Adaptive Shared Autonomy between Human and Robot to Assist Mobile Robot Teleoperation

zur Erlangung des akademischen Grades eines
Doktors der Ingenieurwissenschaften

von der KIT-Fakultät für Informatik
des Karlsruher Instituts für Technologie (KIT)

genehmigte

Dissertation

von

Ming Gao
aus Chengdu, China

Tag der mündlichen Prüfung:	14. Dezember 2017
Erster Gutachter:	Prof. Dr.-Ing. Johann Marius Zöllner
Zweiter Gutachter:	Prof. Dr.-Ing. habil. Björn Hein

Abstract

Mobile robot teleoperation has been widely employed when it is impractical or infeasible for humans to be present, yet still requires human judgment and decision-making skills. However, it is frustrating and stressful for human beings to merely simply teleoperate a robot without assistance due to time delay and absence of Situational Awareness (SA). On the other hand, fully autonomous robots, despite recent achievements, cannot yet execute tasks alone based on the current perception and control models. Consequently, both the human and the robot have to remain in the control loop, simultaneously contributing intelligence to the task executions, *i.e.* the human has to *share autonomy* with the robot during operation. But the challenge is how to best coordinate the two sources of intelligence from the human and the robot, to guarantee a safe and efficient task execution in remote.

Therefore, a novel strategy is proposed in this thesis. It models the user intention as a contextual task to complete an action primitive, and provides appropriate motion assistance to the human operator upon the task recognition. In this way, the robot copes intelligently with the on-going tasks based on the contextual information, relieves the workload of the human operator and improves the task performance. To implement this strategy and account for the uncertainties in acquiring and processing environment information and user input, *i.e.* the contextual information, a probabilistic shared autonomy framework is presented to infer the contextual task the human operator performs with uncertainty measurements, and appropriately assist the human operator with the task execution according to these measurements. Since the way the human operator performs a task is implicit, it is non-trivial to model the motion pattern of the task process manually, thus a set of data-driven approaches are adopted to derive the policies of various task executions from human demonstrations, to adapt to the needs of the human operator in an intuitive way over long time. The feasibility and scalability of the proposed framework and techniques have been extensively evaluated in a variety of experiments both in simulation and on real mobile robot. With the proposed approaches, the teleoperator can be actively and appropriately assisted by increasing the cognition capability and the autonomy flexibility of the robot.

Kurzfassung

Die Teleoperation vom mobilen Roboter wird in großem Umfang eingesetzt, wenn es für Mensch unpraktisch oder undurchführbar ist, anwesend zu sein, aber die Entscheidung von Mensch wird dennoch verlangt. Es ist für Mensch stressig und fehleranfällig wegen Zeitverzögerung und Abwesenheit des Situationsbewusstseins, ohne Unterstützung den Roboter zu steuern einerseits, andererseits kann der völlig autonome Roboter, trotz jüngsten Errungenschaften, noch keine Aufgabe basiert auf die aktuellen Modelle der Wahrnehmung und Steuerung unabhängig ausführen. Deswegen müssen beide der Mensch und der Roboter in der Regelschleife bleiben, um gleichzeitig Intelligenz zur Durchführung von Aufgaben beizutragen. Das bedeut, dass der Mensch die Autonomie mit dem Roboter während des Betriebes zusammenhaben sollte. Allerdings besteht die Herausforderung darin, die beiden Quellen der Intelligenz vom Mensch und dem Roboter am besten zu koordinieren, um eine sichere und effiziente Aufgabenausführung in der Fernbedienung zu gewährleisten.

Daher wird in dieser Arbeit eine neuartige Strategie vorgeschlagen. Sie modelliert die Benutzerabsicht als eine kontextuelle Aufgabe, um eine Aktionsprimitive zu vervollständigen, und stellt dem Bediener eine angemessene Bewegungshilfe bei der Erkennung der Aufgabe zur Verfügung. Auf diese Weise bewältigt der Roboter intelligent mit den laufenden Aufgaben auf der Grundlage der kontextuellen Informationen, entlastet die Arbeitsbelastung des Bedieners und verbessert die Aufgabenleistung. Um diese Strategie umzusetzen und die Unsicherheiten bei der Erfassung und Verarbeitung von Umgebungsinformationen und Benutzereingaben (*i.e.* der Kontextinformationen) zu berücksichtigen, wird ein probabilistischer Rahmen von Shared Autonomy eingeführt, um die kontextuelle Aufgabe mit Unsicherheitsmessungen zu erkennen, die der Bediener mit dem Roboter durchführt, und dem Bediener die angemesse Unterstützung der Aufgabenausführung nach diesen Messungen anzubieten. Da die Weise, wie der Bediener eine Aufgabe ausführt, implizit ist, ist es nicht trivial, das Bewegungsmuster der Aufgabenausführung manuell zu modellieren, so dass eine Reihe von der datengesteuerten Ansätzen verwendet wird, um das Muster der verschiedenen Aufgabenausführungen von menschlichen Demonstrationen abzuleiten, sich

an die Bedürfnisse des Bedieners in einer intuitiven Weise über lange Zeit anzupassen. Die Praxistauglichkeit und Skalierbarkeit der vorgeschlagenen Ansätze wird durch umfangreiche Experimente sowohl in der Simulation als auch auf dem realen Roboter demonstriert. Mit den vorgeschlagenen Ansätzen kann der Bediener aktiv und angemessen unterstützt werden, indem die Kognitionsfähigkeit und Autonomieflexibilität des Roboters zu erhöhen.

Table of Contents

Acronyms

ACR	Average Correspondence Rate
ANOVA	Analysis of Variance
ARD	Automatic Relevance Determination
ASA	Adaptive Shared Autonomy
ASAP	Adaptive Shared Autonomy Platform
BCI	Brain-Computer Interface
BIC	Bayesian Information Criterion
BN	Bayesian Network
BP-AR-HMM	Beta-Process Autoregressive Hidden Markov Model
BVs	Basis Vectors
CBR	Case-Based Reasoning
CDF	Cumulative Density Function
CR	Correspondence Rate
CTRM	Contextual Task Recognition Module
DBN	Dynamic Bayesian Network
DC	Doorway Crossing
DP	Dirichlet Process
DPGMM	Dirichlet Process Gaussian Mixture Model
DVR	Driver-Vehicle-Road
DWA	Dynamic Window Approach

EM	Expectation-Maximization
FANN	Fast Approximate Nearest Neighbours
FT	Free Travel
GMM	Gaussian Mixture Model
GMR	Gaussian Mixture Regression
GOA	General Obstacle Avoidance
GP	Gaussian Process
GUI	Graphical User Interface
HLA	High Level Autonomy
HMM	Hidden Markov Model
ICAD	Intra-cluster Average Dissimilarity
IMU	Inertial Measurement Unit
KL	Kullback-Leibler
KNN	K Nearest Neighbour
LfD	Learning from Demonstration
LHMM	Layered Hidden Markov Model
LLA	Low Level Autonomy
LR	Linear Regression
LRF	Laser Range Finder
LSD	Least Significant Difference
LWPR	Locally Weighted Projection Regression
MAP	Maximum a Posteriori
MCAM	Motion Command Arbitration Module

MICAD	Mean of Intra-cluster Average Dissimilarity
MLA	Middle Level Autonomy
MPC	Model Predictive Control
NE	Normalized Entropy
OA	Object Approach
OB	Object Bypass
OI	Object Inspection
OpenCV	Open Source Computer Vision library
PCL	Point Cloud Library
POMDP	Partially Observable Markov Decision Process
RANSAC	Random Sample Consensus Algorithm
RBF	Recursive Bayesian Filter
RF	Random Forest
RKM	Robot Kinematic Model
ROS	Robot Operating System
SA	Situational Awareness
SE	Square Exponential
SLAM	Simultaneous Localization and Mapping
SOGP	Sparse Online Gaussian Process
SVM	Support Vector Machine
TLX	Task Load Index
TSP	Tele-Sensor Programming
WF	Wall Following

Conventions

Matrix	big, bold: $\mathbf{X}, \mathbf{Y}, ...$
Scale	small, not bold: $x, y, ...$
Set	big, not bold: $X, Y, ...$
Vector	small, bold: $\mathbf{x}, \mathbf{y}, ...$, sometimes also expressed with an overhead arrow: $\overrightarrow{x}, \overrightarrow{y}, ...$

Symbols and Operators

exp(x)	Exponential function with exp(x)=e^x
$\lvert\mathbf{M}\rvert$	Determinant of Matrix **M**
$\mathcal{N}(x; \mu, \sigma^2)$	x is distributed normally with mean μ and variance σ^2
$P(x\lvert y)$	Conditional probability
$P(x)$	Probability
$\mathbb{R}^n$	Set of real vectors with n dimensions
$\mathbf{x}$	Vector **x** with components in column $(x_1 \cdots x_n)^T$
$\lvert X\rvert$	Number of elements in the set X
$\lvert x\rvert$	Absolute value of x
$\langle \mathbf{x}, \mathbf{y} \rangle$	Scalar product of **x** and **y**
x_t	x at time t

Chapter 1

Introduction

1.1 Motivation

Teleoperation is "to operate a vehicle or a system over a distance" [54], which usually comprises a robotic technology where a human operator controls a robot in remote. Teleoperation has been widely applied to situations where the *onsite* operation of the human is infeasible, yet requiring human judgment and decision-making skills. Typical examples are the handling of nuclear materials, control of small models, space and underwater exploration, and telepresense for social needs in domestic scenarios.

Due to time delay and absence of SA [54], it is grueling for the human operator to simply teleoperate the robot without any assistance. On the other hand, fully autonomous robot cannot yet carry out complex tasks alone based on the current perception and control models though a lot of efforts have been made [119]. Even when autonomous operation is possible in some situations, there are operators preferring to be in the loop to quickly obtain and interpret the information gathered by the robot [130]. Therefore, in spite of recent advances in robot autonomy, teleoperation remains an indispensable modality for mobile robot operation.

Therefore, a teleoperation system is supposed to provide the human operator with comprehensive information regarding both surrounding and state of the robot, including layout information of environments, location and status of physical components of the robot by improving the SA during teleoperation. Moreover, to relieve the workload of the human operator and efficiently improve the remote task performance, it is ideal during the teleoperation that the robot copes intelligently with the on-going tasks utilizing the contextual information, *i.e.* the user inputs and the environmental perceptions. In [75], it is demonstrated

that putting the human and the automaton in a collaborative control loop improves the task performance: the human is able to focus on a given task while the automaton takes care of routine tasks or disturbances that would otherwise distract the human away from the on-going task. The close coupling of user and robot in actually controlling the robot facilitates the user to perceive the robot actions while allowing the human operators to contribute their intentions simultaneously. Although this is a promising concept, the major challenge is how to best coordinate the two sources of intelligence from the human and the robot, to guarantee a safe and efficient task execution in remote. This lies at the core of the research focus in the field of *shared autonomy* [170].

In the following sections, the key terms involved in this thesis and the research problem this thesis addresses will be firstly explained. An overview of the proposed concept will then be presented.

1.2 Problem Statement

This thesis aims to realize a novel *Context-aware Adaptive Shared Autonomy* system to assist mobile robot teleoperation. To help understand such core concept, and propose the research problem this thesis addresses, the involved key terms are explained in this section.

Term **"Shared Autonomy"**. *Autonomy* refers to the capability of reasoning about and mapping inputs from the environment into a variety of actions [74]. Considering teleoperation as a control process, *Shared Autonomy* means that both the human operator and the robot exert actions to influence the control process during teleoperation by reasoning about the environment and making decisions to fulfil certain goals of the process.

