HMM anomaly detector limitations, the classified accuracy for anomaly types and causes are 90.0%, 54.0% respectively. In [41], Bjreland et al. introduced a monitoring system that also can detect, classify, and correct anomalous behaviors using a predictive model. Those two integrated system give us a lot of inspiration and confidence for improving the robot introspection with the anomaly monitoring and diagnoses.

The anomaly diagnoses has been used to determine the source of anomalies while running manipulator or mobile robots [42]. Several common time series classification algorithms are distance based metrics, such as the k-nearest neighbors (KNN) approach have proven to be successful in classifying multivariate time series [43]. Plenty of research indicates Dynamic Time Warping (DTW) as the best distance based measure to use along KNN [44]. Besides the distance based metrics, the feature based algorithms have been used over the years [45], which rely heavily on the features being extracted from the time series data or modeling the time series with the parametric methods. Hidden State Conditional Random Field (HCRF) and Hidden Unit Logistic Model (HULM) are two successful feature based algorithms that have led to state of the art results on various benchmark datasets [46]. However, HCRF is a high computational efficiency algorithm that detects latent structures of the input time series using a chain of k -nominal latent variables. Further, the number of parameters is linearly increasing of latent states required. To overcome this, HULM proposes using H binary stochastic hidden units to model 2^H latent structures of the data with only $O(H)$ parameters. Another approach for multivariate time series classification is by applying dimensional reduction techniques or by concatenating all dimensions of a multivariate time series into a univariate time series [47].

5.3 Problem Statement

Another main task of robotic introspection is to accurately evaluate the potential pattern of real-time multi-modal sensing data. It is not limited to the identification of robot movement and abnormal monitoring, but also has the ability to classify conventional abnormalities. Anomaly diagnoses mainly refers to the identification of conventional learned abnormal types through supervised learning after the robot detects the occurrence of abnormalities. An intelligent robot system includes but is not limited to the following types of abnormalities: system abnormalities, internal sensors anomalies, motion instruction failures, noise or damage to sensors and actuators, robot-human-environment interactions, or external disturbances in the environment. Park et al. [50] carried out a diagnoses of the types of robot anomalies and their causes, and integrated this function into a human-robot interactive robot system for feeding disabled people. The types of anomalies considered in this section mainly

come from external disturbances caused by system anomalies (kinematic anomalies, sensor failures, etc.) or human improper interference behavior or robot ends in a human-machine collaboration environment. In order to improve the compactness and portability of the proposed robotic sensing system, this section still adopts the method of non-parametric Bayesian hidden Markov model to implement multi-class anomaly diagnoses.

5.4 Collection and Augmentation of Anomaly Sample

The collection of anomalous samples in this book takes into account the precursory period and the duration of a certain period of time when the anomaly occurs, which is determined by the value of the given anomalous window range win_len (for ease of adjustment, the duration of the precursory period and duration is usually equal, both are half of win_len). As shown in Fig. 5.1, a sample with an anomaly type of “tool collision” is extracted (light red background area) around the trigger time (black vertical dashed line) of the previous anomaly monitor.

In addition, due to the great randomness and uncertainty of the occurrence of anomalies, the number of samples between different types of anomalies is extremely unbalanced, and for some anomalies (collision between the robot and the environment), the impact on the robot body may even cause damage There is no guarantee

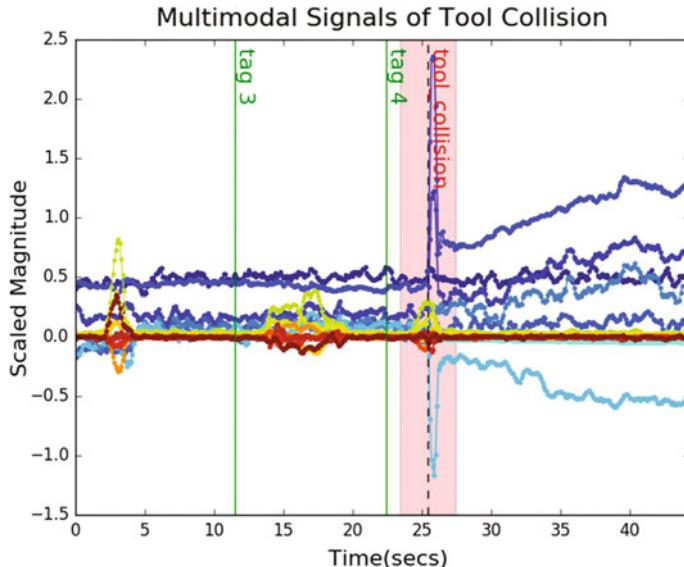


Fig. 5.1 An example of extracting an “tool collision” anomaly sample with a given when anomaly detected

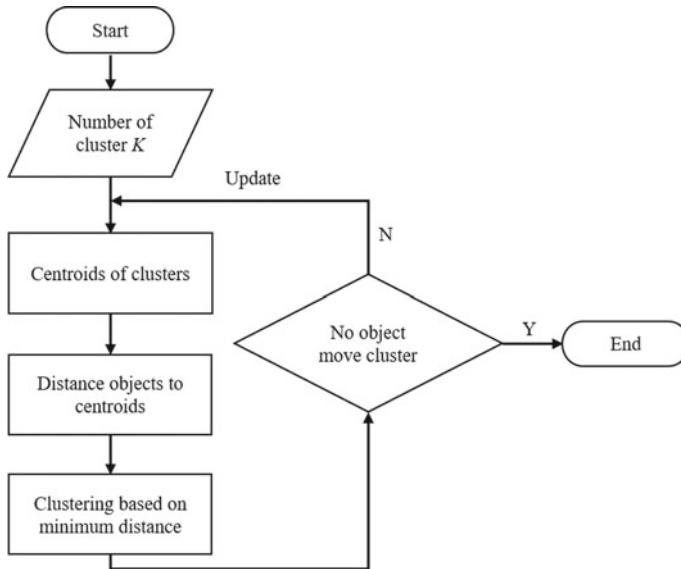


Fig. 5.2 The implementation flowchart of K-means Algorithm

for the number of samples for model training. In view of this problem, this article considers the data enhancement processing for the abnormal types of fewer samples collected. The main goal is to generate synthetic data based on the statistical characteristics of the data.

The proposed multimodal augmentation method is inspired by the *K – means* clustering method in traditional machine learning, combined with Dynamic Time Warping (DTW) to enhance the data. Among them, the implementation flowchart of the *K – means* clustering method is shown in Fig. 5.2.

In the algorithm, K represents the number of categories, Means represents the mean. As the name suggests, this method is a method for Dimensional or multidimensional data points, and even the time series considered in this article) clustering algorithm, the core idea is to use a preset K value and the initial centroid of each category (Centroid) to distance (Euclidean distance, Manhattan distance and Time series distance measure (DTW) is used to cluster the data points that are closer, so that the average value of the clustered iteration is preferentially obtained with the smallest objective function value, as shown in the following formula:

$$J = \sum_{j=1}^K \sum_{i=1}^N \|x_i^{(j)} - c_j\|^2 \quad (5.1)$$

where, the symbol J is the objective function of the *K – means* algorithm; K is the number of clusters; N is the number of data points; $\|\cdot\|$ is expressed as the distance

Algorithm : Data Augmentation for Sparse Anomaly Class

Input: $\{y_n\}_n^{N_c}$: Sparse dataset with N_c data-points (sequences), each of length T_n ;
 N : Number of data-points for class $c \in \mathcal{C}$;
 N_b : For determining the number of centroids for K-means clustering
 N_k : Number of the iteration of K-means;
 N_d : Number of runs for DTW averaging algorithm;
 K : Number of centroids for K-means clustering
Result: Synthetic anomaly sample generation: $\{\hat{y}_s\}_s^{N_s}, N_s > N_c$

```

1 K ← ceil(|N|/Nb). ceil(x): The smallest integer greater than or equal to x
2 while Nk > 0 do
3   CENTROIDS ← pick K random data points from {yn}nNc;
4   Assign each data point yn to the nearest CENTROID using DTW;
5   Remove CENTROIDS with only one data-point;
6   for CENTROID in CENTROIDS do
7     | CENTROID ← update the centroid using the subset of data-points which are assigned to centroid;
8   end
9   {ŷs}sNs ← append the CENTROIDS
10  Nk ← 1
11 end

```

Fig. 5.3 The pseudocode for data augmentation for sparse anomaly class

function between the data points and the centroid; and c represents the centroid of the category. From the implementation of the $K - means$ algorithm, we know that by clustering data points close to the centroid, and data points in the same category have similar statistical characteristics, the goal of satisfying data enhancement is to hope to generate similar data based on sparse dataset New data for characteristics. To this end, the proposed algorithm, after initializing K random centroids, loops through the following two steps: (1) Dividing. Divide each data point to the nearest centroid according to the distance measurement of DTW; (2) Update. According to the selected mean measurement method, the centroid is moved to the center of various types of designated data points. This time, each time the updated centroid is used to generate a composite data, the algorithm's pseudo code is shown in Fig. 5.3.

Meanwhile, an example of one-dimensional data enhancement for an abnormal type is shown in Fig. 5.4. As can be seen from Fig. 5.4, the proposed algorithm can effectively capture the statistical (such as peak and mean) characteristics of the original data.

5.5 Anomaly Diagnose Based on Nonparametric Bayesian Model

The anomaly diagnoses is triggered once an anomaly is detected. A system can possibly address a wide variety of types of anomalies including low-level hardware anomalies: sensor and actuator noise or breakage; mid-level software contingencies like: logic errors or run-time exceptions; high-level misrepresentations: poor modeling of the robot, the world, their interactions, or external disturbances in the

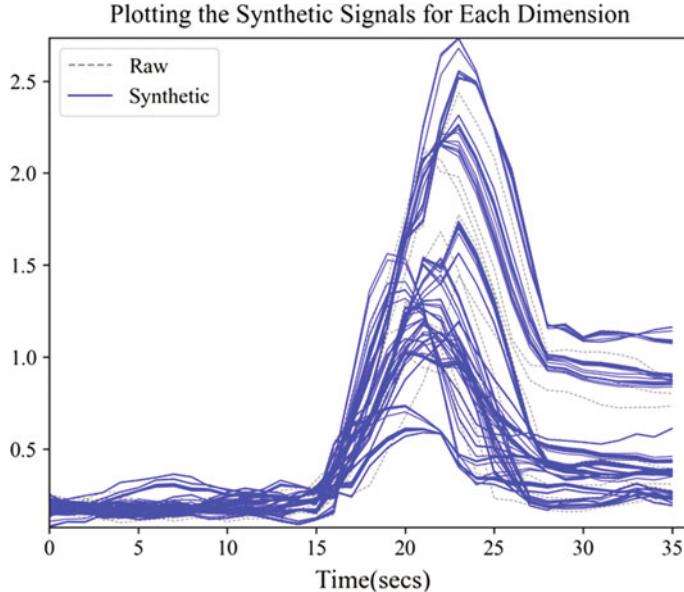


Fig. 5.4 An example for illustrating the data augmentation of anomaly sample

environment). In [48, 50], Park et al. identified both the anomaly class and the cause. In this work, we deal with anomalies caused by external disturbances generated either by intrusive human behavior or resulting from poor modeling or anticipatory ability on the robot's end.

5.5.1 Multiclass Classifier Modeling

We consider the problem of multiclass classification based on multivariate time series, that is, to find a function $f(y^n) = c^n$ given the *i.i.d* training data $Y = \{y^n\}_{n=1}^N$, $C = \{c^n\}_{n=1}^N$, where $y^n = [y_0^n, y_1^n, \dots, y_{T_n}^n]$ is an observation sequence and $c^n \in \{1, \dots, c\}$ its corresponding anomaly class. An observation $y_t^n \in \mathbb{R}^d$ consists of the same multimodal features as described in Sect. 4.6.1 at time-step t . Our objective is diagnoses, where given a new test observation sequence \hat{y} , we have to predict its corresponding anomaly class \hat{c} . Here, the \hat{y} is recorded by considering a *window_size* around the anomaly occurred moment, and the \hat{c} is labeled manually during training procedure. We represent each anomaly class by a separate sHDP-VAR-HMM as outlined in Sect. 4.6.1, the Θ_c are the parameters for class c . It would be straightforward to estimate the posterior density of parameters $p(\Theta_c|Y, C)$ by

Algorithm 1: Training model for anomaly diagnosis

Input:

- N_c : Number of sequences for class $c \in \mathcal{C}$;
- $\{y_n\}_n^{N_c}$: Dataset with N_c sequences, each of length T_n ;
- N_i : Number of the maximum iteration for learning;
- N_r : Number of runs for the whole learning;
- random_state*: The random number generator;
- k_splits*: Number of folds;
- a, b, d, e : Hyper-prior for concentration parameters;
- v, Δ, V, M, s_F : Hyper-prior for the MNIW distribution;
- κ : The self-transition bias;
- K : The truncation active states.