Depending on the authority of the autonomy source during the process, there are three different perspectives when designing a shared autonomy system, *i.e.* human-centered [14], robot-centered [151], and equal-authority (*i.e.* multi-agent system [51]). Considering the highly unstructured nature of many teleoperation missions, which till now still require the human operator to always remain in the control loop to provide cognitive judgments and decision-makings with the highest authority, a *human-centered perspective* is taken to implement the shared autonomy system to assist mobile robot teleoperation in this thesis.

Term **"Adaptive"**. By taking a human-centered perspective to design shared autonomy system in this thesis, being *Adaptive* refers to that the robot is able to appropriately adapt its

behaviors to fulfil the needs of the human operator during task executions, in either short or long term. Therefore, the robot is supposed to assist the human operator by seamlessly switching the autonomy authority in a task-appropriate way during operation (*i.e.* being adaptive in short term). Moreover, the robot should not be limited on performing certain autonomous tasks during its life of deployment, rather even a non-professional user can customize such robot system in an intuitive way (*i.e.* being adaptive in long term). This holds true especially considering the trend that robotic systems are becoming ubiquitous in human society.

Term **"Context"**. In teleoperation, from the perspective of robot, *Context* refers to certain situation in remote, which is associated with a set of environment information, including its structure and semantic elements, together with user inputs to exert actions on the hardware of the robotic system. Such combined *contextual information* vaguely indicates the operational intentions of the human user, and which semantic element is *relevant* to the intention of the human operator during the task process, since it is assumed that the human operator makes action decisions based on both the reasoning of the environment information and the task requirements in mind.

Term **"Context-aware"**. With the explanation of the term *context* above, *Context-aware* means that the robot is able to capture and understand the situation from gathering the contextual information from its sensors, *i.e.* the environment perceptions and the user inputs, thus the the robot is able to further provide appropriate assistance to the intentional task the human operator performs, *i.e.* the robot is ideally supposed to assist "what the human operator intends to do", instead of "what the human operator is doing". This is critical for an efficient remote task execution when the robot assists the human operator in a shared autonomy system.

With the above defined terms, the research problem, where this thesis addresses, can be precisely formulated as below:

While both human and robot remain in the control loop, how can the robot appropriately assist the human operator with the teleoperation for efficient task executions, and adapt its behaviors to fulfil the needs of the human operator during task process, in both short and long terms, by capturing and understanding contextual situations in remote?

To answer this question, a set of approaches are developed in this thesis to interpret the contextual situation and provide adaptive context-aware assistance to promote the perfor-

mance of mobile robot teleoperation in unstructured indoor scenarios. Their concepts will be briefly presented in the following section.

1.3 Concept Overview

As inspired by the fact that human beings are capable of independently coordinating their joint actions with each other by inferring the others' action intentions [199], an appealing strategy is to recognize the user intention [1, 206, 24, 82]. With this strategy, the robot estimates the user intention with raw inputs and selects its action to optimize the desired task. This strategy is promising because it decouples the overall shared autonomy problem into two sub-problems: intention recognition and motion assistance [82]. With the prediction of the user intention during operation, this strategy can also improve the task performance even in the presence of time delays [21].

According to this strategy, this thesis focuses on recognizing the intentional task of the human operator with the contextual information, *i.e.* to infer the *contextual task* the human operator intends to perform, and provide assistance to the estimated contextual task. More specifically, the notion *task* refers to a *metric representation* of the user intention for a robot to complete an *action primitive*, *i.e.* a sequence of time-instant actuator movements lasting for a short period to achieve a target. Therefore, a contextual task refers to an action primitive associated with certain semantic component of the environment, *e.g.* to operate the robot to cross a nearby doorway, or control it to inspect a close object in the environment, *etc.* Modeling the user intention as contextual task to complete an action primitive is based on the following considerations.

A sequence of action primitives can constitute a long time complex task on the higher level. Consequently, recognizing action primitives performed by the human operator lays the foundation to infer a complex task over a long period the human operator executes, which is computationally more efficient than directly modeling and estimating a long time complex task on the high level. Additionally, when the human operator needs immediate assistance locally, it is more effective to use action primitives to represent such need than a complex task model across a long time. Last but not least, action primitives are more flexible and meaningful in describing the human behaviors than reactive actions on the lower level, which mostly pertains to either *obstacle avoidance* or *collision stopping*.

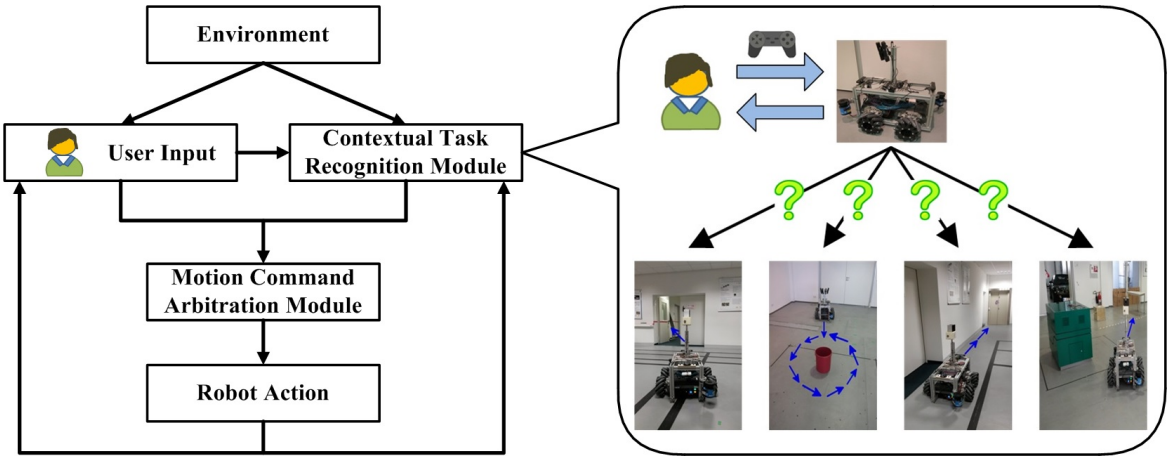


Fig. 1.1 Overall architecture of the proposed contextual-task aware shared autonomy system to assist mobile robot teleoperation. The concept of the CTRM is illustrated besides: a human operator teleoperates a mobile robot with joystick for task execution. The robot perceives the environment with its sensors, such as 2D laser scanners, and receives user inputs. The CTRM is responsible for recognizing the on-going contextual tasks the human operator performs with user inputs and environment perceptions. A contextual task refers to an *action primitive* the human operator executes via the robot, and is associated with a semantic component of the environment. Examples of it are Doorway Crossing, Object Inspection, Wall Following and Object Bypass, as shown at the bottom from left to right, and the arrows denote the movement directions of the robot.

The overall architecture of the shared autonomy system proposed in this thesis is depicted in Fig. 1.1. It is assumed that, the human operator teleoperates a mobile robot with a joystick-like device, and performs intentional tasks to complete action primitives based on the environment perceptions, yet the human operator does not communicate his/her task intentions explicitly to the robot during operation. Being abstracted from specific robotic hardware configurations, the proposed shared autonomy system receives user inputs and environment observations from the robotic sensors, and reasons about the contextual information to make inference regarding the task the human operator intends to execute, *i.e.* the Contextual Task Recognition Module (CTRM) shown in the architecture figure. This module is the cornerstone of the proposed shared autonomy system, and its concept is illustrated besides. Being embedded into certain robotic motion controller, the proposed system provides motion assistance to the human operator by blending the user input and the autonomous motion commands generated by the motion controller to safely execute the estimated task, *i.e.* the Motion Command Arbitration Module (MCAM) shown in Fig. 1.1.

Although this is an appealing strategy, inferring the contextual task the human operator performs is challenging. Firstly, the way the human operator executes a contextual task to

complete an action primitive is implicit, so it is non-trivial to manually derive a general and flexible computational model to describe various task processes. Secondly, there can exist multiple targets at the same time in real applications, each of which corresponds to a hypothesis to interpret the intentional task of the human operator, *e.g.* there could be multiple traversable doorways nearby in an indoor scenario where the robot is deployed, thus the approach to recognize contextual task should consider such situation, and be able to find out the correct target and associated task type[1] from multiple hypotheses. Last but not least, the observations of the robot, including environment perceptions and user inputs, which are obtained from the on-board sensors of the robot, are decorated by noise. Therefore, the contextual task estimation method is supposed to account for such uncertainties, when it uses the data to make inference. In this thesis, a set of innovative solutions are proposed to address these challenges, which will be detailed in the following section as the scientific contributions.

1.4 Contributions

The contributions of this thesis are as follows:

Because of the uncertainties in contextual task recognition, a probabilistic shared autonomy framework is proposed to assist mobile robot teleoperation. It infers the intentional task of the human operator with the contextual information in a probabilistic way, and provides corresponding motion assistance to the human operator based on the probability of the estimated task. In this way, the level of autonomy is seamlessly switched between the manual control (when the probability of the task recognition is low) and the autonomous control (when the probability of the task recognition is high) in a task-appropriate way during operation. To build the models for describing various task executions, a set of data-driven approaches are employed to learn how the human operator performs contextual tasks from demonstrations, in order to render the proposed framework adaptive to the needs of the human operator in the long run in an intuitive way. They constitute the major parts of the scientific contributions of this thesis, and are presented respectively in below.

[1] In this thesis, a *task* is an instance of the corresponding task type, while a *task type* represents a group of motions sharing similar patterns associated with certain semantic target. For example, “doorway crossing” is a task type, while “cross the doorway in front” is a task of this type. Therefore, a task type can have multiple instances due to the existence of multiple semantic targets in a scenario where the robot is deployed.

In the first approach, Gaussian Mixture Regression (GMR) models are employed to learn the motion patterns of various task types from human demonstrations in a batch way, where each task type to be estimated is described with a set of simple and salient *task features*. To incorporate the uncertainty of estimating multiple contextual tasks during operation, a Recursive Bayesian Filter (RBF) is adopted as the base framework, which is combined with the learned GMR models to stably and smoothly infer the most probable operational intention of the human operator among multiple candidates over time. Furthermore, the system is extended to learn the motion patterns online and incrementally from the demonstrations by proposing a fast online and incremental contextual task learning approach. With the incrementally incorporated demonstration dataset of each task type, a state-of-art Fast Approximate Nearest Neighbours (FANN) search algorithm is employed to retrieve a small training dataset closest to the current robot state, and a GMR model with a very few number of the mixture components is built online and combined with the RBF to estimate the likelihoods of the corresponding candidates.

In the second approach, contextual task recognition is formulated as classification problem. Envisioning a life-long active learning scenario, Gaussian Process (GP) regression model is used to learn from demonstrations and classify multiple contextual task types with the corresponding task features. Using Gaussian regression model results into closed-form solutions of the posterior predictive distribution and the model evidence respectively. By maximizing the closed-form model evidence, the hyper-parameters of the model can be optimized. In comparison with other state-of-art classifiers, such as Support Vector Machine (SVM), probably the most popular approach on this topic, GP Classifier possesses a superior introspective capability by expressing high uncertainty when facing previously unknown patterns due to the predictive mean and variance obtained from the posterior predictive distribution. This property is significant for mission critical decision making in the long run, such as motion assistance by task recognition during mobile robot teleoperation in this thesis. To keep the model sparse to limit the amount of storage and computation required, full GP is approximated with a state-of-art Sparse Online Gaussian Process (SOGP) algorithm, to maintain scalability to large datasets without compromising classification performance.

The last approach aims to relieve the workload of manually labeling the demonstration data into task types for robot to learn. To achieve this, an unsupervised contextual task learning and recognition approach is proposed, consisting of two phases. First, Dirichlet Process Gaussian Mixture Model (DPGMM) is applied to cluster the human motion patterns performing multiple contextual task types from unannotated demonstrations, where the number of possible motion modes is inferred from the data itself instead of being manually

specified *a priori* or determined through model selection. Post clustering, the SOGP classifier is employed to classify the learned motion patterns during operation, due to its superior introspective capability over other state-of-art classifiers and scalability to large datasets. By applying the proposed fast online GMR approach to the classified motion cluster with the recognized contextual information, it is able to predict local trajectory the human operator intends to follow in the short term to perform corresponding contextual task in a data-driven manner. This is critical, since it links the task recognition part and the motion assistance part of the proposed shared autonomy framework: the inferred trajectory can be used as the *reference model* by the state-of-art mobile robot motion controller to generate feasible motion commands, which are blended with the user inputs according to the estimation confidence, to assist the human operator to carry out the intentional tasks *actively* and *appropriately*.

The final contribution in this thesis is that the proposed framework is extensively evaluated and validated with a variety of experiments both in simulation and on real robot in various indoor scenarios. The results demonstrate the scalability and effectiveness of the proposed shared autonomy framework in learning, recognizing and utilizing the motion pattern knowledge of the human operator in data-driven manners from human demonstrations. A carefully controlled and repeatable user study is conducted and verifies with rigorous statistical analysis that, the proposed system outperforms the baseline control modes in safely and efficiently assisting the human operator to execute remote tasks both in terms of objective and subjective metrics. This result further confirms the merit of the contributions of this thesis.

1.5 Generalization to Other Applications

Although the subject of this thesis is to assist mobile robot teleoperation, the proposed framework generalizes to assist the teleoperation of a broad range of robot systems, *e.g.* telepresense robot, undersea vehicle and flying robot *etc.*, considering that it is abstracted from specific robotic hardware configurations, and is data-driven. Modification is only necessary in very specific parts of the architecture, such as in the interfaces between human and robot, or in the set of available robot actions.

The proposed approach can also be interesting for sharing autonomy between human and robot *locally*, such as the driving of a semi-autonomous vehicle on road or a truck for construction, and the operation of an intelligent wheelchair in indoor scenarios. In these

situations, different kinds of mobile robots are operated in either semi-structured (on-road vehicle) or unstructured environments. Under the help of the robot, the human operator can either avoid dangerous maneuvers resulting from fatigue over long driving or overcome handicaps, and complete intentional actions efficiently. Moreover, in such situations, there still exists *context* influencing the human operational actions, and the associated motion patterns can be extracted from the environment observations and user inputs, and utilized by the robot to provide corresponding assistance to the human user appropriately. This is the strategy explicitly adopted and implemented in the proposed framework of this thesis. In assisting the driving of a semi-autonomous vehicle, the use of a joystick versus a steering wheel and gas pedal is considered to be a difference in user interface, yet this is abstracted away from the rest of the framework. Furthermore, in construction equipment, the widely use of joystick simplifies the integration of the proposed framework with real robots.

1.6 Document Outline

This thesis is organized as follows: Chapter 2 presents an overview of related works and discusses the pros and cons of different approaches. Existing research gaps are indicated to motivate the proposed approaches. Chapter 3 mathematically formalizes the proposed probabilistic shared autonomy system for assisting mobile robot teleoperation through contextual task recognition, facilitating a better understanding of the detailed introductions of the proposed scientific contributions afterwards. Additionally, a reactive shared autonomy system based on Bayesian Network (BN) is also reported in this chapter. Even though it is not among the major contributions, its technical limitations in assisting mobile robot teleoperation reveal possible improvements and lead to an alternative, Bayesian contextual task recognition approach, including the methods for learning the motion patterns of various task executions from human demonstrations in both batch and incremental ways. This approach is described in Chapter 4. Chapter 5 introduces how to utilize the SOGP algorithm to learn and classify multiple contextual tasks from human demonstrations. Chapter 6 reports the technique details of the approach to achieve unsupervised learning and recognizing the contextual tasks from unlabeled demonstration data, and the algorithm to predict the local trajectory the human operator intends to follow in the short term to perform the corresponding task with the recognized contextual information. Finally, Chapter 7 presents the experiments which have been carried out to extensively evaluate the proposed approaches. The thesis is closed with a conclusion in Chapter 8, where the limitations are analyzed and the outlook for future research is indicated.

Chapter 2

State of the Art

This chapter introduces the state-of-art research achievements in the field of shared autonomy between human and robot for assisting human with collaborative task execution. Firstly, the involved themes of this thesis, including the concept of shared autonomy and its background in the context of human-robot interaction, are clarified based on the related works, and the criteria to categorize the studies conducted in this field are proposed, with the focus on how the human operator is assisted by the robot for task execution during interaction. Following that, an overview of approximately 80 related works from 2000 to 2016 is presented to provide a comprehensive view of the achievements in this field. Then several noteworthy studies are individually introduced and discussed in detail. Finally, the chapter is closed with a summary and a discussion on the existing research gaps in this field.

2.1 Themes Clarification

Shared Autonomy is a concept for system design, also named *mixed-initiative interaction* [7] or *adjustable autonomy* [181] in the literature. Regarding human-robot interaction, *Shared Autonomy* referes to distributing intelligence to man and machine for solving problems cooperatively during operation [20]. It also refers to the full or partial replacement of a function that has to be carried out by the robot [151], to combine the strengths of the robot (normally in data acquisition, navigation and planning) and those of the human (in cognition and reasoning), to perform a complex task cooperatively, which is hardly to achieve with either human or robot alone, such as teleoperation in unknown and hazardous environments.

The benefits of employing the concept in implementing human-robot system by making robots and humans work as partners haven been clearly pointed out in [56, 97, 152].

Depending on the level of automation, Sheridan proposed in [170] that the spectrum of control modes in a man-machine system spans between *manual control* and *fully automatic control*, and *supervisory control* bridges the gap between the two extremes, which is further divided into two modes according to the level of the authority of the automaton: *control trading* and *control sharing*. *Control sharing* means that a human operator (supervisor) exerts control on some control variables of an automaton while the automaton itself has control over the remaining variables in contrast to the *control trading* where the supervisor issues a command to initiate a task, and the automaton performs the task autonomously having full control on all control variables during the procedure leading to the relatively higher involvement in the control loop with *control sharing* than with *control trading* for the supervisor. Machine autonomy represents the automatic exertion of control on the automaton entity itself, and since *shared autonomy* and *supervisory control* are both characterized by the level of involvement of autonomy during control flow (*i.e.* from *control sharing* to *control trading*), the two concepts share the same implications and will be used interchangeably in the following parts of this chapter.

In the field of shared autonomy between human and robot, a majority of the state-of-art works deal with teleoperation [46] (including telepresence robot [126]) to address the leading challenge on best coordinating the two sources of intelligence from human and robot to maximize the task performance of the whole system. But there are still many important works addressing the same challenge from that the human shares autonomy with the robot *locally*, such as with intelligent wheelchair [111], walking assistant for the elderly [171] and the driver-assisted automotive (usually refers to semi-autonomous vehicle) [150]. Therefore the related works from both situations, *i.e.* the autonomy is shared locally and remotely, will be introduced together below.

Autonomy sharing is not a concrete approach for human-robot system design but rather an identified concept from observations. Hence this concept has to be transformed into a practical paradigm to achieve a specific human-robot system. Goertz [71] presented manipulators for dealing with radioactive material that are able to turn cranks based on imprecise operator inputs in 1963, introducing one of the first instances of assistive teleoperation. In 1992, Hirzinger *et al.* [84] introduced the ROTEX experiment – the flight of a small multi-sensory robot on space shuttle Columbia. To deal with the large communication delay in space robotics, they tested the idea of Tele-Sensor Programming (TSP), distributing the robot to

the independent execution of partial low-level tasks based on local sensory feedback, and the human operator to the handling of coarse planning activities on a high task-oriented level, which is one of the first studies proposing the notion *shared autonomy* to describe the concept for the human-robot system design. Since then, state-of-art studies in this field have provided a great variety of implementations of the concept on different human-robot systems thanks to the advances in science and technology and the broad integration of robot into the human society in the last decades.

Unfortunately, till now there are no review works regarding this research field. Goodrich *et al.* [74] presented a survey of human-robot interaction in 2007, but it does not include the works conducted in the last decade. Moreover, the survey focuses on presenting a general overview about human-robot interaction by trying to cover all points of the field, instead of specifically discussing the research achievements and existing gaps in the field of shared autonomy between human and robot for assisting human with task execution. To facilitate the analysis of the state-of-art works in this field, certain criterion is needed to categorize them.

It is noticed that to assist task executions, the ultimate goal of a robotic system is to exert action, which is formulated by its autonomy, including information acquisition and process, decision making and action generation. Following the statements in [10], the robotic action can be approximately grouped into three control levels based on the complexity and the time span of the action: low-level actions (instantaneous and reactive), middle-level actions (often referred to *action primitives*) and complex behavioral actions for high-level control. The higher level action consists of the elementary actions at the lower level, and requires higher autonomy level for information acquisition and process, decision making and action generation. Depending on the level of the autonomy the robot contributes to the control loop during autonomy sharing, most state-of-art works in the field of shared autonomy between human and robot can be categorized into three groups: the robot operates with High Level Autonomy (HLA); the robot assists the human operator with Middle Level Autonomy (MLA); and the robot participates with Low Level Autonomy (LLA). Regarding reasoning about when and how to share autonomy between human and robot, the state-of-art works can be further indexed by whether they adopt a *model-based* approach, *i.e.* the shared autonomy policy is explicitly handcrafted, or a *data-driven* approach, *i.e.* such model is learned with the given demonstration data, to achieve it. Another indispensable factor to categorize them is the mechanism/device they employ for the information exchange between human and robot[1], such as mechanical *joystick*, *haptic device*, *Brain-Computer Interface*

[1]The focus is how the user issues commands to the robot.

(BCI), *dialogue*, *2D mouse*, and *multi-modal interface* (including gesture and speech), since the interaction modality relates to how the user input is modeled and integrated into a shared autonomy system.

In the next section of this chapter, an overview of approximately 80 related studies will be presented following these criteria, aiming to provide a comprehensive view on the achievements in the last decades in this research field.

2.2 Overview of Related Works

This section provides a comprehensive overview of the state-of-art research achievements in the field of shared autonomy between human and robot for assisting human with collaborative task execution. Table 2.1 lists approximately 80 related works using the criteria for categorization proposed in the previous section, together with the year of the works. Next to the category, there is a short introduction regarding the motivation, *e.g.* the research aim or the targeted application scenario of the proposed approach, and the employed strategy of each referred study in the table. Resulting from the lack of the common database and that the experiments were conducted with different subjects in different scenarios, it is unfortunately impossible to compare the performances, *e.g.* running time and user preferences, across the proposed shared autonomy systems/approaches in the table.