Result: sHDP-VAR-HMM models for each class

$$\{y_n\}_{train}, \{y_n\}_{test} = \text{KFold_split}(\{y_n\}_n^{N_c}, k_splits, random_state)$$

```

for  $k$  in  $k\_splits$  do
    for  $i$  in  $N_r$  do
        for  $n$  in  $N_i$  do
            if not converged then
                |  $\Theta_c, loss = \text{HDPHMM}(\{y_n\}_{train}, a, b, d, e, v, \Delta, V, M, s_F, \kappa, K)$ 
            else
                | sHDP-VAR-HMM with  $\Theta_c$  and  $loss$ 
            end
        end
         $\{loss\}_i^{N_r} \leftarrow loss$ 
        if  $i == N_r$  then
            |  $\Theta_c \leftarrow$  with the minimum loss value
        end
    end
     $\mathcal{L}_{test\_mean} = \frac{\sum_{i=1}^{N_{test}} p(\{y_n\}_i | \Theta_c)}{N_{test}}, i \in N_{test}$ 
     $\{\mathcal{L}_{test\_mean}\}_k^{k\_splits} \leftarrow \mathcal{L}_{test\_mean}$ 
    if  $k == k\_splits$  then
        | return  $\Theta_c$  with the maximum  $\mathcal{L}_{test\_mean}$ 
end

```

Fig. 5.5 An example for illustrating the progress for training nonparametric Bayesian model for anomaly diagnoses

$$p(\Theta_c | Y, C) = p(\Theta_c) \prod_{c^n=c} p(y^n | \Theta_c) p(c). \quad (5.2)$$

That is, each sHDP-VAR-HMM is trained separately and a conditional density $p(y|c)$ for each class is trained throughout the process as defined in Fig. 5.5.

5.5.2 Experiments and Results

- **Anomaly Dataset in Kitting Experiment**

The dataset captures sensory-motor signatures of the Kitting task under anomalous scenarios as outlined in this paper. A total of 136 anomalous events were recorded in 142 experimental executions across all skills. The proportions for each anomaly class were as follows: TC 15.7%, HC 16.7%, HCO 16.7%, NO 13.0%, OS 16.7%, WC 15.7%, and OTHER 5.6%. To intuitively explain, analyze, and propose corresponding diagnose methods, all the anomalies identified using our proposed method are taken into further consideration. That is, we extract all the anomaly data points from each abnormal movement and concatenate them with labels, in which the anomalous data point is also restricted with the same features as considered in anomaly identification. Those identified anomalies are visualized via the t -distributed Stochastic Neighbor Embedding (t-SNE) method [51] and labelled manually, as shown in Fig. 5.6.

- **Anomaly Diagnoses Window Considerations and Online Recording**

Note that when an anomaly monitoring is flagged, we consider a window of time duration $\pm \text{window_size}$ for the subsequent anomaly diagnoses.¹ In cases where an anomaly is detected towards the beginning of a skill execution, and the duration of existing data prior to the detection is less than our window_size . In this work, we do not extract data for diagnoses as we deem a minimal presence of signals before and after the detection crucial for the diagnoses. The sensory-motor data collected at this stage, allows to build basic models of the anomalies described in Chap. 4.

Our system recorded sensory data online. Upon detection of an anomaly, the time-step at which the flag occurred is recorded. Then, we record (online) the multimodal signatures (F/T, Cartesian velocity, and Tactile signals) before and after the anomaly (also referred to as the sample) according to the window_size . Signals were resampled at 10Hz to achieve temporal synchronization for all modalities and the further preprocessed as described in preprocessing step.

- **Parameter Settings and Results**

To avoid overfitting, we performed *3-fold* cross validation for each anomaly type separately. An 18 dimensional feature vector was used as presented in Equation 4.53. We compare against five baselines consisting of parametric HMMs (with a fixed number of hidden states), various observation models, and various variational inference methods. The training and testing dataset for all the diagnoses methods was the same in this comparison.

(1) *Parametric HMM Settings:* We train 3 differing types of HMMs for each anomaly class. Each HMM uses four different numbers of hidden states $K \in \{3, 5, 7, 10\}$ to train each class. We need to estimate the transition matrix, mean, and covariance parameters. To this end, K-Means++ [52] clusters the data and yields

¹We use the Redis database for this purpose (<https://redis.io/>) as rosbags can only be processed offline.

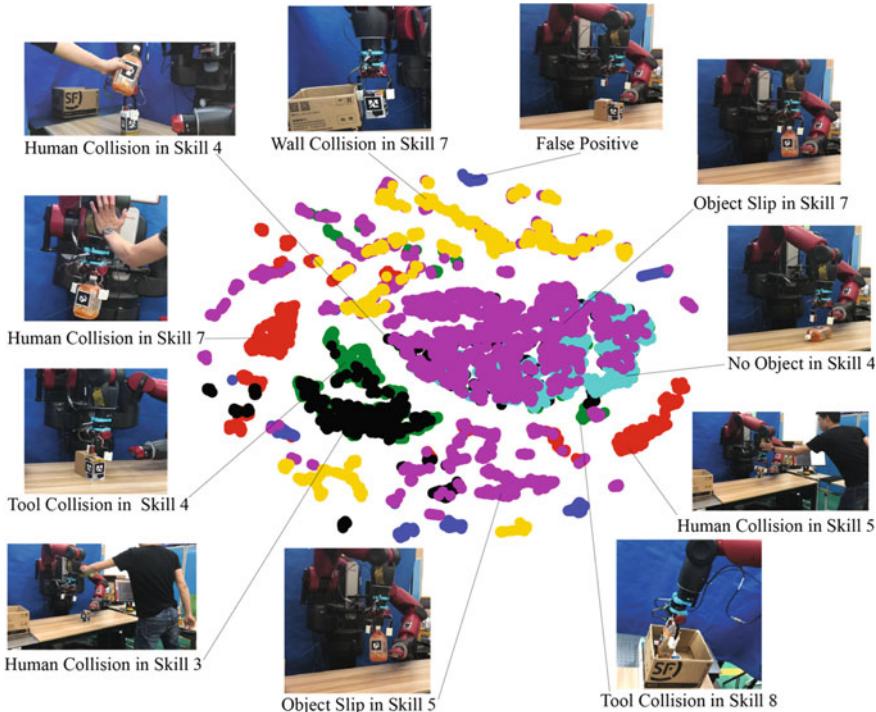


Fig. 5.6 Visualization of all the identified anomalies using t-SNE method. We heuristically label the clusters according to the analysis of unexpected anomalies and the data points density in such a kitting experiment

initial estimates. During testing, we evaluate a test sample against all classes and select the class with the largest log-likelihood. The parametric HMMs result summary is presented in Table 5.1.

- **HMM-Gauss-EM:** A classifier based on a classical HMM with Gaussian observations was trained independently for each anomaly class. The standard Baum-Welch Expectation Maximization algorithm [53] was used to learn the HMM model.

- **HMM-Gauss-VB:** A classifier based on classical HMM with Gaussian observation was trained independently for each anomaly class. The standard Variational Bayesian (VB) inference algorithm [54] was used to learn the HMM model.

- **HMM-AR-VB:** A classifier based on a classical HMM with first-order autoregressive Gaussian observations was trained independently for each anomaly class. The standard VB inference algorithm [54] was used to learn the HMM model.

In conclusion, we find that for parametric HMMs (i) the best diagnoses accuracy rate was 95.7% when using 5–7 fixed states; (ii) Variational inference algorithms outperformed the standard EM algorithm; (iii) The Autoregressive observation model classified better than the Gaussian model due to its linear conditional nature; (iv) Parametric HMMs, in general, are less effective to jointly model the dynamics of the

Table 5.1 The diagnoses accuracy of both classical parametric HMMs and non-parametric HMMs. The best model (observation model and number of hidden states) for each anomaly type is highlighted in blue

States	Methods	TC	HC	HCO	NO	OS	WC	OTHER	Total Accuracy
3	HMM-Gauss-EM	1.0	0.89	0.89	0.78	0.83	1.0	0.67	0.876
	HMM-Gauss-VB	1.0	0.94	0.95	0.96	0.88	1.0	0.67	0.915
	HMM-AR-VB	1.0	0.89	0.90	0.93	0.94	1.0	0.67	0.929
5	HMM-Gauss-EM	1.0	0.78	0.84	0.79	0.79	1.0	0.67	0.838
	HMM-Gauss-VB	0.94	0.94	0.89	0.86	0.85	1.0	0.67	0.889
	HMM-AR-VB	1.0	0.89	0.95	1.0	0.98	1.0	0.67	0.957
7	HMM-Gauss-EM	1.0	0.83	0.95	0.71	0.77	0.83	0.67	0.826
	HMM-Gauss-VB	1.0	0.83	0.86	0.79	0.82	1.0	0.67	0.863
	HMM-AR-VB	1.0	0.95	1.0	0.93	0.96	1.0	0.67	0.957
10	HMM-Gauss-EM	0.94	0.72	0.85	0.86	0.79	0.94	0.67	0.826
	HMM-Gauss-VB	0.94	0.83	0.95	0.72	0.86	1.0	0.67	0.876
	HMM-AR-VB	0.95	0.83	0.82	0.93	0.92	0.95	0.67	0.889
10	HDPHMM-Gauss-VB	0.94	0.95	0.95	0.86	0.94	0.90	0.67	0.915
	HDPHMM-Gauss-moVB	0.88	0.95	0.95	0.93	0.94	0.95	0.83	0.929
	HDPHMM-AR-VB	0.94	0.84	0.86	0.93	0.96	0.95	0.83	0.915
	HDPHMM-AR-moVB	1.0	0.95	0.95	0.93	0.98	1.0	1.0	0.971

robotic task. Therefore, we consider a Bayesian nonparametric version of HMMs with a hierarchical prior which shares statistical strength across the training samples.

(2) *Nonparametric HMMs Setting*: We also train 4 kinds of classifiers base on non-parametric HMMs, independently for each anomaly class. We specify the truncation number of states as K=10 as explained in Sect. 4.6.1. Comparing to the parametric HMMs, the actual number of hidden states of each anomaly class is automatically learned from data in non-parametric fashion. The same procedure during testing as described in parametric HMMs. Benefits from the automatic state inference with HDPHMM, the auto-regressive correlation of the anomaly data, and the effective variational inference techniques. The summary of diagnoses results of non-parametric HMMs are presented in Table 5.1. Those numbers in blue are denoted the optimal diagnoses accuracy of specific anomaly type across all the methods, respectively.

- *HDPHMM-Gauss-VB*: A classifier based on HDPHMM with Gaussian observation was trained independently for each anomaly class. The standard VB inference algorithm [55] was used to learn the HDPHMM model. Similiar to the method proposed in [57], instead of the blocked Gibbs sampling algorithm, we learn the model by variational Bayesian inference.

- *HDPHMM-Gauss-moVB*: A classifier based on HDPHMM with Gaussian observation was trained independently for each anomaly class. A memoized online variational inference algorithm (moVB) [56] based on scalable adaptation of state complexity is used to learn this HDPHMM model.

- *HDPHMM-AR-VB*: A classifier based on HDPHMM with first-order autoregressive Gaussian observation was trained independently for each anomaly class. The standard VB inference algorithm [55] was used to learn the HDPHMM model.

- *HDPHMM-AR-moVB*: Finally, our results were evaluated on the HDPHMM with first-order autoregressive Gaussian observation for each anomaly class. A memoized online variational inference algorithm (moVB) [56] based on scalable adaptation of state complexity was used to learn this HDPHMM model.

Given that non-parametric sHDP-VAR-HMM learns the complexity of the model from the data, it produces more accurate models as is reflected by the higher diagnoses accuracies shown in Table 5.2. The learned number of states for the different anomaly types is shown in Table 5.3. Note that for equivalent parametric HMMs, a tedious model needs to be computed for each class to optimize the state cardinality between types. The diagonal elements represent the number of points for which the predicted label is equal to the true label, while off-diagonal elements are those that are mislabeled by the classifier. It is evident from the confusion matrix that the diagnoses outperforms other baseline methods.

5.5.3 Discussion

The multiclass classifier that is flagged when an anomaly is detected, was also tested through the sHDP-VAR-HMM on seven anomalies and six baseline methods. Our evaluations showed that we could not only detect anomalies reliably (overall accuracy

Table 5.2 Confusion matrix for diagnoses results on kitting experiment dataset with seven anomalies by HDPHMM-AR-moVB method

	TC	HC	HCO	NO	OS	WC	FP
TC	1.00	0.00	0.00	0.00	0.00	0.00	0.00
HC	0.00	1.00	0.00	0.00	0.00	0.00	0.00
HCO	0.00	0.00	0.95	0.00	0.05	0.00	0.00
NO	0.00	0.07	0.00	0.93	0.00	0.00	0.00
OS	0.00	0.00	0.00	0.00	1.00	0.00	0.00
WC	0.00	0.00	0.00	0.00	0.00	1.00	0.00
FP	0.00	0.00	0.00	0.00	0.00	0.00	1.00

Table 5.3 The number of hidden states being active for different anomaly classes using HDPHMM-AR-moVB method. An active state is one in which at least one observation is assigned to this state

TC	HC	HCO	NO	OS	WC	FP
4	7	5	4	8	6	3

of 91.0%, as presented in Chap. 4) but also classify them precisely (overall accuracy of 97.1%).

With regards to anomaly diagnoses, anomalies usually occur from various sources such as sensing errors, control failures, or environmental changes. If a robot identifies the type, it may be beneficial to prevent or at least recover from the anomalous situation. In our proposed diagnoses method, we trained the sHDP-VAR-HMM model for each anomaly type separately. To address this, Natraj Raman had proposed a signal discriminative HDP-HMM for all classes albeit with an extra level that is class specific. Thus, an interesting future work direction consists in investigating a multilevel sHDP-VAR-HMM for all classes for multiclass classification and the extensions of the observation model by using higher order autoregressive Gaussian models.

Development of robot introspection is expected to have a direct impact on a large variety of practical applications, such as that can be used to estimate the log-likelihood of failure and prevent the failure during robot manipulation task. Also, the improvement of safety in human-robot collaborative environment by assessing the quality of learned internal model for each skill, which can speed up the anomaly recovery and/or repair process by providing the detailed skill identification and anomaly monitoring.

There are a number of limitations in our work. Currently we do not explicitly reduce the influence of outliers occasionally found in the derived log-likelihood values for specific hidden states. The outliers have a significant impact on the calculation of the threshold and our approach needs to address them specifically to avoid their impact. Additionally, we note the fact that our kitting experiment was not conducted under real factory conditions or in a real household daily task. Thus the verifiability of the work in real-world settings is unclear and further testing in real-factory conditions is necessary. The kitting experiment provides a proof-of-concept and the authors would like to extend their work to actual scenarios through corporate partners.

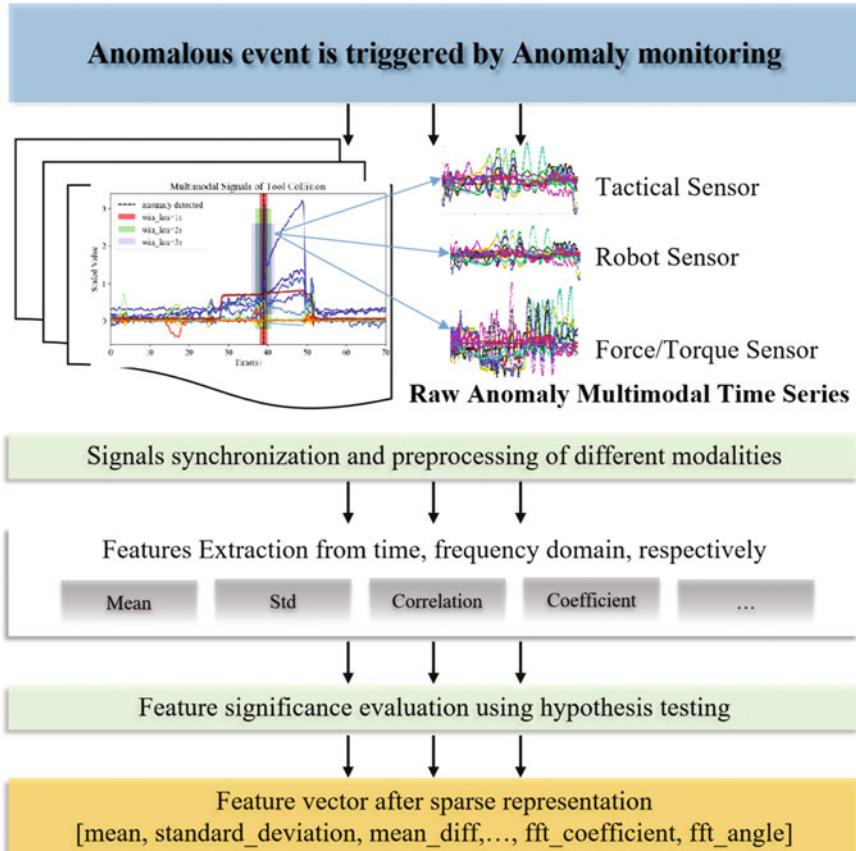


Fig. 5.7 The implementation procedure of sparse representation

5.6 Anomaly Classifier Based on Feature Extraction

The procedure of the sparse representation system for multimodal time-series is shown in Fig. 5.7.

5.6.1 Anomaly Sample Collection

- **Sensory Preprocessing**

The original multimodal sensory data includes a 6 DoF force and torque signals from F/T sensor, a 6 DoF Cartesian velocity signal from a manipulator's end-effector, 56 DoF tactical signals from a left and a right tactile sensor panels. To produce

more consistent data content, we do not directly concatenate individual data sources, instead temporal synchronization is done across modalities at 10HZ. Also, different preprocessing techniques are done for specific modalities.

Wrench modality: Takes a force and torque time-series vector sequence and for each element represents the magnitude of each dimension ($f_x, f_y, f_z, t_x, t_y, t_z$). Empirically, we wish that anomalies HC and TC can effectively flag external perturbations caused from different directions. We also consider the norm of force n_f and torque n_t respectively:

$$n_f = \sqrt{f_x^2 + f_y^2 + f_z^2}, \quad n_t = \sqrt{t_x^2 + t_y^2 + t_z^2} \quad (5.3)$$

Velocity modality: We measure the linear (l_x, l_y, l_z) and angular (a_x, a_y, a_z) Cartesian velocity (the endpoint state of a Baxter right hand), which are reported with respect to the base frame of the robot. As with the wrench source, we also consider the norm of the linear velocity n_l and angular velocity n_a respectively.

$$n_l = \sqrt{l_x^2 + l_y^2 + l_z^2}, \quad n_a = \sqrt{a_x^2 + a_y^2 + a_z^2} \quad (5.4)$$

Taxel modality: Due to the high dimensionality of tactical sensor, we do not process all the original signals as the model input. After empirical testing, the standard deviation of each tactile sensors s_l, s_r were selected as the preferred features and defined as:

$$s_l = \sqrt{\frac{1}{28} \sum_{i=1}^{28} (l_i - \mu_l)^2}, \quad s_r = \sqrt{\frac{1}{28} \sum_{i=1}^{28} (r_i - \mu_r)^2} \quad (5.5)$$

where the $\mu_l = \frac{1}{28} \sum_{i=1}^{28} l_i$ and $\mu_r = \frac{1}{28} \sum_{i=1}^{28} r_i$ is the mean of each tactical panel, respectively.

The above three preprocessing operations capture unexpected changes in the original signals since the signals are slightly different from the robot variable progress in normal executions. We then concatenate all the features to represent robot executions both in nominal and anomalous cases. Note that raw concatenated features can easily result in high False Positive Rates (FPR) during execution due to significant task execution variability across the same task in our experiments. For instance, the F/T signals vary different across differing objects of different weights; or similarly, human collisions which occur from different directions and magnitudes.

A standardization method is used to scale original signals ξ_o by its mean and standard deviation according to:

$$\xi(*) = \frac{\xi_o(*) - \text{mean}(\xi_o(*))}{\text{std}(\xi_o(*))}, \quad * \in \{f_x, f_y, \dots, s_l, s_r\}. \quad (5.6)$$

Finally, our eighteen dimensional multimodal signal(wrench, velocity, and tactical modalities) is represented as:

$$\begin{aligned} y_t = [\xi(f_x), \xi(f_y), \xi(f_z), \xi(t_x), \xi(t_y), \xi(t_z), \\ \xi(n_f), \xi(n_t), \xi(l_x), \xi(l_y), \xi(l_z), \xi(a_x), \\ \xi(a_y), \xi(a_z), \xi(n_l), \xi(n_a), \xi(s_l), \xi(s_r)]. \end{aligned} \quad (5.7)$$

5.6.2 Anomaly Features Extraction

In order to keep temporal consistency, anomaly signals are considered for a given *window_size* which is fixed around events flagged as anomalous. For instance, if a tool collision is detected as shown in Fig. 5.8, then *window_size* can evaluate how the diagnoses reactivity performs in our system. Generally, the *window_size* will equal a power of two (the a preferred size when including the Fast Fourier Transformation (FFT) between the time and frequency domain).

As described above, the sparse representation is applied for each extracted sample window. According to our previous work on anomaly diagnoses [20], the features are extracted in both the time domain and frequency domain. For anomaly diagnoses, we empirically consider the independent features and the corrective features along a specific modality signal as in Eq. 5.7. Here, the original multimodal signals with 18 DoFs and 12 feature types are considered in both the time and frequency domains, where the final feature vector is of length 558.

A: Independent features

Here, we calculate the features along a specific modality signal $\xi\{\ast\} = (x_1, x_2, \dots, x_n)$ with n data points

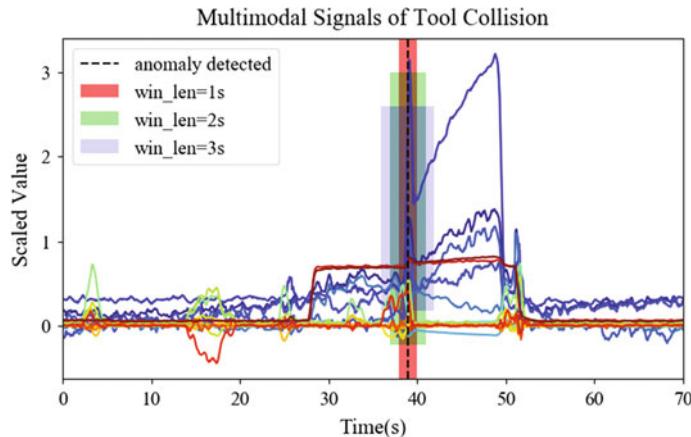


Fig. 5.8 Our robot introspection system for extracting the anomaly data when anomaly detected. For instance, the data of tool collision is represented with a given *window_size* = ± 2 s in red background

1. mean

$$\mu = \frac{1}{n} \left(\sum_{i=1}^n x_i \right) \quad (5.8)$$

2. standard_deviatiion

$$\sigma = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - \mu)^2} \quad (5.9)$$

3. mean_diff: calculate the mean over the differences between subsequent values

$$\mu_{diff} = \frac{1}{n} \sum_{i=1}^{n-1} (x_{i+1} - x_i) \quad (5.10)$$

4. mean_abs_diff: calculate the mean over the absolute differences between subsequent values

$$\mu_{abs_diff} = \frac{1}{n} \sum_{i=1}^{n-1} |x_{i+1} - x_i| \quad (5.11)$$

5. abs_energy: calculate the absolute energy, that is the sum over the squared values

$$e_{abs} = \sum_{i=1}^n x_i^2 \quad (5.12)$$

B: Correlative features

1. autocorrelation: calculate the autocorrelation of the specified lags $k \in \{1, 2, 3, 4\}$ given the mean μ and variance σ^2 , respectively.

$$\mathcal{R}_k = \frac{1}{(n-k)\sigma^2} \sum_{t=1}^{n-k} (x_t - \mu)(x_{t+k} - \mu) \quad (5.13)$$

2. mean_autocorrelation: calculate the mean of the autocorrelation which taken over all possible lags $l \in \{1, \dots, n\}$

$$\mu(\mathcal{R}) = \frac{1}{n-1} \sum_{l=1}^n \mathcal{R}_l \quad (5.14)$$

3. std_autocorrelation: calculate the standard deviation of the autocorrelation which taken over all possible lags $l \in \{1, \dots, n\}$

$$\sigma(\mathcal{R}) = \sqrt{\frac{1}{n-1} \sum_{l=1}^n (\mathcal{R}_l - \mu(\mathcal{R}))^2} \quad (5.15)$$

4. ar_coefficient: get the first 5 coefficients by fitting the unconditional maximum likelihood of an autoregressive model $\mathcal{AR}(p)$ with order p . In case of our implementation, the order is a fixed $p = 10$. The $\mathcal{AR}(5)$ model is defined as

$$x_t = \varphi_0 + \sum_{i=1}^{p=10} \varphi_i x_{t-i} + \varepsilon_t \quad (5.16)$$

where ε_t is drawn from a Gaussian white noise with a mean of zero and unit variance.