Regarding reasoning about when and how to share autonomy between human and robot, as shown in Table 2.1, most works incorporating the robotic autonomy at high level (HLA) or low level (LLA) employ model-based approaches to implement it by programming the robot with the state-of-art techniques in perception and control in advance, *e.g.* using prefedined metrics to evaluate and blend user inputs together with the behavior-based approach [11] or the potential field approach [89] to achieve robotic assistance at low level [194, 185, 113]. There are also works in the group of MLA adopting model-based approaches to build shared autonomy policies, such as [162] and [140]. Data-driven approaches are broadly employed by the works in the group of MLA, to devise shared autonomy models for the robot by learning from demonstration data, such as GMR [82], Hidden Markov Model (HMM) [142], Linear Regression (LR) [41], GP [87], Dynamic Bayesian Network (DBN) [201] or SVM [1]. There are still works in the group of HLA or LLA adopting data-driving approaches, such as Partially Observable Markov Decision Process (POMDP) [179] or Case-Based Reasoning (CBR) [193]. Two referred works [73, 159] incorporate the robotic intelligence on all three

action levels with model-based approaches, yet the human user needs to explicitly issue command or requirement to switch between the different autonomy levels through the user interface.

Regarding the mechanism of how the human and the robot exchange information during interaction, dialogue is widely employed between the human supervisor and the robot in works in the group of HLA. In these works, the robotic autonomy handles the whole procedure of executing a complex task, and the human supervisor is responsible to provide suggestions or corrections through the user interface to intervene the process to fulfil the task aim when deviation exists. Various communication channels/devices are employed by works in the groups of MLA and LLA. The prevailing one is continuous mechanical joystick. Such device is easy to model and inexpensive, yet provides no feedback to the human during operation. An alternative interface to the mechanical joystick is the haptic joystick, which is hard to model and expensive, yet capable of providing force feedback to the human user. Appropriate force feedback can greatly enhance the SA of the user during the task process, and efficiently assist the user with the task execution [141, 4, 58], but delicate control models are needed to achieve this. Another emerging and exciting user device is BCI. It transforms the mind of the human operator to the task command for the robot without hand movement. Due to the high computational demand and the limitations in the computational models for decoding human mind, the BCI currently yields only discrete low-level motion commands from human, such as left, right, front and back. Therefore, many works adopting BCI incorporate the robotic intelligence at low level (LLA) [120, 185, 28, 31, 29], *e.g.* assisting obstacle avoidance during operation. In the group of MLA, there are several works assisting the goal-based actions captured by BCI [91, 158, 48]. In addition to the above channels/devices, multi-modal interface is also a popular choice for information exchange in shared autonomy between human and robot, including gesture and speech/voice, to increase the interaction efficiency by facilitating a natural interaction between human and robot [55, 149, 172, 90, 47].

Category	Motivation and Approach
2000 [107]	Assist the operation of an intelligent walker for the frail blind.
MLA	The manually devised BN combines user input with elementary
Data-driven	semantic information of the environment derived from the sensors,
Force-torque handle	*e.g.* the doorway position, the structure of the corridor, to provide
	a context-aware estimate and assistance of the current navigation
	goals of the user in indoor scenarios.

Continued on next page

Category	Motivation and Approach
2001 [55] HLA Model-based Multi-modal Interface	Assist mobile robot teleoperation. This paper presents the design of a multi-modal user interface to supervise the mobile robot teleoperation. The data from a variety of 3D sensors (ladar, sonar, stereo vision) are fused for display to increase the SA of the human operator, and the robot is able to query the suggestions from the human operator through dialogues during task execution. The human operator can also command the robot with gestures, yet without speech recognition.
2001 [39] MLA Model-based Haptic device	Assist an undersea manipulation task (the *connector-mating* task in this paper) in remote. Within the proposed shared control approach, the connector automatically rotates to the estimated socket orientation, and the operator retains the control of the translational motion of the remote robot to perform the connector insertion. The effectiveness of the propose approach is evaluated together with various display sources.
2002 [172] LLA Model-based Multi-modal Interface	Assist wheelchair driving. Within the proposed approach, the voice of the human operator is used to control an intelligent wheelchair in an intuitive way. The user drives the robot with discrete verbal commands, *e.g.* go forward, turn soft/hard left, *etc.*, and the robot provides safety assistance during command execution, *e.g.* obstacle avoidance.
2002 [36] LLA Model-based Joystick	Assist mobile robot teleoperation. This work employs the potential field algorithm to blend the user input and obstacle information from the depth sensor (sonar in this paper), to guarantee a safe execution of the user command.
2003 [56] HLA Model-based Dialogue	Assist mobile robot teleoperation. A collaborative control approach is proposed, where the robot asks questions to the human operator by launching a dialogue, in order to obtain human assistance in cognition and perception when it encounters difficulties in executing remote tasks. This approach enables the human supervisor to function as a resource for the robot and help compensate for limitations of autonomy. A user study is performed to evaluate the effectiveness of the proposed method.

Continued on next page

Category	Motivation and Approach
2003 [112] MLA Data-driven Haptic device	Assist telemanipulation. The proposed approach employs HMMs to automatically segment and recognize user motions in a combined curve following and object avoidance task, and the segmentation results are used to provide appropriate assistance in the form of a *virtual fixture*.
2003 [121] MLA Model-based Haptic device	Assist telemanipulation. This work describes and demonstrates control algorithms for developing motion constraints. They are designed to enhance the accuracy and speed of a user manipulating in an environment with the assistance of a cooperative or telerobotic system. The proposed method uses a pool of preferred directions created off-line or in real-time with sensor data, to generate virtual fixtures both in open loop and closed loop.
2003 [200] LLA Model-based Force-torque handle	Assist operation of an intelligent walker for the elderly people. This work infers the desired navigation heading of the human user from measuring forces and moments applied to the handles of the walker, then the robotic walker assists the user intention (the desired driving heading) while taking safety into consideration by perceiving the environment during operation.
2003 [204] LLA Model-based Force-torque handle	Assist operation of a personal aid for mobility and health monitoring for the elderly people. A bi-level shared control system is reported. Its first level employs a six-axis force/toque sensor to provide a natural and intuitive human-robot interface by mapping the user force input into the robot motion command. The second-level implements a shared controller to blend the control inputs of the user and the robot based on metrics evaluating the efficiency of the user tracking a pre-planned trajectory.
2004 [145] MLA Model-based Virtual keyboard	Assist wheelchair driving. This work presents a shared control approach incorporating human inputs in motion planning for a smart wheelchair. Depending on whether the user input leads to the goal target, or is going to result into obstacle collision, it will be used to fully or partially control the robot during the execution of a plan or behavior.

Continued on next page

Category	Motivation and Approach
2005 [75] MLA Model-based Haptic device	Assist vehicle teleoperation. This work investigates the use of haptic interface to realize and test the idea of a human driver sharing control of vehicle heading with an automatic controller in the task of path following. This work demonstrates the benefits of the proposed haptic shared control approach in improving the execution performance of not only the primary driving task, but also the secondary task.
2005 [90] HLA Model-based Multi-modal interface	Facilitate natural programming of a mobile service robot in domestic environments. This work presents a multi-modal user interface, which allows user to employ gestures and speeches to program a cleaning robot interactively for executing a sequence of tasks in indoor scenarios in a natural way.
2005 [1] MLA Data-driven Haptic device	Assist telemanipulation. A HMM/SVM hybrid state sequence analyser is reported to obtain an online state estimation regarding which sub-trajectory the human operator is performing, hence the assistance can be provided adaptively to the inferred subtask in the form of virtual fixture.
2005 [104] MLA Model-based Gesture	Assist remote object manipulation. A traded/shared control approach is introduced to assist object manipulation in remote using non-contacting vision-based human-robot interface, where the human operator firstly guides the robot arm to the target object, then transfers control to the robot for performing fine alignment and centering of the gripper with the object. After the alignment, the human operator controls only the motion of the end-effector in one of its degrees of freedom, while the robot controls the other degrees of freedom of it.
2005 [206] MLA Data-driven Haptic device	Assist telemanipulation. HMM is used with the decomposed velocity vector during trajectory following using the end-effector, to classify and recognize three kinds of user motion intentions, *i.e.* following the preplanned path, aligning target and avoiding previously unknown obstacles, and designs different assistance functions for the three subtasks respectively, attempting to provide appropriate assistance upon consideration of the user motion intentions during operation.

Continued on next page

Category	Motivation and Approach
2006 [141] MLA Model-based Haptic device	Assist teleoperation task with force feedback. A haptic shared control approach is reported, where the haptic device contributes to task execution via force commands from an automatic controller. The proposed haptic shared control assistance is compared to passive virtual fixtures and no assistance to assess performance enhancement for a dynamic manual target-hitting task.
2007 [194] LLA Model-based Joystick	Assist mobile robot teleoperation. This work proposes a method to share control between human operator and robot at reactive level, where the user inputs are blended with the autonomous commands from the robot at each point of a given trajectory by estimating their respective local navigational efficiencies at each time instant.
2007 [41] MLA Data-driven Joystick	Assist wheelchair driving. This work models the wheelchair driver as path tracking controller, and employs RBF and linear regression to learn and recognize which trajectory the wheelchair driver is following, hence the corresponding assistance can be provided to help with the trajectory following during operation.
2007 [6] MLA Data-driven Joystick	Assist wheelchair driving. A probabilistic framework is presented for adaptive plan recognition and shared control to assist wheelchair driving. Users are modeled as path tracking controllers steering the wheelchair based on their mental trajectories. As users might not have a constant driving performance, or since they might suffer from various forms of disabilities, a personalized driving profile is introduced to account for these effects. It relates the actual driving behavior of the user to a reference, idealized user corresponding to the previously mentioned path tracking controller. With the estimated user plan, an adaptive decision approach is taken to assist the wheelchair driver to track the fine path by incorporating the uncertainty on the estimated plan.

Continued on next page

Category	Motivation and Approach
2007 [96] HLA Model-based GUI	Assist remote environment recognition with a group of heterogeneous mobile robots. This work reports to build a shared environment representation by a human-robot team, including ground and flying robots. Two probabilistic fusion models, *i.e.* geometric and visual, are used to describe outdoor environment features, and both are able incorporate observations from robotic platforms and human operators. Hence, humans and robots form a heterogeneous sensor network.
2008 [87] MLA Data-driven Joystick	Assist wheelchair driving. SOGP is employed to learn and recognize the user plan (modeled as the mental trajectory of the wheelchair driver for tracking in this work), which can be adapted online to any type of driving style.
2008 [24] MLA Model-based Joystick	Assist wheelchair driving. A computational model is proposed to describe the action of the wheelchair driver crossing doorways. The proposed shared control system provides assistance to the user upon recognizing the doorway-crossing action with the model from the user inputs and robot states in the environment, *e.g.* robot poses regarding candidate doorways.
2008 [3] MLA Data-driven Force-torque sensor	Assist teleoperation. A layered HMM is employed to model human manipulation skills in the form of trajectory tracking hierarchically, aiming to provide appropriate assistance to telemanipulation upon recognition of the human motion intentions with the learned models.
2008 [179] HLA Data-driven Joystick	Assist wheelchair driving. This paper presents a POMDP-based shared control system, in order to predict the intended semantic destination of the user in a typical office arena, *e.g.* kitchen, bathroom, *etc.*, with minimal user input obtained from a standard wheelchair joystick, and provide navigation assistance towards the estimated destination.