5. partial_autocorrelation: calculate the value of partial autocorrelation function of given lag $k \in \{1, 2, 3, 4, 5\}$, denoted $\alpha(k)$, is the autocorrelation between x_t and x_{t+k} with the linear dependence of x_t on x_{t+1} through x_{t+k-1} removed. As such, the function is defined as

$$\begin{aligned} Cov(x_t, x_{t-k}) &= Cov(x_t, x_{t-k} | x_{t-1}, \dots, x_{t-k+1}) \\ var(x_t) &= Var(x_t | x_{t-1}, \dots, x_{t-k+1}) \\ var(x_{t-k}) &= Var(x_{t-k} | x_{t-1}, \dots, x_{t-k+1}) \\ \alpha(k) &= \frac{Cov(x_t, x_{t-k})}{\sqrt{var(x_t)var(x_{t-k})}} \end{aligned} \quad (5.17)$$

C: Spectrum-based features

1. fft_coefficient: Get the first 5 Fourier coefficients of real part by FFT algorithm, respectively. In this implementation, we define the discrete Fourier transform function as

$$\begin{aligned} \mathcal{F} &= \sum_{m=0}^{n-1} a_m \exp\left\{-2\pi \frac{mk}{n}\right\}, \quad k = 0, \dots, n-1 \\ a_m &= \exp\{2\pi i f m \Delta t\} \end{aligned} \quad (5.18)$$

where, Δt is the sampling interval.

2. fft_angle: Calculate the angle of obtained complex value \mathcal{F} of the first 5 Fourier coefficients, respectively.

$$\theta = \tan^{-1} \frac{\mathcal{F}_{imag}}{\mathcal{F}_{real}} \quad (5.19)$$

5.6.3 Experiments and Results

- **Experimental Setup**

A kitting experiment consists of 5 basic nodes: Home, Pre-pick, Pick, Pre-place, and Place. The experiment is implemented in the following order with those nodes by 6 skills with the ROS-SMACH,² (Skill 1): Home → Pre-pick; (Skill 2): Pre-pick → Pick; (Skill 3): Pick → Pre-pick; (Skill 4): Pre-pick → Pre-place; (Skill 5): Pre-place → Place; (Skill 6): Place → Pre-place, as shown in Fig. 4.11.

The primary goal of the kitting task is designed to transport 6 different objects to a fixed container. The right arm of Baxter humanoid robot is used to pick objects and equipped with a 6 DoF Robotiq FT sensor, 2 Baxter-standard electric pinching fingers. Each finger is further equipped with a multimodal tactile sensor that a four by seven taxel matrix that yield absolute pressure values. The left hand camera is placed flexibly in a region that can capture objects with a resolution of 1280×800 at 1 fps (we optimize pose accuracy and lower computational complexity in the system). The use of the left hand camera facilitated calibration and object tracking accuracy. All code was run in ROS Indigo and Linux Ubuntu 14.04 on a mobile workstation with an Intel Xeon processor, 16GB RAM, and 8 cores.

When robot collaboratively works with human in a shared workspace, so many external disturbances are likely to occur. Those anomalies in Fig. 4.12 are considered in the Kitting experiment, which including the following 7 types: Tool Collision (TC) that may be derived from the visual error or the user accidentally collide with the object during robot moving to grasp it (see Fig. 4.12b); Human Collision (HC) is usually happened by a user to unintentionally collide with the robot arm in the human-robot collaboration environment(see Fig. 4.12a). We treat the human collision differently from whether the robot carrying object or not. Thus, Human Collision with Object (HCO) is assumed that the human collision while robot carrying object from the node *Pre – pick* to *Pre – place* (see Fig. 4.12d). The object have been knock down by the robot during grasping may induce the No object (NO) or missed-grasps(see Fig. 4.12f). Another common anomaly is Object Slip (OS) that the picked object may slip from the robot’s gripper if the grasping pose is not optimal or the robot moves at high speed. Finally, the False Positive (FP) is labeled when some unexpected disturbances may be detected by the anomaly detector for a variety of reasons, for instance, the system error, the object is placed at unreachable zone, without feasible inverse kinematic solution, and so on. In the rest of this paper, we intentionally achieve the spare representation of the seven types of anomalies while preserved sufficient diagnoses accuracy, respectively.

- **Results and Analysis**

The dataset contains a total of 108 samples from 137 experimental recordings of kitting task and the proportion for each anomaly are TC 15.7%, HC 16.7%, HCO 16.7%, NO 13.0%, OS 16.7%, WC 15.7%, FP 5.6%, respectively. For evaluating the

²<http://www.ros.org/SMACH>.

Table 5.4 Multiclass classifiers

Index	Classifier	Comment
1	BernoulliNB	Binarizing the dataset with the boolean threshold μ of each modality axis, which indicates that feature values below or equal to the threshold are replaced by 0, otherwise, by 1. The smoothing parameter $\alpha = 1.0$ and with a uniform prior on the class distribution
2	GaussianNB	Default settings
3	DecisionTree	A non-parametric supervised learning method used for diagnoses. A major advantage is that it does not require huge data preparation
4	RandomForest	Default settings
5	LinearSVC	Default settings
6	LogisticRegression	Default settings
7	SGDClassifier	Preprocessing procedure is added to convert the dataset with zero mean and unit variance
8	RidgeClassifier	Default setting
9	KnnDtw	n_neighbors=2 and max_warping_window=10

performance of the proposed sparse representation of multivariate time-series, we took the following 9 representative classifiers³ into consideration and the parameter settings are described respectively in Table 5.4.

As described above, the feature selection is a process where the most significant features in predicting the outcome are selected automatically. However, irrelevant features decrease the model's accuracy and increase computational cost. We therefore calculate the *p_value* of each extracted feature by using the hypothesis testing method in spscitefresh2016. That is, we perform a singular statistical test checking the hypotheses for each extracted feature f_1, f_2, \dots, f_n ,

$$\begin{aligned} H_0^i &= \{x_i \text{ is irrelevant for predicting class } y\}; \\ H_1^i &= \{x_i \text{ is relevant for predicting class } y\}; \end{aligned} \quad (5.20)$$

The result of hypothesis test in Eq. 5.20 is a *p_value*, which assess the probability that feature x_i is not relevant for predicting class y . As such, we define the score of feature by calculating the negative logarithmic value on the *p_value*. Large scores $-\log(p_{values})$ indicate features, which are relevant for predicting the target.

The performance of different classifiers is shown in Fig. 5.9, as you can see (reading left to right on the graph), the accuracy indicates to increase as the number of features are added, until a point beyond which there seems to be too few features for the classifier to make any reliable conclusions. Specifically, those features are extracted from a sorted feature vector in descending order by score values.

³<http://scikit-learn.org/>.

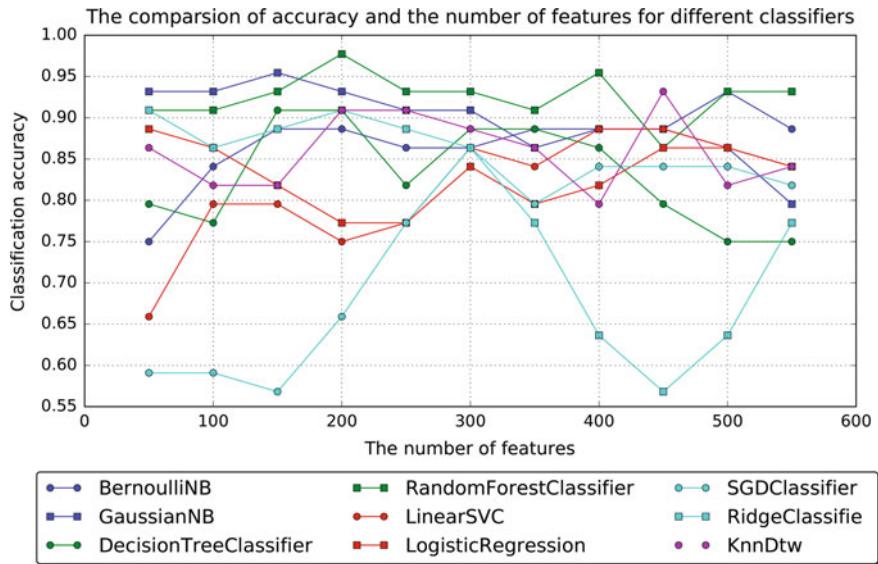


Fig. 5.9 The comparison between diagnoses accuracy and the number of feature on different classifiers

5.6.4 Discussion

This work implements sparsely represent the recorded multimodal time-series with relevant features as small as possible while preserving diagnoses accuracy. We propose the multivariate features are extracted in both time domain and frequency domain and not only consider the static statistical characteristics, but also including the correlation and interaction of each dimensional sensory signal. Results indicate that the data set can be significantly reduced up to 72.2% ~ 86.1% (the number of features is 100 and 200, respectively) of the raw data while keep the average diagnoses accuracy at about 85% with small data preparation. Future work should therefore include analyzing the trade-off between the value *window_size* and the diagnoses accuracy. So as to represent the recorded multimodal time-series with relevant features as small as possible while preserved diagnoses accuracy.

5.7 Summary

In this chapter, anomaly diagnose methods using the Bayesian nonparametric hidden Markov models and sparse representation by statistical feature extraction when anomaly triggered are introduced. Specifically, the detail procedure for anomaly sample definition, the supervised learning dataset collection as well as the data augmen-

tation of insufficient samples are also presented. The proposed methods is verified with a multi-step human-robot collaboration objects kitting task on Baxter robot, the performance and results are presented of each method respectively. That is, a multi-target classifier based on the nonparametric Bayesian model is proposed, Which using the sHDP-VAR-HMM model to model the anomaly sample data of various anomaly types. For the task of Baxter robot performing kitting experiment, resulting in the diagnoses accuracy of a total of 7 types of anomaly events in is 97.1%. Additionally, for the sparse representation for anomaly diagnoses, we extract a total of 31 features of time and frequency domains of the anomaly sample. Then, the extracted features importance analysis using the hypothesis testing method, after that the filtered features are verified on 9 common out-of-the-box supervised learning methods for multi-class diagnoses in *sklearn*.

References

1. Krogh A, Larsson BÈ, Von Heijne G, et al. Predicting transmembrane protein topology with a hidden Markov model: application to complete genomes[J]. *J Mol Biol.* 2001;305(3):567–80.
2. Och FJ, Ney H. The alignment template approach to statistical machine translation[J]. *Comput Linguist.* 2004;30(4):417–49.
3. Och FJ, Ney H. A comparison of alignment models for statistical machine translation[C]. In: Proceedings of the 18th conference on computational linguistics-volume 2. Association for Computational Linguistics;2000. p. 1086–90.
4. Karlof C, Wagner D. Hidden Markov model cryptanalysis[C]. In: International workshop on cryptographic hardware and embedded systems. Springer;2003. p. 17–34.
5. Green PJ, Noad R, Smart NP. Further hidden Markov model cryptanalysis[C]. In: International workshop on cryptographic hardware and embedded systems. Springer;2005. p. 61–74.
6. Bhar R, Hamori S. Hidden Markov models: applications to financial economics[M]. Springer Science Business Media;2004.
7. Rogemar S Mamon, Robert J Elliott. Hidden markov models in finance[M]. Springer;2007.
8. 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.
9. Veenendaal A, Daly E, Jones E, et al. Sensor tracked points and HMM based classifier for human action recognition[J]. *Comput Sci Emerg Res J.* 2016;5:4–8.
10. Ren L, Patrick A, Efros AA et al. A data-driven approach to quantifying natural human motion[C]. ACM Trans Graph (TOG). ACM;2005;24:1090–97.
11. Hovland GE, McCarragher BJ. Hidden Markov models as a process monitor in robotic assembly[J]. *Int J Robot Res* 1998;17(2):153–68.
12. Alshraideh H, Runger G. Process monitoring using hidden Markov models[J]. *Quality Reliab Eng Int.* 2014;30(8):1379–87.
13. Chuk TY, Chan AB, Shimojo S et al. Mind reading: discovering individual preferences from eye movements using switching hidden Markov models[C]. In: Proceedings of the 38th annual conference of the cognitive science society, CogSci 2016.
14. Manogaran G, Vijayakumar V, Varatharajan R et al. Machine learning based big data processing framework for cancer diagnosis using hidden Markov model and GM clustering[J]. *Wirel Person Commun.* 2018;102(3):2099–16.
15. Geurts P. Pattern extraction for time series classification[C]. In: European conference on principles of data mining and knowledge discovery. Springer;2001. p. 115–27.

16. Hüskens M, Stagge P. Recurrent neural networks for time series classification[J]. Neurocomputing. 2003;50:223–35.
17. Graves A. Supervised sequence labelling[M]. Supervised sequence labelling with recurrent neural networks. Springer;2012. p. 5–13.
18. Orsenigo C, Vercellis C. Combining discrete SVM and fixed cardinality warping distances for multivariate time series classification[J]. Pattern Recognit. 2010;43(11):3787–94.
19. Lee Y-H, Wei C-P, Cheng T-H, et al. Nearest-neighbor-based approach to time-series classification[J]. Decis Support Syst. 2012;53(1):207–17.
20. M-L Zhang, Z-H Zhou. A k-nearest neighbor based algorithm for multi-label classification[C]. In: 2005 IEEE international conference on granular computing. IEEE;2005. vol. 2, p. 718–21.
21. Seto S, Zhang W, Zhou Y. Multivariate time series classification using dynamic time warping template selection for human activity recognition[J]. 2015. [arXiv:1512.06747](https://arxiv.org/abs/1512.06747).
22. Berndt DJ, Clifford J. Using dynamic time warping to find patterns in time series[C]. In: KDD workshop. Seattle, WA, 1994, vol. 10, p. 359–70.
23. Baydogan MG, Runger G, Tuv E. A bag-of-features framework to classify time series[J]. IEEE Trans Pattern Anal Mach Intell. 2013;35(11):2796–802.
24. Han M, Liu X. Feature selection techniques with class separability for multivariate time series[J]. Neurocomputing. 2013;110:29–34.
25. Fulcher BD, Jones NS. Highly comparative feature-based time-series classification[J]. IEEE Trans Knowl Data Eng. 2014;26(12):3026–37.
26. Nanopoulos A, Alcock R, Manolopoulos Y. Feature-based classification of time-series data[J]. Int J Comput Res. 2001;10(3):49–61.
27. Utomo C, Li X, Wang S. Classification based on compressive multivariate time series[C]. In: Australasian Database Conference. Springer;2016. p. 204–14.
28. Jaakkola T, Diekhans M, Haussler D. A discriminative framework for detecting remote protein homologies[J]. J Comput Biol. 2000;7(1–2):95–114.
29. Raman N, Maybank SJ. Action classification using a discriminative multilevel HDP-HMM[J]. Neurocomputing. 2015;154:149–61.
30. Di Lello E. Bayesian Time-Series Models: Expert Knowledge-Driven Inference and Learning for Engineering Applications[J]. 2015.
31. Karim F, Majumdar S, Darabi H, et al. LSTM fully convolutional networks for time series classification[J]. IEEE Access. 2018;6:1662–9.
32. Karim F, Majumdar S, Darabi H, et al. Multivariate LSTM-FCNs for Time Series Classification[J] 2018. [arXiv:1801.04503](https://arxiv.org/abs/1801.04503).
33. Che Z, Purushotham S, Cho K, et al. Recurrent neural networks for multivariate time series with missing values[J]. Sci Rep. 2018;8(1):6085.
34. Yi Zheng, Qi Liu, Enhong Chen, et al. Time series classification using multi-channels deep convolutional neural networks[C]//International Conference on Web-Age Information Management. Springer, 2014: 298–310.
35. Fu Y, editor. Human activity recognition and prediction. Switzerland: Springer; 2016.
36. Geurts P. Pattern extraction for time series classification. In: European conference on principles of data mining and knowledge discovery, 3. Berlin, Heidelberg: Springer; 2001. p. 115–27.
37. Pavlovic V, Frey BJ, Huang TS. Time-series classification using mixed-state dynamic Bayesian networks. In: Proceedings. 1999 IEEE computer society conference on computer vision and pattern recognition (Cat. No PR00149). IEEE;1999 Jun 23. vol. 2, p. 609–15.
38. 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.
39. Park D, Kim H, Hoshi Y, Erickson Z, Kapusta A, Kemp CC. 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 Sep 24. p. 5406–13.
40. Pettersson O. Execution monitoring in robotics: a survey. Robot Auton Syst. 2005;53(2):73–88.
41. Bjäreland M. Model-based execution monitoring. InLinköping Studies in Science and Technology, Dissertation No 688. 2001. <http://www.ida.liu.se/labs/kplab/people/marbj>.

42. Muradore R, Fiorini P. A PLS-based statistical approach for fault detection and isolation of robotic manipulators. *IEEE Trans Ind Electron.* 2011;59(8):3167–75.
43. Orsenigo C, Vercellis C. Combining discrete SVM and fixed cardinality warping distances for multivariate time series classification. *Pattern Recognit.* 2010;43(11):3787–94.
44. Seto S, Zhang W, Zhou Y. Multivariate time series classification using dynamic time warping template selection for human activity recognition. In: 2015 IEEE symposium series on computational intelligence. IEEE;2015 Dec 7. p. 1399–06.
45. Karim F, Majumdar S, Darabi H, Harford S. Multivariate lstm-fcns for time series classification. *Neural Netw.* 2019;1(116):237–45.
46. Pei W, Dibeklioğlu H, Tax DM, van der Maaten L. Multivariate time-series classification using the hidden-unit logistic model. *IEEE Trans Neural Netw Learn Syst.* 2017;29(4):920–31.
47. Baydogan MG, Runger G. Learning a symbolic representation for multivariate time series classification. *Data Min Knowl Discov.* 2015;29(2):400–22.
48. Park D, Erickson Z, Bhattacharjee T et al. Multimodal execution monitoring for anomaly detection during robot manipulation[C]. In: 2016 IEEE international conference on robotics and automation (ICRA). IEEE;2016. p. 407–14.
49. 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 Ind Electro,* Early access. 2020. <https://doi.org/10.1109/TIE.2020.2970635>.
50. Park D, Kim H, Hoshi Y et al. A multimodal execution monitor with anomaly classification for robot-assisted feeding[C]. In: 2017 IEEE/RSJ international conference on intelligent robots and systems (IROS). IEEE;2017. p. 5406–13.
51. Bunte K, Haase S, Biehl M, Villmann T. Stochastic neighbor embedding (SNE) for dimension reduction and visualization using arbitrary divergences. *Neurocomputing.* 2012;1(90):23–45.
52. Arthur D, Vassilvitskii S. k-means++: The advantages of careful seeding. Stanford; 2006 Jun 7.
53. Welch LR. Hidden Markov models and the Baum-Welch algorithm. *IEEE Inf Theory Soc Newslett.* 2003;53(4):10–3.
54. McGrory CA, Titterington DM. Variational Bayesian analysis for hidden Markov models. *Aust N Z J Stat.* 2009;51(2):227–44.
55. Hughes MC, Sudderth EB. Bnpy: Reliable and scalable variational inference for bayesian nonparametric models. In: Proceedings of the NIPS probabilistic programming workshop, Montreal, QC, Canada. 2014. p. 8–13.
56. Hughes MC, Sudderth E. Memoized online variational inference for Dirichlet process mixture models. In: Advances in neural information processing systems. 2013. p. 1133–41.
57. Di Lello E, Klotzbücher M, De Laet T, Bruyninckx H. 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. IEEE; 2013 Nov 3. p. 5827–33.

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.



Chapter 6

Learning Policy for Robot Anomaly Recovery Based on Robot Introspection



Abstract In this chapter, the anomaly recovery would be acted when both of the anomaly monitoring and diagnoses are analysed, which aim to respond the external disturbances from the environmental changes or human intervention in the increasingly human-robot scenarios. To effectively evaluate the exploration, we summarize the anomalies in a robot system include only two catalogues: accidental anomalies and persistent anomalies. In particular, we first diagnose the anomaly as accidental one at the beginning such that the reverse execution is called. If and only if robot reverse many times (not less than twice) and still couldn't avoid or eliminate the anomaly, the human interaction is called. Our proposed system would synchronously record the multimodal sensory signals during the process of human-assisted demonstration. That is, a new movement primitive is learned once an exploring demonstration acquired. Then, we heuristically generate a set of synthetic demonstrations for augmenting the learning by appending a multivariate Gaussian noise distribution with mean equal to zeros and covariance equal to ones. Such that the corresponding introspective capabilities are learned and updated when another human demonstration is acquired. Consequently, incrementally learning the introspective movement primitives with few human corrective demonstrations when an unseen anomaly occurs. It is essential that, although there are only two different exploring strategies when anomaly occurs, numerous exploring behaviors can be generated according to different anomaly types and movement behaviors under various circumstances.

6.1 Introduction

In recent years, with the widespread promotion of the cooperative robot and human beings' inclusive operation, the robot's abnormal recovery behavior cannot be performed by the traditional robot's own motion planning algorithm, and human expectations for robot motion should be brought into play. The recovery strategy will reflect the human-centered concept of human-machine collaboration. For this reason, the robot anomaly recovery problems considered in this article mainly refer to the recoveries learned from humans by themselves in the case of robots that can recovery anomalies (not considering abnormal situations that cannot be recovered

autonomously such as power interruptions or system crashes). The strategy makes the robot's motion change from an abnormal state to a normal state. In other words, how to learn from humans to obtain corresponding recovery strategies for different abnormal types of robots is the key issue of this chapter, and it is required that the proposed recovery strategies have certain expansibility and generalization.

6.2 Related Work

To address the unpredictable abnormalities in robot operation tasks in an unstructured dynamic environment, Gutierrez et al. [1] proposed a way of expanding the original operation graph to recovery unforeseen abnormal events. The overall idea of the method is to use finite states machine learns an initial robot operation diagram for limited human demonstration movements, then re-plans the robot's movement when there is an abnormality in the actual operation application, and adds this recovered movement to the operation diagram, as the operation diagram continues Perfect, so that we can always find a feasible recovery behavior for the abnormal event. Paster et al. [2, 3] proposed the establishment of a specific library of motion primitives for the robot's operation tasks, and taught the different ways to complete the task through human experience. After combining the methods of motion primitive selection, the abnormal premise was detected. Next, the next primitive corresponding from the motion primitive library is completed [4, 5], thereby completing the behavior of abnormal recovery. Different from the robot's own method of motion conversion and selection, Salazar-Gomez et al. [6] proposed to implement robot's abnormal recovery by introducing human observation and control [7]. This method is a kind of human-in-the-loop (Human-in-the-loop) human-robot interaction. This has the advantage that humans have a more intuitive understanding of the robot's movements. When abnormal or error occurs, they can think about feasible solutions in time. Such recovery behavior is more Vulnerable to researchers and more conducive to applications in the human-machine environment. Niekum et al. [8] also described the operation tasks of the robot through the finite state machine. In the case of detecting an abnormal event, humans assist the robot to complete the current sub-task through kinematic teaching.

With associating this recovery behavior with the type of anomaly and update it to the original operation diagram, so that the robot can adopt a specific recovery behavior in the face of different anomalies. Mao et al. [9] proposed a method of human-assisted learning to implement robot abnormal recovery, where the robot's movement from the initial motion primitive will collide with the environment. At this time, the human manual Stop the robot and start the teach and teach mode to re-teach the recovered motion primitives for the robot to complete the abnormal recovery. Karlsson et al. [10, 11] proposed an online method of dynamically modifying robot motion primitives to achieve anomaly recovery. This method also pre-parameterized the robot's motion by DMP. In the event of an abnormality, it is indicated by human motion Teach ways to learn recovery behavior. In [12], the authors proposed the abort and retry

behavior in grasping by minimizing the overhead time as soon as the task is likely to fail. As an alternative, Johan S. Laursen et al. introduced a system for automatically reversed in robot assembly operations using reverse execution, from which forward execution can be resumed [13]. Similarly, a recovery policy by modelling reversible execution of robotic assembly is proposed in [14]. Another perspectives for anomaly recovery, Arne Muxfeldt et al. proposed a novel approach for recovering from errors during automated assembly in typical mating operations [15–17], which is based on automated error detection with respect to a predefined process model such that a recovery strategy can be selected from an optimized repository.

In summary, the use of human-machine collaboration to demonstrate abnormal recovery behavior has been favored by a large number of researchers, because in daily life or human-machine collaboration environment, the abnormal recovery behavior of robots cannot be achieved by the traditional robot itself. The motion planning algorithm should be carried out, and human expectations for robot motion should be brought into play. The human-assisted robot abnormality recovery strategy will further reflect the “human-centered” human-machine collaboration concept. In view of this, based on anomaly monitoring and classification, this paper proposes two types of recovery strategies for accidental and persistent anomalies, and different strategies will have different recovery behaviors under different types of anomalies.

6.3 Statement of Robot Anomaly Recovery

Specifically, the task segmentation, anomaly monitoring and diagnosis can be implemented through the generative methods such as the Bayesian non-parametric time series model that described in Chaps. 3–5 respectively. With the help of the task generalization and introspective capabilities for each movement primitives, two task exploration policies are proposed for responding the anomalies: reverse execution or human interaction. These policies can be learned from the context of manipulation tasks in an increasing manner. An overview of our SPAI framework can be described by a graph \mathcal{G} composed of N_b behaviors (or sub-tasks) \mathcal{B} , which are interconnected by edges \mathcal{E} such that $\mathcal{G} : \{\mathcal{B}, \mathcal{E}\}$, $\mathcal{B} = \{\mathcal{B}_1, \dots, \mathcal{B}_{N_b}\}$, $\mathcal{E}_{s,t} = \{(s, t) : s, t \in \mathcal{B}\}$. Behaviors in turn consist of nodes \mathcal{N} and edges \mathcal{E} , such that: $\mathcal{B} = \{\mathcal{N}, \mathcal{E}\}$. Nodes can be understood as phases of a manipulation task. In our work, we prefer to name them milestones $\mathcal{N}_i = (1, \dots, N_i)$, as they indicate particularly important achievements in a task. With task exploration behaviors, we may introduce a exploring nodes \mathcal{N}_{ij} in-between milestones creating a new branch to the subsequent milestone. It is also possible to introduce further exploring nodes \mathcal{N}_{ijk} on already existing branches. The full set of nodes in a task then is described as the union of milestone nodes with all branched nodes $\mathcal{N} = \{\mathcal{N}_i \cup \mathcal{N}_{ij} \cup \dots \cup \mathcal{N}_{ij\dots q}\}$. Node Transitions \mathcal{T} , behave as with behavior transitions \mathcal{E} , so $\mathcal{T}_{s,t} = \{(s, t) : s, t \in \mathcal{N}\}$. In our framework, a node is composed of modules. Modules are dependent processes that play a key role in the execution of a manipulation phase and could include demonstration collection, segmentation, movement learning, introspection, task representation and

exploration, vision, natural language processing, navigation, as well as higher level agents. In this chapter, we restrict a node to segmentation, generalization, introspection and exploration modules.