Continued on next page

Category	Motivation and Approach
2008 [189] LLA Model-based Head-movement sensor	Assist wheelchair driving. A shared control strategy is introduced to assist wheelchair driver in performing obstacle avoidance tasks. The proposed system uses a head-movement sensor as the use interface, which outputs discrete motion commands. The decisions are made from the proposed shared control approach, which blends user motion commands, accessible space of the environment from the laser scanner, and motion smoothness to provide goal directions for safe navigation based on cost functions.
2009 [153] LLA Model-based Joystick	Assist mobile robot teleoperation. This paper presents a reactive shared control approach. It blends user inputs with autonomous commands from the robot according to the navigation efficiency of the human operator and the robot respectively. The navigation efficiency of the human operator and the robot is computed from a set of manually devised criteria at every time instant given a trajectory to follow. Hence, the proposed approach guarantees the operation safe while respecting the control authorities of the human user. The autonomous commands are generated with the potential field algorithm.
2009 [193] LLA Data-driven Joystick	Assist wheelchair driving. To improve the user acceptance of the assistance from the robot, this work employs CBR to learn how the user drives the wheelchair beforehand. Then the learned model generates corresponding motion commands to be blended with the user inputs, when similar situations are encountered during operation.
2009 [120] LLA Model-based BCI	Assist wheelchair driving. This paper presents a BCI shared control system, where the wheelchair driver issues four discrete motor commands (turning left and right, and going forward and back) through the BCI, and an autonomous navigation system safely executes the issued commands.
2009 [91] MLA Model-based BCI	Assist wheelchair driving. A BCI shared control approach is reported, where a desired location is selected from a list of predefined locations from a BCI, and then sent to an autonomous system. The autonomous controller then drives the wheelchair to the desired locations safely.

Continued on next page

Category	Motivation and Approach
2009 [190] MLA Model-based Joystick	Assist wheelchair driving. This work considers to recognize and assist three kinds of tasks a wheelchair performs, *i.e.* general obstacle avoidance, corridor and wall following, and door passing, within a RBF. According to the proposed filter model, the user inputs and the environment perceptions are considered separately when recognizing the three tasks, instead of being used together, then if there is a door nearby, the intended task is highly door passing, without taking user input into consideration. Such approach can yield problem when there exists multiple candidate environment components simultaneously.
2010 [191] LLA Model-based Joystick	Assist wheelchair driving. The proposed approach estimates how much help the user needs in a reactive manner, and continuously blends the user inputs with the robot motion commands based on their navigation efficiency. Such efficiency of both sides is evaluated with a set of manually devised criteria. To improve the stability and smooth the motion blending, instant commands are modulated by a factor depending on human efficiency in a shifting time window.
2010 [185] LLA Model-based BCI	Assist mobile robot telepresense. The human operator issues discrete motion commands from BCI, *e.g.* turn left or right, and the proposed shared control approach will modify the user input with the highest priority by turning the robot towards the opposite direction where the obstacle is detected until the path is free.
2010 [140] MLA Model-based Joystick	Assist the teleoperation of tracked vehicles. Based on the continuous three-dimensional terrain scanning, the proposed shared autonomy system assists the control of the flippers of the robot to relieve the workload of the human operator, when the robot is teleoperated to traverse rough terrains.
2010 [158] HLA Model-based BCI	Assist wheelchair driving. Within the proposed system, the user can select a destination amongst a list of predefined semantic locations in an indoor scenario, then the wheelchair is guided on the planned paths which are ensured smooth and safe. The user can modify the global plan by stopping the robot using BCI, and selecting a new destination again.

Continued on next page

Category	Motivation and Approach
2010 [98] HLA Model-based Dialogue	Assist mobile robot teleoperation. The human operator supervises the autonomous navigation of a mobile robot from its sensor data. The operator functions as remotely located, valuable information sources, and the robot decides when to query the operator based on the Value-Of-Information theory within a manually designed inference diagram during operation by launching a dialogue, *i.e.* the operator is only queried if the expected benefit of the observation exceeds the cost of obtaining it.
2010 [148] HLA Model-based Low-throughput device	Assist wheelchair driving. The proposed approach is designed for low-throughput interfaces, *e.g.* BCI. During operation, the robot keeps suggesting the most probable action, as analyzed from the environmental context with a BN, to the user through movement. The user can either accept or reject the proposition by issuing the feedback through the input device.
2010 [73] LLA+MLA+HLA Model-based Multi-modal Interface	Assist the operation of an intelligent shopping trolley. The control of the robot is shared or traded between the robot and its dedicated user ranging from closely coupled haptic-based interaction up to loosely coupled command-based interaction. Four operational modes, *i.e.* the manual steering, following, guiding and autonomous mode, are implemented, and the transitions between them are achieved with the associated interaction modalities.
2011 [192] LLA Model-based Joystick	Assist wheelchair driving. This paper presents a reactive shared control approach, where the motion commands from the robot and the user become different goals in a Potential Field. Then the attractors of the robot and the user are weighted by their respective local efficiency at each time instant, and the proposed approach blends both inputs according to their weights.
2011 [113] LLA Model-based Joystick	Assist wheelchair driving. The user inputs are evaluated against a set of manually devised criteria, and a mini-max multi-objective optimization algorithm is employed to decide the weights of the user inputs for combination with the automatic motion commands during operation at every time instant given a global trajectory.

Continued on next page

Category	Motivation and Approach
2011 [147] LLA Model-based Joystick	Assist wheelchair driving. This work extracts a disability profile from a wheelchair user from the way the user drives. This profile is then used to increase or decrease the amount of help provided by the wheelchair depending on the user driving skills. The profile is extracted by means of hierarchical clustering of traces captured from a large number of volunteers driving the wheelchair in a real indoor environment. The proposed shared control model blends user and robot commands at reactive level after weighting them by their corresponding local efficiency computed from their expected performance at each situation.
2011 [30] LLA Model-based Mechanical Button	Assist wheelchair driving. The wheelchair user issues discrete motion commands from two temporally-constrained discrete buttons, and the robot is responsible for obstacle avoidance. Two reactive shared control strategies, *i.e.* a simple discrete policy and a more dynamic proportional policy, are compared with an intelligent wheelchair in two indoor scenarios with different clutter levels.
2011 [151] HLA Model-based GUI	Assist remote mobile manipulation task. Within the proposed shared autonomy system, the robotic mobile manipulator is responsible for executing most tasks to perform an assigned fetch task in remote autonomously, and the human operator is queried to provide perception assistance to the robot, when it fails to detect the desired manipulation target in a cluttered environment. In this way, a shared autonomy system at high level is realized.
2011 [118] HLA Model-based BCI	Assist wheelchair driving. Since BCI-actuated commands are issued sparsely, this paper proposes a two-layer shared controller for assisting navigation. One of the layers enables BCI inputs from the wheelchair driver according to certain situations, and the other layer blends the user inputs with the robot motion commands to generate a smooth shared motion command to drive the robot.
2012 [4] MLA Model-based Haptic device	Assist automotive driving. This work discusses several realizations of haptic shared control in literature, *e.g.* haptic shared control with fixed or variable authority, and conducts case studies in the context of automotive driving, *e.g.* for lane keeping and curve negotiation, to demonstrate the effectiveness of haptic shared control approach for assisting automotive driving in general.

Continued on next page

Category	Motivation and Approach
2012 [46] MLA Data-driven Gesture	Formalize assistive teleoperation to consist of intention prediction of user intention and its arbitration with the user input. This paper employs simplified assumptions to derive the model for user intention recognition. The efficiency of assistance regarding various factors, *e.g.* the arbitration type, the prediction correctness and the task difficulty, is evaluated within a user study.
2012 [82] MLA Data-driven 2D Mouse	Assist telemanipulation. This paper reports a shared autonomy approach applied to the problem of controlling a 6D robot manipulator using 2D mouse input. The proposed system recognizes and predicts the *free-form* tasks the human operator performs, *i.e.* the tasks associated with no semantic components in the working space, with a RBF and a learned GMR model. A cooperative motion planner continuously updates the robot trajectories to achieve the inferred tasks by repeatedly solving optimal control problems.
2012 [58] MLA Model-based Haptic device	Assist teleoperation of multiple quadrotors. This paper presents a bilateral shared control architecture. It consists of a topological motion controller for the mutual interactions in the quadrotor formation, and a human assistance module to accept human intervention. It increases the telepresence of the human operators with the force-feedback.
2012 [162] MLA Model-based Joystick	Assist the human operator to safely control a quadrotor to inspect a vertical infrastructure, such as a utility pole on the street. This paper reports a shared control model. Within the proposed model, the robot is responsible to keep safe distance to the target for inspection, and assist the human operator to always face the target, while the human operator can focus on inspection task by issuing just approximate motion commands without worrying about the safety of the robot during operation.
2012 [48] MLA Model-based BCI	Assist mobile robot telepresence. The human operator selects visible target destinations or exploration areas using the BCI, and the robot executes the motion commands autonomously in remote.

Continued on next page

Category	Motivation and Approach
2012 [27] LLA Model-based Keyboard	Assist mobile robot teleoperation. The human operator issues discrete motion commands, *e.g.* turn left or right, to simulate the BCI outputs, and the robot is responsible for obstacle avoidance. This paper designs a set of task-dependent performance metrics to adapt the assistance level from the robot during operation, *i.e.* to quantitatively decide how much assistance the human operator needs.
2012 [26] MLA Model-based Joystick	Assist wheelchair driving. The proposed collaborative control system employs a multi-hypotheses method to predict the navigation intentions of the human operator, *e.g.* which door to pass, and if necessary, adjusts the control signals to achieve the desired goal safely. This work conducts a comprehensive evaluation in indoor scenarios to demonstrate the effectiveness of the proposed approach to improve performance on not only primary driving task but also secondary task during operation.
2012 [123] MLA Model-based Haptic device	Assist mobile robot teleoperation. Within the proposed shared control system, the robot takes charge of executing cyclic motions, while the human operator concentrates on modifying online some geometric properties of the desired path. This modification from the user is processed by the robot in order to produce an available trajectory respecting tracking feasibility, obstacle avoidance, closeness to the desired trajectory of the human operator, and proximity to some points of interest. The human operator is informed of the global deformation of the path with force feedback.
2012 [70] MLA Model-based Haptic device	Assist road vehicle teleoperation in urban environments. Within the proposed shared autonomy system, the human operator takes charge of situation analysis, behavioral decision making, and provides a sequence of trajectories to command a robotic road vehicle for path following in remote. The generated trajectories consist of parameterized curves overlain by velocity control. The robot executes the commanded trajectory autonomously.

Continued on next page

Category	Motivation and Approach
2012 [202] HLA Model-based Joystick	Assist mobile robot teleoperation. This paper presents a quantitative measure that infers the degree of trust of the human operator in the robot autonomy, as well as an adaptation strategy for adjusting the robotic autonomous behaviors to accord with the expectations of the human operator after receiving intervening commands from the human team member. The application scenario is that a human operator supervises an autonomous flying robot to do boundary tracking in remote.
2012 [165] HLA Model-based GUI	Assist telemanipulation. A telemanipulation task is executed within a state machine autonomously, and supervised by a human operator. A generalized logic recovery algorithm is reported to decide when to query the human intervention for state machine recovery, if certain state fails during operation, *e.g.* fails to detect or manipulate the target object in a cluttered environment, balancing the system performance and work load of the human supervisor.
2013 [88] MLA Data-driven Joystick	Assist wheelchair driving. A probabilistic framework is proposed based on RBF to recognize local navigation trajectories of human user driving an intelligent wheelchair, aiming to provide assistance to the execution of the estimated local plan. The proposed probabilistic framework employs GP Regression to learn the likelihood model of the filter, and a DBN with a probabilistic distance likelihood to calibrate the transition model of the filter.
2013 [173] LLA Data-driven Joystick	Assist wheelchair driving. The proposed shared autonomy system employs GP to learn when how to help a wheelchair user with the difficult tasks from expert demonstrations. The decision to provide assistance is solely made by the expert knowledge, and the environment is described by using simply all the laser data without further process, which may render the proposed approach constrained to a specific scenario.