6.4 Learning and Application of Anomaly Recovery Policy

6.4.1 *Learning for Unstructured Demonstrations*

As we know, a sentence in language is made up of words according to grammatical rules, and a word is made up of letters according to word formation. Correspondingly, a complex and multi-step robot manipulation task can be represented as a sentence, in which a coupled robot movement primitive is equivalent to a word, and the robot movement primitive of each degree of freedom (DoF) of can be considered as letter such that the robot manipulation task can be represented with a set of movement primitives. To this end, how to effectively learn a multi-functional movement primitive is a critical problem for intelligent robot performing complex tasks in unstructured environment. If so, that is an interesting idea for generalizing the task representation with respect to the external adjustment as well as improving the diversity and adaptability of tasks. As a consequence, the task is consist of sequential movement primitives, which not only take the kinesthetic variables into consideration but also equip with introspective capacities such as the identification of movement, anomaly detection and anomaly diagnoses. We now introduce a Bayesian nonparametric model for robust learning the underlying dynamics from unstructured demonstrations.

• Demonstrations

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.

• Learning from Demonstration

Learning from demonstration (LfD) has been extensively used to program the robots, which aiming to provide a natural way to transfer human skills to robot. LfD is

proposed by simply teaching a robot how to perform a task as human-like, in which users can demonstrate new tasks as needed without any prior knowledge about the robot. However, LfD often yields weak interpretation about the environment as well as the task always is single step and lab-level such that lacks of robust generalization capabilities in dynamic scenarios, especially for those complex, multi-step tasks, e.g. human-robot collaborative kitting task designed in this paper.

For this reason, we present the powerful algorithms that draw from recent advances in Bayesian nonparametric HMMs for automatically segment and leverage repeated dynamics at multiple levels of abstraction in unstructured demonstrations. The discovery of repeated dynamics provides discriminating insights for understanding the task invariants, high-level description from scratch, and appropriate features for the task. In this paper, these discoveries could be concatenated using a finite state representation of the task, and consisted of movement primitives that are flexible and reusable. Thus, this implementation provides robust generalization and transfers in complex, multi-step robotic tasks. We now introduce a flowchart which integrates three major modules that critical for implementing the complex task representation from unstructured demonstrations.

• Segmentation

The aim of the segmentation is exploring the hidden state representation of the demonstrations, in which a specific hidden state usually denotes the clustering observations would be sampled from a statistical model, e.g. multivariate Gaussian distribution. 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 segmentation approach is a hidden state determined with Bayesian nonparametric hidden Markov model, which leads to tackle the generalization problem in a more natural way to meet the need of real-world applications.

As we discussed above, the HDP-VAR-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 $z_t = k \in \{1, 2, 3, \dots, K\}$. We denote Θ_s to represent all the parameters of the trained HDP-VAR-HMM from nominal demonstrations, including the hyper-parameters for learning the posterior and the parameters for the observation model definition.

$$z = \arg \max_{1, \dots, K} p(z_t | y_t, \Theta_s). \quad (6.1)$$

Here, the z would be a variable value, that is, we can concatenate the derived hidden state sequences and group the starting and goal observations for each sequence, respectively.

Assume that we record N multimodal demonstrations via kinesthetic fashion and jointly model them using the HDP-VAR-HMM, result in N hidden state sequences $\mathcal{Z} = \{\mathcal{Z}_1, \mathcal{Z}_2, \dots, \mathcal{Z}_N, \}$, where the element of \mathcal{Z}_i , is integer. Here, the problem is to segment the demonstrations into time intervals associated with individual movement

primitives by state-specific way. After the complex task segmentation, we achieve the task representation as presented in Chap. 3 using Finite State Machine (FSM) and Dynamical Movement Primitives (DMP).

6.5 Reverse Execution Policy for Accidental Anomalies

Reverse execution allows the robot retry the current movement or several movements that independent with the current state for resolving the accidental faults, such as human collision and mis-grasp. The key question is how far back must we revert in the task? To address this, we are currently evaluating different methodologies to learn reverse policies. Ideally, the performing critic is able to include all task-relevant information: the state of the robot, the state of the environment, the affordances of a task, and the relationship between these elements. However, this is not trivial, we are studying whether we could integrate decision making processes from multiple users. It's also difficult to measure the motivation of users to select a given node. Human users might have key awareness of the task that may render them select nodes for different reasons. Expected Utility is an area of study in Risk management [18]. An utility probabilistic model is designed to reflect a users intrinsic motivations, not limited to utility, risk propensity, and the influence in learning within a single decision episode, or across episodes. We expect to present preliminary results at the workshop.

An intuitive illustration of reverse execution is presented in Fig. 6.1. We assume the robot performs the current behavior $\mathcal{B}_i = \{\mathcal{N}_s^i, \mathcal{N}_g^i\}$ and an accidental fault \mathcal{F}_x is detected, where \mathcal{N}_s^i and \mathcal{N}_g^i indicate the starting and goal node, respectively. Subsequently, a new exploring node named r for responding this fault is autonomously appended to original task graph \mathcal{G} as illustrated in Sect. 6.3, that is

$$r : \mathcal{T}_{\mathcal{B}_i, \mathcal{B}_*} | \mathcal{F}_x, \mathcal{B}_i \quad (6.2)$$

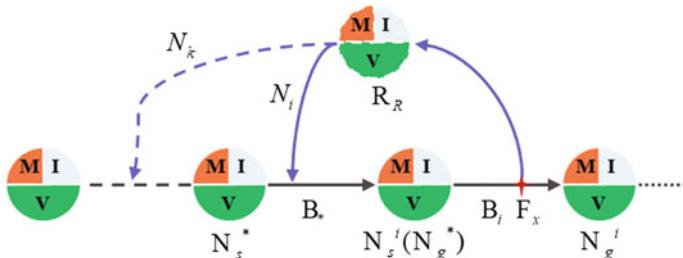


Fig. 6.1 Exploring the task via reverse execution when a fault event occurs

As formulated in Eq.(6.4), the symbol $\mathcal{B}_* = \{\mathcal{N}_s^*, \mathcal{N}_g^*\}$ denotes a optimal way that consists of one or more selected movement primitives for retrying under current fault type and behavior situation. So, the critical problem is how to parameterize the transition probability $p(\mathcal{T}_{\mathcal{B}_i, \mathcal{B}_*})$ given \mathcal{F}_x and \mathcal{B}_i when more than one ways to explore. To address this, a statistical distribution is introduced, which the instances are independent identically distributed and belong to a discrete distribution as well as the sum of the transient probabilities of all the samples after a fault occurrence is equal to 1. For these reasons, we define the $p(\mathcal{T}_{\mathcal{B}_i, \mathcal{B}_*})$ is modelled with a multinomial distribution that the random variables is the respective frequency counted based on human intention when a fault occurs. For example, $\mathbf{N}_{\mathcal{F}_x} = (N_1, N_2, \dots, N_K)$ is a frequency distribution of a random variables vector, where K represents the total number of movement primitives that from beginning to current movement \mathcal{B}_i with fault \mathcal{F}_x (including the \mathcal{B}_i) and $N_i, i \in \{1, 2, \dots, K\}$ denotes how many times of movement \mathcal{B}_i is successfully executed. Therefore, the probability mass function of transition $p(\mathcal{T}_{\mathcal{B}_i, \mathcal{B}_*})$ is formulated as a multinomial distribution that model a total $N = \sum_{i=1}^{i=K} N_i$ reverse executions, that is,

$$p(\mathbf{N}|N, \theta) = \frac{N!}{N_1! \dots N_K!} \prod_{i=1}^K \theta_i^{N_i} \quad (6.3)$$

where, θ_i indicates the probability of movement primitive \mathcal{B}_i is selected, which subject to $\theta_i \in [0, 1]$ and $\sum_{i=1}^K \theta_i = 1$. Therefore, we use the multinomial distribution not only to intuitively depict the expectation of human intention on the recovery behavior when an abnormal occurrence is detected in human robot interaction scenarios, but also to provide an indirect way to express the intuitive understanding of human for expecting the motion of the robot end-effector, the related manipulation objects as well as the complex relationship between “human-robot-environment”.

6.6 Human Interaction Policy for Persistent Anomalies

Human interaction allows the robot explore the current movement by human-assist demonstration for resolving the persistent faults that can't be restored by reverse execution, such as tool collision and wall collision. The key question is how to fast capture and learn from the interactive demonstration base on the defined task representation. This exploring policy is activated when the system fail more than two times when reverse executions is carried out for resolving the fault \mathcal{F}_x . The human interaction policy is an another exploring way that through human assisted demonstration when robot encounter a persistent fault. It's essential that synchronously capture the human kinesthetic demonstration that not only the kinematic variables but also the relative coordinate frame (updating the task structure defined in Sect.6.3 by understating the transformation relationship between demonstration and original movement) at each time step. After that, the movement is formulated by the DMP techniques intro-

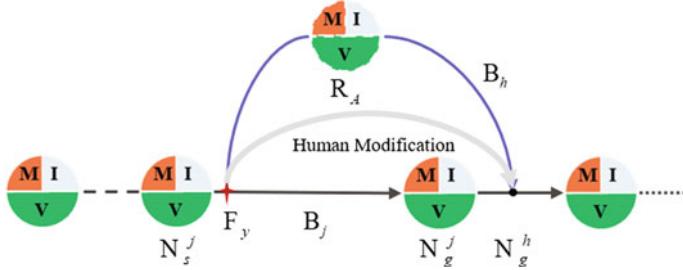


Fig. 6.2 Exploring the task via human interaction when a fault event occurs

duced in Chap. 3. In particular, the human interaction policy is not only limited to the originally designed movements, but also can be applied to the exploring movement (reverse execution or human interaction). Theoretically, our system can handle any kinds of failure situation with those aforementioned two exploring policies such that potentially achieve the long-term autonomy during robot manipulation task. Additionally, experiential verification indicates that another problem arises, there are faults also exist in reproducing the human interaction. To address this problem, we introduce a data augmentation method (without detail statement in this paper) for training a new exploring movement and introspective model (described in Chap. 4) when the same persistent fault is continuously encountered more than three times in the same (multimodal signals are recorded at each time). Consequently, we can achieve the critical behavior for incrementally addressing the faults by “exploration of exploration” and “anomaly monitoring and diagnose in exploration”.