Continued on next page

Category	Motivation and Approach
2013 [201] MLA Data-driven GUI	Assist remote object grasping. This paper reports a collaborative human-robot system. It takes advantage of the semantic knowledge of a human co-worker who provides additional context information and interacts with the robot through a GUI. A BN is employed to learn and encode the dependencies between the information provided by the user. The output of this model generates a ranked list of grasp poses best suitable for a given task, which is then passed to the motion planner.
2013 [29] LLA Model-based BCI	Assist wheelchair driving. A shared control architecture with BCI is presented, where the wheelchair driver issues discrete motion commands from a BCI, and the proposed shared control system safely executes the user motion commands by taking obstacle information into consideration, which is obtained from a occupancy grid.
2013 [164] MLA Model-based Haptic device	Assist automotive driving for lane keeping on wheel steering. This paper presents a shared control approach derived from an H2-Preview optimization control problem based on a global DVR system. The DVR model adopts a cybernetic driver model to take into account any driver-vehicle interactions.
2013 [83] MLA Model-based Data glove	Assist remote object grasping. Within the proposed shared autonomy approach, the human operator issues open/close commands from finger movements to initialize an automatic grasping of the target object. This work also conducts a user study evaluating the grasping performance and perceived workload of the human operator with the proposed shared autonomy approach when working with different assistance modes and hand kinematics.
2013 [175] MLA Data-driven Haptic device	Assist remote repairing task for a broken hard drive. This paper reports a haptic shared control system employing a discrete HMM to recognize the actions of the human operator from the haptic data. It designs context-specific assistance functions to assist the human operator upon action recognition.

Continued on next page

Category	Motivation and Approach
2013 [35] MLA Model-based BCI	Assist remote grasping with a brain-controlled robot hand. The proposed shared control approach extends the traditional virtual fixture concept to coordinated motion in high-dimensional control spaces, *i.e.* to guide both translational and rotational degree-of-freedom of the robotic hand towards whole sets of robot poses that would allow an object to be grasped. The proposed approach allows for irregular fixture shapes.
2013 [72] MLA Data-driven Joystick	Assist wheelchair driving. The proposed shared control approach employs GMR model to learn task variability from demonstration examples, and extracts allowable user command constraints from the variance. This variance is then used to blend user and robot control in challenging navigation scenarios, like door passing. The human operator needs to explicitly indicate the start of the task execution to activate the proposed approach.
2013 [8] MLA Model-based Haptic device	Assist vehicle teleoperation. A shared control approach is presented to assist semi-autonomous vehicle teleoperation for hazard avoidance and stability control based on the design and selective enforcement of constraints. It identifies safe trajectory homotopies, and allows the operator to navigate freely within them. It exerts control action only as necessary to ensure that the vehicle does not violate the pre-defined safety constraints.
2014 [142] MLA Data-driven Force-torque handle	Assist the operation of an intelligent walker for the elderly. This work employs HMM to analyze the walking patterns of the human user, *i.e.* the gaits, from demonstration data collected from a LRF, aiming to provide context-based support, and intuitive assistance to the user of the device, *i.e.* mainly the elderly people, in domestic environments.
2014 [177] LLA Model-based Joystick	Assist mobile robot teleoperation. A MPC shared control framework is reported to blend human inputs with autonomous motion inputs to assist mobile robot teleoperation with obstacle avoidance. The proposed framework also considers that how the human input differs from that of an autonomous controller in addition to threat of collision. It is applied to a high speed differential drive robot moving through an obstacle field.

Continued on next page

Category	Motivation and Approach
2014 [116] LLA Model-based Virtual joystick	Assist teleoperation of an omni-directional mobile robot. The proposed shared control system decides the human and robot interactive gains by computing user's confidence factor from a set of manually devised criteria. The robot assists a user with the remote motion control by compensating local insufficiency of the human control.
2014 [207] LLA Model-based Joystick	Assist wheelchair driving and human following. The proposed shared control system blends the user inputs, from either manual driving commands or the pose of the target human to follow, with the autonomous motion commands using virtual field force principle taking obstacle information into account, to generate a safe shared control command to drive the robot.
2014 [197] LLA Model-based Haptic device	Assist operation of a load carrying robot for the elderly people. The proposed shared controller blends the user inputs with the autonomous motion commands from the user tracking controller and the obstacle avoidance controller according to their weights computed from the system stability analysis, to generate a new shared control output to command the robot.
2014 [124] MLA Model-based Haptic device	Assist mobile robot teleoperation. This paper presents a haptic shared control approach, where the human operator can modify the shape of the navigation path of the robot by controlling the motion of a finite number of control points. An autonomous controller corrects in real time the human inputs in order to facilitate path tracking for the mobile robot, and ensures the safety, regularity of the generated path, and its attraction to nearby points of interest.
2014 [9] MLA Model-based Haptic device	Assist vehicle teleoperation. The proposed shared control approach identifies path homotopies, bounds a desired homotopy with constraints, and allocates control as necessary to ensure that these constraints remain satisfied without unduly restricting the human operator.

Continued on next page

Category	Motivation and Approach
2014 [203] HLA Model-based Joystick	Assist mobile robot operation for collaborative task, *e.g.* the human-robot team is assigned with covering or patrolling over multiple terrain boundaries such as coastlines and roads. This paper introduces a shared autonomy framework to quickly adapt the autonomous behavior of the robot with simple intervening commands from human operators. During operation, it continuously explores the space of all possible parameter configurations for the robot system online, using the information deduced from the latest intervening commands of the human team member, attempting to quickly configure the robot autonomy with respect to changes in task objectives and conditions.
2014 [15] HLA Model-based GUI	Assist telemanipulation. This paper presents a knowledge-driven tablet computer application for commanding a robot on a high level of abstraction. The application guides an operator to make decisions based on the current world state of the robot, and enables the operator to command object-centered actions. These actions are interpreted symbolically and geometrically by the robot autonomously.
2015 [169] HLA Model-based Haptic device	Assist teleoperation of surgical robot. This paper reports a paced shared-control framework for teleoperated surgical systems. It includes a *dominance* factor to set the authority of the human operator and autonomous agent over the slave robot, as well as an *aggressiveness* factor to set the performance pace of the autonomous agent according to the state of the slave robot.
2015 [205] HLA Model-based Haptic device	Assist mobile robot teleoperation. This paper proposes the admittance and impedance models to smoothly blend user inputs with robot autonomy, and this is enabled by the force feedback capability of the haptic interface. With the proposed haptic shared control approach, the human operator can provide flexible assistance to the robot upon observation of its difficulty in remote task execution through the GUI, *e.g.* the robot gets stuck during navigation, with the grasp motion on the haptic device.

Continued on next page

Category	Motivation and Approach
2015 [159] LLA+MLA+HLA Model-based GUI	Assist mobile robot teleoperation. This paper presents an assistive robot vehicle prototype in the context of a smart environment. It allows the human operator with physical disability to interact with a smart environment by means of hand-free devices. The proposed system also integrates four control strategies ranging from low level to high level, *i.e.* manual control, shared control with assistance to obstacle avoidance, point to go, and fully autonomous. The human operator is required to explicitly select one of the four modes to control the client vehicle in remote.
2015 [198] LLA Model-based Haptic device	Assist telemanipulation. This paper proposes a shared control model, where the human operator is made to concentrate on the motion of the end-effector of the manipulator, while the robot manipulator takes charge of avoiding obstacle without compromising the motion performance of the end effector.
2015 [94] MLA Data-driven Gesture	Assist remote object grasping. This paper formulates the shared autonomy problem of assisting remote object grasping as a POMDP with uncertainty over the goal of the human operator. It employs the maximum entropy inverse optimal control to infer a distribution over the goal of the human operator with the input history. Since it is intractable to solve the POMDP to select the optimal action, the *hindsight optimization* is used to approximate the solution.
2015 [59] HLA Model-based Un-specified	Research on the theory of control switching in a general shared autonomy system under temporal logic specifications. This paper presents a two-stage policy synthesis algorithm for generating Pareto efficient coordination and control policies with respect to user specified weights. The proposed algorithm is applied to a cooperative motion planning problem for a mobile robot in a stochastic environment in simulation.

Continued on next page

Category	Motivation and Approach
2016 [34] HLA Model-based Joystick	Assist mobile robot teleoperation. This paper introduces a straightforward shared autonomy approach, where the human operator decides either to take full control of the robot, or simply indicate a target for the robot to autonomously navigate to during operation. Such strategy is called *Human-Initiative* in this work. This approach and the other two control modes for comparison, *i.e.* the manual mode and the fully autonomous mode (where the user can only assign navigation targets to the robot), are evaluated in a user study in simulation. The experimental results confirm the effectiveness of the proposed Human-Initiative approach in enhancing the performance of the human operator on the primary task compared with the other two modes.

Table 2.1 Category and short introduction (motivation and approach) of the related works in the field of shared autonomy between human and robot for assisting human with collaborative task execution from 2000 to 2016.

2.3 Detailed Introduction and Discussion of Several Representative Works

In this chapter, several representative related works will be discussed in details.

1. Formalization of Assistive Teleoperation [46]

By summarizing the related works in the field of assistive teleoperation, this paper formalizes assistance under the general framework of policy blending, and presents a principled analysis of the main components of an assistive teleoperation system: prediction of user intention and its arbitration with the user input (the conceptual depiction is shown in Fig. 2.1). In this work, it is emphasized that the robot is supposed to predict the user intention, and assist to accomplish it, rather than directly execute the user input, due to the interface inadequacies. The authors define the prediction problem, discuss about simplifying assumptions that make

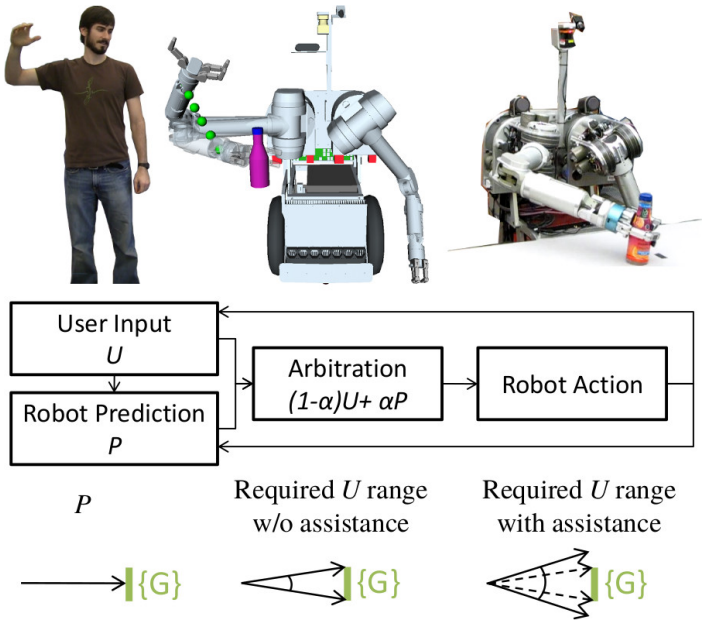


Fig. 2.1 (Top) The user provides an input U. The robot predicts their intentions, and assists them to achieve the task. (Middle) Policy blending arbitrates user input and robot prediction of user intentions. (Bottom) Policy blending increases the range of feasible user inputs (here, $\alpha = 0.5$). Cited from [46].

it tractable, and test the built model on data from users teleoperating a robotic manipulator under various configurations in a user study.