Without loss of generality, an intuitive illustration of human interaction is presented in Fig. 6.2. We assume the robot performs the current behavior $\mathcal{B}_j = \{\mathcal{N}_s^j, \mathcal{N}_g^j\}$ and an accidental fault \mathcal{F}_y is detected, where \mathcal{N}_s^j and \mathcal{N}_g^j indicate the starting and goal node, respectively. Subsequently, a new exploring node named A for responding this fault is autonomously appended to original task graph \mathcal{G} as illustrated in Sect. 6.3, that is

$$A : \mathcal{T}_{\mathcal{B}_j, \mathcal{B}_h} | \mathcal{F}_y, \mathcal{B}_j \quad (6.4)$$

where, \mathcal{B}_h indicate the human interactive demonstration $\mathcal{B}_h = \{\mathcal{N}_s^h, \mathcal{N}_g^h\}$, that have the same starting node \mathcal{N}_s^j with movement \mathcal{B}_j and a new goal node \mathcal{N}_g^h is derived from demonstration. Equipping this exploring movement \mathcal{B}_h by formulating it using the dynamical movement primitive. Additionally, we define the goal pose (end-effector or joint variables) of demonstration as P that only task kinematic variables into consideration for task exploration. The P can be adapted by the transformation relationship between \mathcal{N}_s^j and \mathcal{N}_g^h , then \mathcal{N}_g^h derived by

$$\mathcal{N}_g^h = P \mathcal{T}_{\mathcal{N}_s^j, \mathcal{N}_g^j} \quad (6.5)$$

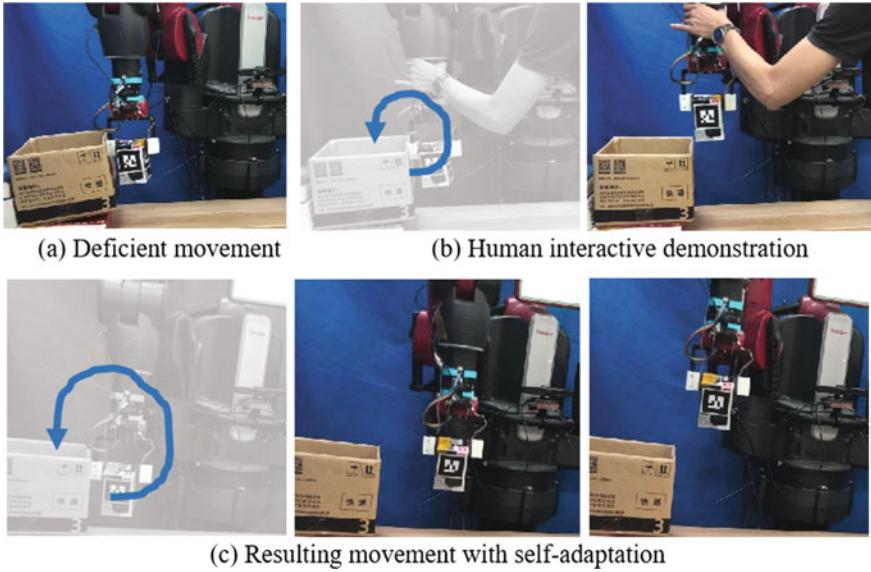


Fig. 6.3 Illustrates the human interactive demonstration in our kitting experiment when the robot collides with the packaging box during transporting an object

As Fig. 6.3 shows a human robot collaborative task is designed in latter experimental verification, where a Baxter robot encounters a persistent fault (wall collision) during transporting object from human-over. In this situation, the robot likely to encounter a fault along the deficient movement (as shown in Fig. 6.3a), and the fault can't be eliminated by reverse execution because of the fixed obstacle (box) on the right hand side of robot. An modulated trajectory should be introduced for updating the original task representation, as shown in Fig. 6.3b a human interactive demonstration for exploring the failure task. Subsequently, a transformation in Eq. (6.5) is derived by learning from interactive demonstration, which can be explored in a new scenario when encountering a same fault, as shown in Fig. 6.3c.

6.7 Experiments and Results

6.7.1 Platform Setup

To evaluate the proposed method for incremental learning the introspective movement primitives. We designed a HRC task for picking and placing object into a

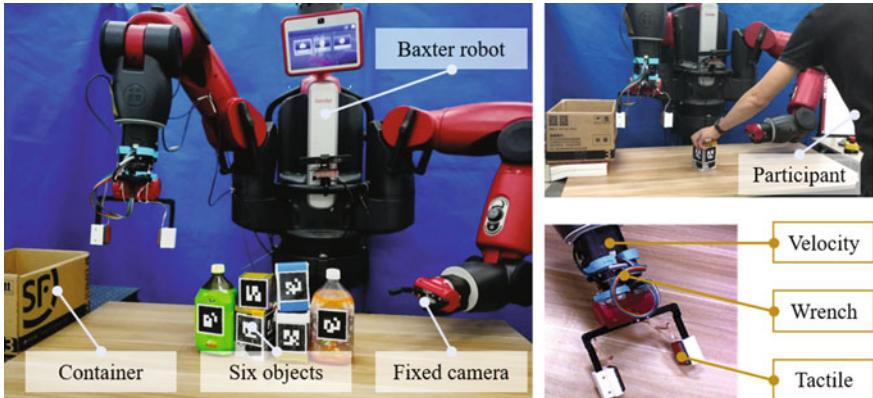


Fig. 6.4 An autonomous kitting experiment by Baxter robot: The robot is designed to transport 6 marked objects with variable weights and shapes to a container, where the external anomalies may arise from accidental collisions (human-robot, robot-world, robot/object-world), mis-grasps, object slips So as to identify those unexpected anomalies, the robot arm is integrated with multimodal sensors (shown in bottom-right), including internal joint encoders, F/T sensor, tactile sensor

container using Baxter robot and integrated the ROS Indigo¹ and Moveit² as the middle-wares in our system. Specifically, a human co-worker is tasked to place a set of 6 objects marked with Alvar tags³ on the robot’s reachable region (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 once the first object is placed on the table, the robot’s left arm camera identifies the object and the robot’s right arm picks and places it in a container located to the right of the robot, as shown in Fig. 6.4. Multiple sensors were installed for effectively sensing the unstructured environment and potential faults in such a kitting experiment. Here, the right arm of Baxter robot is equipped with a 6 degrees of freedom (DoF) Robotiq F/T sensor and 2 Baxter-standard electric pinching fingers, where each finger is further equipped with a multimodal tactile sensor composed of a 4×7 taxel matrix that yields absolute pressure values. In addition, Baxter’s left hand camera is placed flexibly in a region that can capture objects in the collection bin with a resolution of 1280×800 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. After there aforementioned integration, the robot picks each object and transports it towards the container, after which, the robot appropriately places each of the six objects in different parts of the container, several snapshots are visualized in Fig. 6.5.

In consideration of the redundant features would aggravate computational efficiency and increase false-positive rate (fault occurs even when robot’s movement is

¹<http://wiki.ros.org/indigo>.

²<https://moveit.ros.org/>.

³http://wiki.ros.org/ar_track_alvar.

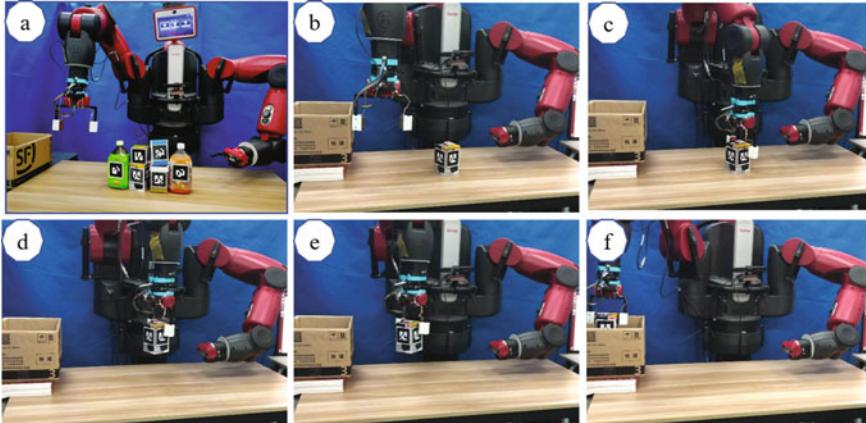


Fig. 6.5 The Snapshots of human-robot kitting experiment with a Baxter robot. We subjectively assume this task consists of 5 individual movements, including **a** Home → Pre-pick; **b** Pre-pick → Pick; **c** Pick → Pre-pick; **d** Pre-pick → Pre-place; **e** Pre-place → Place; **f** Reset

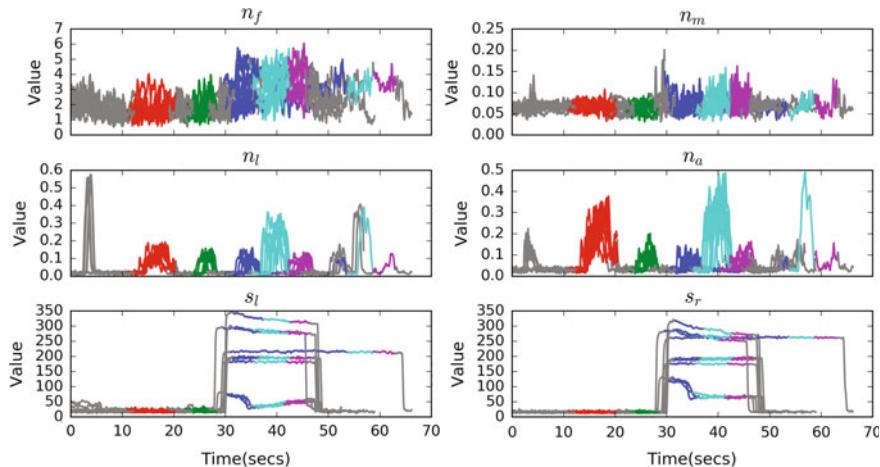


Fig. 6.6 Multimodal signals in kitting task, where the grey represents the invalid robot movements including the adjustment, orientation alignment, reset, etc, other colors are empirically indicated an potential movement primitive along with obvious signals fluctuations

normal), we perform empirical features extraction on the original observation vector to improve identification performance. Specifically, we compute the norm of both the force n_f and the moment n_m as features in wrench modality, take the norm of both the Cartesian linear n_l and angular n_a velocities in velocity modality, and consider the standard deviation for each tactile sensor s_l and s_r . Therefore, our feature vector y_t of length 6 and formulated as $y_t = [n_f, n_m, n_l, n_a, s_l, s_r]$, evolving extracted features of ten nominal executions are illustrated in Fig. 6.6.

6.7.2 Parameter Settings for Anomaly Monitoring and Diagnose

We need both qualitative and quantitative analysis of the proposed method. To evaluate the whole performance of IMPs in unstructured scenarios with unexpected faults that including both of the accidental and persistent causes. We organized 5 participants as collaborator (one expert user who confidently know this implementation and other four novice users) in our designed kitting experiment. Novice users first learned from the expert to induce fault during robot executions, which would aggravate the external uncertainty and increase the modeling difficulties. During data collection, each participant performed 1 nominal and 6 executions that at least one fault event by placing the set of 6 household objects in a one-by-one fashion. Consequently, totally perform 30 nominal executions and 180 failure executions. We induce the fault manually for each movement primitive, where including but no limited to the Human Collision (HC), Object Slip (OS), Tool Collision (TC), Wall Collision (WC), etc.

We first evaluate the complex task segmentation from twenty whole kinesthetic demonstrations using Bayesian nonparametric methods. As shown in Fig. 6.7, illustrating the complex task segmentation by learning the underlying dynamics using HDP-VAR-HMM, where each row indicates an independent demonstration and different color represents a specific movement primitive, which no need to guarantee each demonstration have the same segmentation order. After then, we concentrate the ordered hidden state sequence and group them by hidden state transition pair t Pair, where the total number of pair is computed using the permutation combination algorithm, i.e. t Pairs = $C_5^2 A_2^2 = 20$, where K is the dimension of hidden space. Thus, the frequency among the possible transition of the learned five movement primitives is illustrated in Fig. 6.8, where the kitting experiment is definitely begin with movement 2 (clustered by the hidden state 2) and then the successor should be movement 1, subsequent movement is 4 for most cases or movement 3 is the second-choice, and so on. Until now, we can effective learning the complex task representation for a set of nominal unstructured kinesthetic demonstration in a manipulation graph way. With this representation, we can evaluate the following the anomaly monitoring, diagnose as well as task exploration, respectively.

According to Chaps. 4 and 5, our proposed movement primitive equipped with two introspective capabilities: anomaly monitoring and diagnose, that necessary for endowing robot long-term autonomy and safer collaborative interaction in human-robot collaborative scenarios. Particularly, our anomaly monitoring and diagnose are implemented based on the Bayesian nonparametric models proposed in Chap. 2 that using the HDP-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 Λ_k as well as the four parameters v , Δ , V , M of the conjugate prior MNIW are assigned in advance. To guarantee the prior has a valid mean, the degrees-of-freedom variable is set as $v = d + 2$ and Δ is set by constraining the mean or expectation of the covariance matrix $\mathbb{E}[\Lambda_k^{-1}]$ under the Wishart prior in

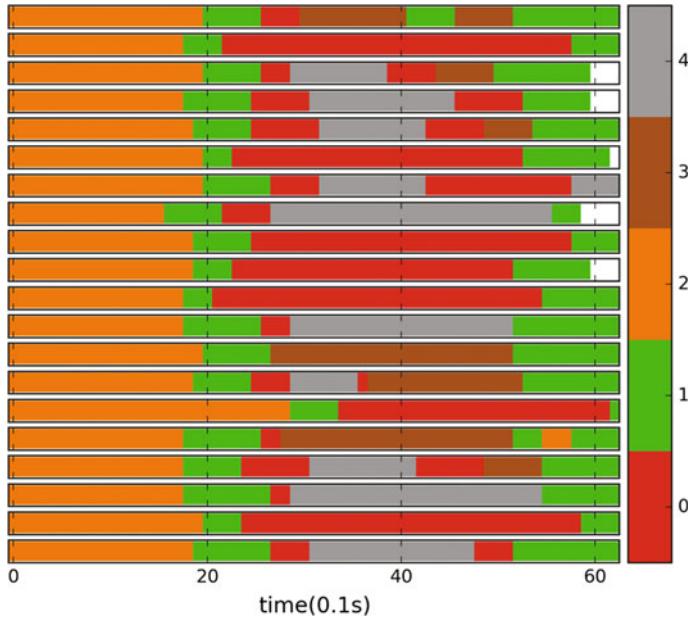


Fig. 6.7 Complex task segmentation by learning the underlying dynamics using HDP-VAR-HMM. Each row indicates an independent demonstration and different color represents a specific movement primitive, which no need to guarantee each demonstration have the same segmentation order