The formalization unifies prior work, and provides common ground for future methods and comparisons of assistive teleoperation, offering a valuable reference for the research conducted in this field. More specifically, the authors made an interesting user study on the relations between the arbitration methods based on the intention prediction, the task difficulty, and the efficiency of the assistance in terms of task completion time and user preferences. Two arbitration methods were evaluated : *timid*, *i.e.* the assistance increases with the confidence of the intention prediction till a predefined maximal value, preventing the robot from taking the full charge of the control process; and *aggressive*, *i.e.* the robot eagerly takes charge as soon as the confidence exceeds a threshold, which presents a valuable guidance for the choice of arbitration strategy adopted by the assistance system. However, this paper focuses on the telemanipulation task: it only considers to predict and assist the motion pattern of reaching the grasp target, while the teleoperation of a mobile robot involves more types of motion patterns, such as moving around a target for inspection or following a wall segment in indoor navigation. Meanwhile, focusing on the formalization of the concept of assistive teleoperation, this paper employs simplified assumptions for ease of deriving the

model for intention prediction, *e.g.* the user intention is to track *optimal* trajectories towards the grasp objects. Although the paper mentions the data-driven approaches to address this problem, it does not introduce them in detail.

2. Recognition and Prediction of Freeform Tasks to Assist Telemanipulation [82]

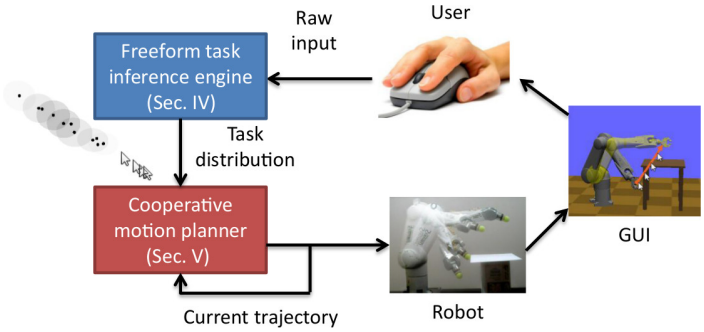


Fig. 2.2 The overall architecture of the system proposed in [82]. Cited from [82].

To assist the human operator more intuitively and responsively, this paper proposes the notion of free-form tasks that encode an infinite number of possible goals (*e.g.* desired target positions) within a finite set of types (*e.g.* reach, pick up) to model the user intention, and employs the data-driven approach to derive the models for the considered free-form tasks (the overall architecture of the proposed system is depicted in Fig. 2.2). The proposed system was evaluated in simulation within the problem of controlling a 6D robot manipulator using 2D mouse input in the context of two tasks: static target reaching and dynamic trajectory tracking.

This paper achieves an impressive state-of-art in the field of assisted teleoperation by sharing autonomy, and inspires part of the contributions made in this thesis. However, the authors consider only free-form tasks which involve with no contextual information of the environment in this paper, while in practice a teleoperation task is mostly associated with certain semantic components of the environment, *e.g.* an object to grasp, a doorway to cross and a wall segment to follow, *etc*. Therefore, such contextual information is supposed to be integrated when interpreting the user intention, and further contributions are needed to recognize the user intention with contextual information, *i.e.* to correctly infer the on-going contextual tasks instead of simple free-form tasks, in order to provide *context-dependent* assistance for efficient task execution in remote. Such point is also mentioned in this paper as one of its future work.

3. User-adapted Shared Control by Recognizing User Intention as Trajectory to Track to Assist Wheelchair Driving [41]

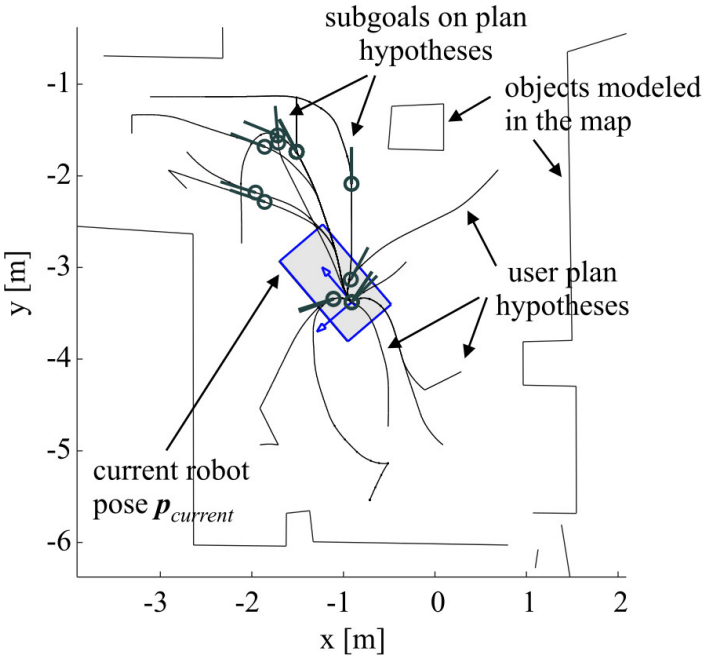


Fig. 2.3 House-like environment where experiments with the intelligent wheelchair have been conducted. The figure depicts the wheelchair along with calculated paths to possible goal poses at a certain moment in time. The paths are computed with a planner in the (x, y, θ) configuration space. The planner takes the robot geometry and kinematic constraints into account. Also shown are subgoals on the paths, which are used to predict the user signals. Cited from [41].

This paper presents a novel framework based on Bayesian approach for the recognition of the user driving plan and sharing control in a probabilistic way, to assist the human user, mostly elderly and physically impaired people, to efficiently drive an intelligent wheelchair in indoor scenarios. Its main innovation with respect to previous approaches is to estimate the user driving intention in order to provide adaptive driving assistance that is able to be tailored to the driving skills of individual users. Towards this aim, the user driving intention is modeled as a goal pose and goal twist together with a trajectory to achieve the goal pose and twist. The wheelchair user is then modeled as a path-tracking controller that outputs uncertain control signals (the concept is illustrated in Fig. 2.3). In this manner, the proposed framework is able to model and estimate even complex maneuvers. Intention paths are calculated in the framework with a fine-motion planner that takes the geometry and kinematic constraints of the robotic platform into account, such that the framework can be adapted to different wheelchair types. Additionally, the decision making is modeled as a POMDP, allowing to execute informed assistance decisions. Experiments have been conducted to

demonstrate the feasibility of the novel framework both in simulation and on a real robotic wheelchair.

This work contributes significantly to the field of shared autonomy with mobile robotics, which is based on and lays the foundations for a group of works concentrating on the trajectory recognition to assist wheelchair driving [42, 43, 40, 6, 87, 88]. However, it is assumed by the proposed approach that the user tracks optimal trajectories to the designated goal states, where the candidate trajectories for estimation and tracking (referred to plan hypotheses in this paper) were generated by the Voronoi graph search and a geometric fine motion planner, which largely limits the assistance provided by the system to the generated plan hypotheses, and hinders the flexibility of the whole assistive system. Consequently, further contributions are required to enhance the flexibility of the assistance provided by the system.

4. Perceptual Shared Autonomy to Assist Mobile Manipulation [151]

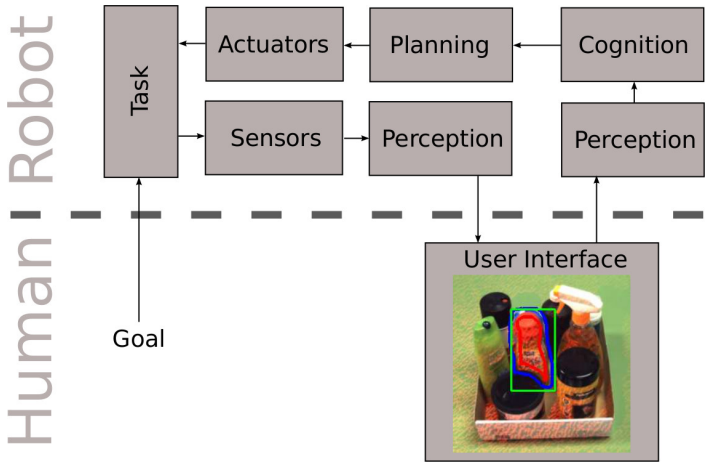


Fig. 2.4 The perceptual shared autonomy system model for robotic manipulation. A human is included to close the perception loop through collaborative object selection. Cited from [151].

This paper presents a shared autonomy system capable of solving the perceptual inference in cooperation with a human, such that a human operator is considered as a valuable and reliable resource by the robot and helps to compensate for limitations of autonomy, aiming to demonstrate that how a human-robot team can work together effectively to solve complex perception tasks in telemanipulation, because reliable perception is one of the most challenging problems in such task for robot. More specifically, the proposed system asks a human operator to identify objects it doesn't recognize or find, by drawing rectangles and performing simple strokes to separate objects from background areas in color images (the

proposed system is depicted in Fig. 2.4). Various experiments with the PR2 robot confirms that this shared autonomy system performs more robustly than the robot system alone.

This paper proposes to incorporate human assistance for complex perception tasks requested by the robot from time to time, while the robot handles the whole process for a complex mobile manipulation task autonomously. It impressively demonstrates the effectiveness of a human-robot team working together, thus clearly makes an important contribution to the field of shared autonomy. However, to implement an autonomous robot to execute a complex task, the robot needs to be programmed with the advanced knowledge of the state-of-art robotic perception and control techniques beforehand. Such requirement brings about a serious limitation to the system for real applications in the long run, *i.e.* the system is limited to execute the pre-programmed task. Unfortunately, to determine all the needed task functions users would like a robot to have in advance is hardly possible due to many social and economic factors. So if the robot is needed to execute new task types different from those already being programmed, to implement the adaptation, the human user is supposed to have a highly professional programming skill and a deep understanding of the robotic techniques, which definitely prevents the average users from incorporating such system into everyday life. To address the issue to promote the applicability of the robotic assistive system in the long period, data-driven approaches are worth considering to derive the models for the tasks from demonstration data in the further contributions.

5. Modelling and Recognition of Human Operational Motions with Layered HMM to Assist Telemanipulation [3]

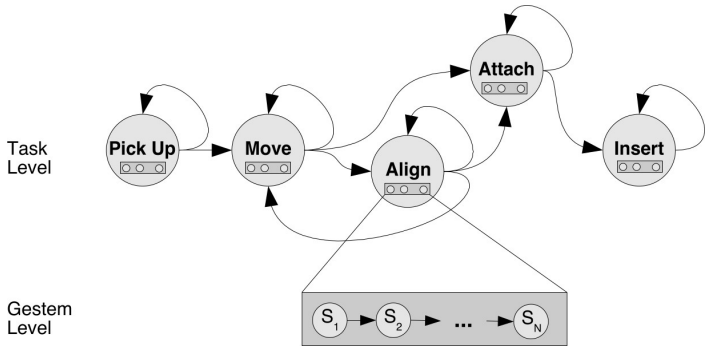


Fig. 2.5 A two level layered hidden Markov model, modeling gestemes at level 2 and a task at level 1. Cited from [3].