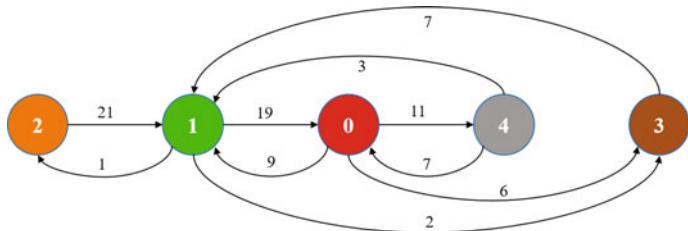


Fig. 6.8 Illustrates the transition frequency among the derived five movement primitives. The hidden state sequences is first taken to extract the movement primitives from a set of demonstrations. A uncompleted manipulation sequence is ordered with $2 \rightarrow 1 \rightarrow 0 \rightarrow 4 \rightarrow 3$ based on the derived sequences. The numbers on the transitions correspond to the transition frequency for weakly representing the task at the beginning, and subsequently strengthening by incremental learning the introspective capabilities and exploration from practical applications

Eq. (4.30).

$$\mathbb{E}[\Lambda_k^{-1}] = s_F \sum_{n=1}^N \sum_{t=1}^{T_n} (y_t - \bar{y})(y_t - \bar{y})^T. \quad (6.6)$$

Assume that we record N sequential data for each skill and the length of sequence $n \in N$ is T_n . Thus, we can easily define the parameter Δ accordingly as

$$\Delta = (v - d - 1)\mathbb{E}[\Lambda_k^{-1}]. \quad (6.7)$$

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 co-variances. 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 Chap. 2 for details).

$$\begin{aligned} M &= \mathbf{0}_d, \\ V &= 1.0 * \mathbf{I}_d. \end{aligned} \quad (6.8)$$

where $\mathbf{0}_d$ and \mathbf{I}_d are the matrix of all zeros and the identity matrix, respectively, each of size $d \times d$. For the concentration parameters, a $Gamma(a, b)$ prior is set on HDP concentration parameters α . A $Beta(c, d)$ prior is set on the self-transition parameter μ and the degree of self-transition bias κ is set to 50. We choose a weekly informative setting and choose:

$$a = 0.5, b = 5.0, c = 1.0, d = 5.0 \quad (6.9)$$

where, the initial transition proportion parameter is defined as $\mu \sim Beta(1, 10)$ and 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 fault diagnose.

The anomaly monitoring is implemented by comparing the cumulative likelihood \mathcal{L} of observed observations, where the fault threshold is calculated from a set of $\mathcal{L}_i, i \in \{1, 2, \dots, 20\}$ nominal demonstrations, and formulated with the expected cumulative likelihood $\mu(\mathcal{L})$ minus the standard deviation $\sigma(\mathcal{L})$ that multiply by a constant value c . Additionally, the fault diagnose is activated when fault detected, which mainly implemented by comparing the sum of log-likelihood of a failure sample in a supervised fashion. Particularly, we use the K-Folders Cross Validation



Fig. 6.9 Visualizing the robot trajectories of reverse execution when accidental faults are detected. As shown, the five normal IMPs is derived by learning from unstructured demonstrations that successfully performing human-robot collaborative kitting task and the dark-blue trajectories are explored with the maximum probability in our defined reverse policy once the fault detected

defined in *sklearn*⁴ (here, $K = 3$) for model selection, in which the objective is the accuracy of anomaly monitoring with a fixed constant value $c = 3$.

To achieve the incremental learning introspective movement primitive from unstructured demonstrations, another critical capability would be the task exploration under the unseen and unexpected situation, especially when robot encounter a failure event. There are two independent exploring policies: reverse execution and human interaction, are proposed for responding the external accidental and persistent faults, respectively. We test the whole performance in two scenarios for evaluating how to combine the anomaly monitoring, fault diagnose, and exploration in practical applications along with the quantitative performance. As illustrated in Fig. 6.9, a set of collective 3D moving trajectories is presented for evaluating the exploring reverse execution when robot encounter accidental faults. Where the kitting task is first represented by five introspective movement primitives, and then the accidental faults (marked in yellow dots) randomly happened during robot execution, the robot immediately perform the reverse execution (dark-blue in color) according the frequency distribution shown in Table 6.1.

Additionally, As illustrated in Fig. 6.10, a set of collective 3D moving trajectories is presented for evaluating the exploring human interaction when robot encounter an wall collision that is a persistent fault. To evaluate the overall performance after human one-shot interaction, we get a 80.95% success rate when the wall collision induced by repeating the experiment 30 times.

⁴<https://scikit-learn.org>.

Table 6.1 The frequency distribution of reverse execution policy in kitting experiment, where we record the data from five participants in total of 30 times of each accidental fault. Additionally, the \cup is an union of robot performing $Pre - pick \rightarrow Pick$ and $Pick \rightarrow Pre - pick$

Fault diagnose results	Fault occurs	Goal IMP	Frequency
Human collision	1	1	25
	2	2	30
	3	3	25
	4	4	25
Tool collision	2	1	2
	3	3	20
Object slip	2	2	5
	3	1	25

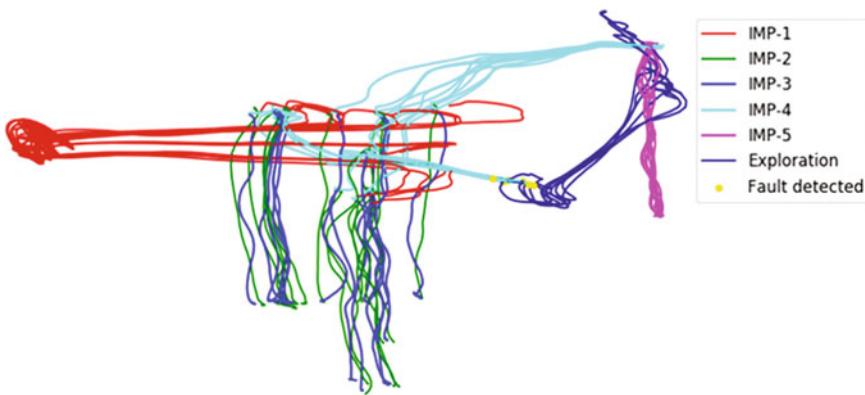


Fig. 6.10 Visualizing the robot trajectories after human interaction when persistent faults are detected. As shown, the five normal IMPs is derived by learning from unstructured demonstrations that successfully performing human-robot collaborative kitting task and the dark-blue trajectories are incrementally explored by updated the demonstrated IMP parameters (including the start and goal pose, shape) once the fault detected

6.7.3 Discussion

In our previous work, we focus on the robot introspective capabilities in robotics, which only take the kinematic variables into consideration such that the task is restricted to be applied in human-robot collaborative scenarios that generalization (including motion, introspection, and decision-making, etc.) as a desirable characteristic. To address this problem, robot movement primitive augmented with introspective capacities *IMPs* is investigated in this paper, which by associating the generalization, anomaly monitoring, fault diagnoses, and task exploration during robot manipulation task. We mainly introduce the IMPs can be acquired by assessing the quality of multimodal sensory data of unstructured demonstrations using a nonpara-

metric Bayesian model, named HDP-VAR-HMM, and IMPs can incremental learning the exploring policy using multimodal distribution and human one-shot modification when robot encounter fault. Particularly, reverse execution and human interaction are two independent policies for task exploration, which proposed to respond the external accidental and persistent fault, respectively. Experimental evaluation on a human-robot collaborative packaging task with a Rethink Baxter robot, results indicate that our proposed method can effectively increase robustness towards perturbations and adaptive exploration during human-robot collaborative manipulation task. We need to emphasize that our method presents a solution for endowing robot with introspection in sense-plan-act control methodology for robot manipulation task.

Recently, we are working on extending IMPs to be more human-like manipulation that including the visual and audial information for human robot collaborative electronic assembly task. We are also investigating the use of variational recurrent autoencoder neural network to facilitate our proposed framework for more complex scenarios.

6.8 Summary

In this chapter, we present two policies to deal with accidental and persistent anomalies respectively: movement reverse execution and human interaction. Reverse execution allows the robot retry the current movement or several movements that independent with the current state for resolving the accidental faults by redo current or some previous movement primitives by updating the parameters of primitives (including shape, starting point and target point, etc.), and to use polynomial distribution to realize the modeling of a primitive after the occurrence of an anomaly occurrence. Human interaction allows the robot explore the current movement by human-assist demonstration for resolving the persistent faults that can't be restored by reverse execution, such as tool collision and wall collision, which mainly to learn the anomaly behavior from the human demonstration, and can realize the growth of the task representation of the recovery behavior. Importantly however we learned that the recovery ability of the system grows in difficulty with an increased number of adaptations as variations in sensory-motor signals increase as more recoveries are attempted.

The proposed recovery policies are verified on a Baxter robot performs kitting experiment tasks, results indicate that the proposed policies meet the extensibility and adaptability for improving the long-term autonomy, an integrated to the SPAIR framework. Consequently, this book provides a efficient theoretical framework and software system for the implementation of the longer-term autonomy and a safer environment for human-robot interaction scenarios. Ultimately the system presented in this book significantly extended the autonomy and resilience of the robot and has broad applicability to all manipulation domains that suffer from uncertainties in unstructured environments: making industrial and service robots prime candidates for this technology.

In this book, anomalies are and will continue to be a reality in robotics despite increasingly powerful motion-generation algorithms, to address them explicitly, we presented a tightly-integrated, graph-based online motion-generation, introspection, and incremental recovery system for manipulation tasks in loosely structured co-bot scenarios, which consist of the movement identification, task representation, anomaly monitoring, anomaly diagnoses, and anomaly recovery.

References

1. Gutierrez RA, Chu V, Thomaz AL, Niekum S. Incremental task model updates from demonstration.
2. Pastor P, Righetti L, Kalakrishnan M et al. Online movement adaptation based on previous sensor experiences. In: 2011 IEEE/RSJ international conference on intelligent robots and systems (IROS). IEEE;2011. p. 365–71.
3. Pastor P, Kalakrishnan M, Righetti L et al. Towards associative skill memories. In: 2012 12th IEEE-RAS international conference on humanoid robots (humanoids). IEEE;2012. p. 309–15.
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. 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.
6. Salazar-Gomez AF, DelPreto J, Gil S et al. Correcting robot mistakes in real time using eeg signals. In: 2017 IEEE international conference on robotics and automation (ICRA). IEEE;2017. p. 6570–77.
7. 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>.
8. Niekum S, Chitta S, Barto AG et al. Incremental semantically grounded learning from demonstration. Robotics: science and systems. Berlin, Germany:Springer;2013. p. 9.
9. Mao R, Baras JS, Yang Y et al. Co-active learning to adapt humanoid movement for manipulation. In: 2016 IEEE-RAS 16th international conference on humanoid robots (humanoids). IEEE;2016. p. 372–8.
10. Karlsson M, Robertsson A, Johansson R. Autonomous interpretation of demonstrations for modification of dynamical movement primitives. In: 2017 IEEE international conference on robotics and automation (ICRA). IEEE;2017. p. 316–21.
11. Karlsson M, Robertsson A, Johansson R. Convergence of dynamical movement primitives with temporal coupling. In: 2018 European control conference (ECC). IEEE;2018 Jun 12. p. 32–9.
12. Rodriguez A, Mason MT, Srinivasa SS, Bernstein M, Zirbel A. Abort and retry in grasping. In: 2011 IEEE/RSJ international conference on intelligent robots and systems. IEEE;2011 Sep 25. p. 1804–10.
13. Laursen JS, Schultz UP, Ellekilde LP. Automatic error recovery in robot assembly operations using reverse execution. In: 2015 IEEE/RSJ international conference on intelligent robots and systems (IROS). IEEE;2015 Sep 28. p. 1785–92.
14. Laursen JS, Ellekilde LP, Schultz UP. Modelling reversible execution of robotic assembly. Robotica. 2018;36(5):625–54.
15. Muxfeldt A, Steil J. Fusion of human demonstrations for automatic recovery during industrial assembly. In: 2018 IEEE 14th international conference on automation science and engineering (CASE). IEEE;2018 Aug 20. p. 1493–00.
16. Muxfeldt A, Steil JJ. Recovering from assembly errors by exploiting human demonstrations. Proc CIRP. 2018;1(72):63–8.

17. Arne M. Automatic error recovery during industrial assembly operations based on human demonstrations. PhD diss.: Technische Universität Carolo-Wilhelmina zu Braunschweig;2018.
18. National Research Council. Modeling human and organizational behavior: application to military simulations. National Academies Press;1998 Aug 31.
19. Wu H, Xu Z, Yan W, Su Q, Li S, Cheng T, Zhou X. Incremental learning introspective movement primitives from multimodal unstructured demonstrations. IEEE Access. 2019;15(7):159022–36.

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.