This work presents a data-driven approach to acquire, represent and model human skills in the context of assisted telemanipulation, with the aim of providing motion assistance upon skill/task recognition. It proposes to use a Layered Hidden Markov Model (LHMM) to model

human skills. Three different HMM models are employed and compared to derive the models for the basic action primitives (referred to *gestemes* in the paper) from the demonstration data at the low level, in order to account for the temporal property of the gestemes, then the gesteme classifiers are further used in LHMM to model a complex task at the high level (the used model is illustrated in Fig. 2.5). The online and off-line classification performance of the three gestemes models was evaluated with respect to the number of gestemes, the influence of the size of training samples, the noise factor and the effect of the number of observation symbols. The LHMM was also applied to data recorded during the execution of a trajectory tracking task in 2D and 3D with a mobile manipulator in order to provide qualitative as well as quantitative results for the proposed approach.

Although this work certainly achieves an interesting state-of-the-art in the field of assisted telemanipulation and shared autonomy, the considered action primitives are fixed trajectory patterns involving with no objects in the environment, *i.e.* the end-effector of the manipulator was force-moved by a human operator simply in either straight line or circle, limiting the flexibility of the proposed approach for broader deployments, especially when it is applied to model and recognize the tasks of mobile robot teleoperation, since it is inadequate to describe these tasks simply with fixed trajectory patterns without considering the contextual information of the environment. Therefore, further contributions are needed to propose novel approaches to model and derive the policies of various teleoperation tasks taking the contextual information into consideration. Moreover, the employment of HMM requires that the number of unobservable hidden states must be specified *a priori* or chosen via model selection, which is prone to over-fitting. This severely limits the usefulness of HMM inference when dealing with unstructured data.

6. Reactive Shared Autonomy with Efficiency-weighted Strategy for Fusing Human and Robot Inputs to Assist Wheelchair Driving [153]

To assist the human user to efficiently execute global trajectories when teleoperating a mobile robot in indoor scenarios, this paper presents a novel approach to share control between mobile robots and humans. The key idea of this paper is to measure the efficiency of both sides at each sampling time in a reactive way with the predefined evaluation criteria. With the metric results, the motion commands from human and robot can be weighted and linearly blended into a single motion command (the proposed approach is illustrated in Fig. 2.6). The authors aim to combine the advantages of both sides in a seamless way, by continuously blending their motion commands. The proposed method was evaluated with a real robot and 13 volunteers, and results showed the effectiveness of the proposed

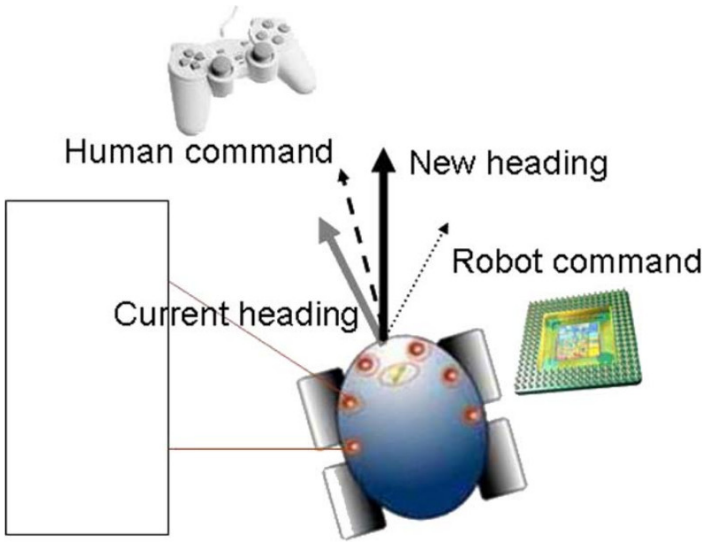


Fig. 2.6 Combination of robot and human commands. Cited from [153].

approach in improving the driving performance of the human operator in most cases, and the navigation performance of the robot in many cases, *e.g.* when pure reactive control failed, such as crossing doors or approaching close obstacles.

This work contributes significantly to the field of shared autonomy with mobile robotics, and lays the foundations for a group of works focusing on the efficiency based reactive shared autonomy to assist mobile robot operation, especially robotic wheelchair driving [193, 52, 191, 192, 147, 113]. Though being intuitive and fast, the proposed approach requires a planned trajectory to work on in advance, which by default restricts the user on the given global trajectory during operation. Another limitation is that the professional knowledge is needed to design the criteria manually, which is the cornerstone of the proposed approach. Therefore, it is hard to envision that the proposed approach is able to scale with more complex situations in assisting robot operation, when the existing criteria need to be adapted.

2.4 Summary and Discussion of Research Gaps

This chapter presents a comprehensive analysis of the state-of-art works in the field of shared autonomy between human and robot for assisting human with collaborative task execution. Shared autonomy is a valuable and promising concept for system design, attracting more research attentions in robotics, thanks to the deeper integration of the robotic system within

the human society. To efficiently implement it on a specific human-robot system, a major challenge needs to be addressed, *i.e.* how to best coordinate the two sources of intelligence from human and robot, to achieve an adaptive and flexible assistance and maximize the performance of the human-robot team through interaction in real applications. Attributed to the advances in perception and control techniques in the last two decades, the state-of-art works in the field of shared autonomy provide a great variety of solutions. These studies focus on various robotic applications, such as telemanipulation and mobile robot teleoperation, and incorporate the robotic intelligence at different control levels: from that the robot handles most parts of the task procedure autonomously, to that the robot provides only time-instant assistance reactively.

However, this challenge remains unsolved, preventing the shared autonomy system from being widely deployed in real applications, especially in those considered for life-long time, where the robots measure their life-time in years instead of minutes. Incorporating the robotic intelligence at the high control level can significantly reduce the human involvements to supervisory actions from time to time, *e.g.* providing feedback or corrections to the robotic system by request, but a profound knowledge of the robotic techniques and computer programming is required to implement the autonomous controller with high level autonomy for certain complex task beforehand. This extremely limits the adaptability of such system to new task types in the long run. Meanwhile, based on the current achievements in the perception and control techniques, it is still impossible to program the robot to execute many complex tasks with high level autonomy in real applications. On the other hand, though being intuitive and fast, providing time-instant assistance can largely restrict the human operator to certain fixed plan the reactive controller of the robot follows, ignoring the human intentions during task executions, which can substantially reduce the assistance efficiency by essentially increasing the human rejections to the robotic assistance in many complex situations. A promising research direction is thus to incorporate the robotic intelligence at the middle control level, *i.e.* providing assistance to action primitives, for two reasons: 1) a sequence of action primitives can constitute complex tasks on the higher level; 2) human intention can be modeled as action primitive to execute, so the robot is able to assist what the human intends to do instead of what the user is currently doing.

However, it is non-trivial to devise the control models for all required action primitives from scratch before deploying the robotic system in real applications. Therefore, data-driven approaches are employed in the literature to derive the control policies for action primitives from demonstration data. Nevertheless, the state-of-art works conducted in this direction either assume the action primitives to be fixed motion patterns, *e.g.* following a

certain trajectory segment, or do not take the contextual information of the environment, *e.g.* objects to be manipulated, into consideration, when deriving the policies for modeling action primitives. To provide adaptive and flexible assistance during operation by efficiently modeling task intention of the human operator, further contributions are needed to encode the contextual information into the models, and learn the motion patterns with the contextual features without making assumptions *a priori*.

Chapter 3

System Formalization

This chapter firstly describes the proposed contextual-task aware shared autonomy system *mathematically*, introducing the mathematical symbols used throughout the following parts of the thesis. Notably, the problem tackled in this thesis is formalized, leading to the major contributions of this thesis presented in the following three chapters. In addition to the formalization, a reactive shared autonomy system based on BN is also reported in this chapter. Although it is not among the major contributions, the discussions regarding its technical limitations in assisting mobile robot teleoperation reveal possible improvements, inspiring the author of this thesis to propose an alternative Bayesian contextual task recognition approach. Such approach learns the motion patterns of various task executions from human demonstrations, and will be detailed in the next chapter.

3.1 Mathematical Description of the Proposed System and Problem Formalization

During mobile robot teleoperation, the human operator observes environment perceptions $\mathbf{z}$ obtained by the robot sensors through certain GUI, and issues user input $\mathbf{u}$ with an input device, *e.g.* a mechanical joystick considered in this thesis, in the form of velocity sent to the robot for execution, attempting to complete an action primitive to fulfil a contextual task i latent in the mind of the human operator. Examples of i include "cross the doorway in front", "inspect the object on the right side of the robot", *etc.* Meanwhile, with $\mathbf{z}$, the robot is able to generate efficient motion commands $\mathbf{g}$ to autonomously execute i based on the state-of-art

works in perception and control. The robot then assists the human operator by blending the user input $\mathbf{u}$ and the robot motion command $\mathbf{g}$ according to an arbitration factor β to achieve the ultimate motion command $\mathbf{v}$ for execution:

$$\mathbf{v} = (1-\beta)\mathbf{u} + \beta\mathbf{g}, \tag{3.1}$$

where $0 \leqslant \beta \leqslant 1$.

To express that the combination is calculated in each sampling cycle of the robot, and the arbitration factor β is not fixed during operation, the time index t is appended to each variable of Eq. (3.1):

$$\mathbf{v}_t = (1-\beta_t)\mathbf{u}_t + \beta_t\mathbf{g}_t. \tag{3.2}$$

Eq. (3.2) mathematically describes the proposed shared autonomy framework, which is abstracted from and hence independent of any specific robotic hardware configuration. In this way, the level of autonomy is seamlessly switched between the manual control (*i.e.* $\beta_t = 0$) and the autonomous control (*i.e.* $\beta_t = 1$) during operation. As can be noticed, to achieve an efficient assistance for the human operator, the key is to decide an appropriate arbitration factor β_t. In this work, β_t is modeled as the measure of *confidence* in recognizing the task i_t latent in the mind of the human operator at the time t, aiming to provide motion assistance in a task-appropriate way.

To compute β_t and account for the uncertainty in data acquiring and processing, with the contextual information, *i.e.* the user input $\mathbf{u}_t$ and the environment information $\mathbf{z}_t$, it is necessary to solve the following probability problem to infer the latent task i_t the human operator is likely to execute:

$$p(i_t|\mathbf{u}_t, \mathbf{z}_t) = ? \tag{3.3}$$

Considering that there exists possibly multiple candidate task instances for inference during operation, β_t is finally obtained by maximizing Eq. (3.3):

$$\beta_t = \max p(i_t|\mathbf{u}_t, \mathbf{z}_t). \tag{3.4}$$

Hence Eq. (3.4) closes the loop of the proposed shared autonomy system (Eq. (3.2)).

Eq. (3.3) formalizes the research problem to be tackled in this thesis, *i.e.* given the contextual information, how can the on-going task the human operator performs be estimated in a probabilistic manner? In the following section, a reactive shared autonomy approach will

be introduced, which assists the human operator by recognizing a set of pre-defined simple situations encountered during operation within a BN, to provide a preliminary solution for this problem. This approach was proposed in the initial phase of the study conducted in this thesis and inspired the major contributions afterwards.

3.2 Reactive Shared Autonomy by Situation Recognition with BN

This section reports a reactive shared autonomy system whose architecture is depicted in Fig. 3.1. The human operator observes the surrounding environment through the robot sensors, and issues the control command via joystick according to the perceived information and the task context (unknown to the robot). The proposed approach identifies possible situations encountered during teleoperation, and provides the corresponding motion assistances at reactive level. A classical design paradigm called *situated-action* [11] is employed to blend the motion commands from the human operator and the robot. The problem is simplified by using the "divide and conquer" strategy to identify the situation and execute the corresponding assistive action exclusively. A BN is used to implement the situation identification to account for the uncertainties of acquiring and processing user inputs and environment information, while assisting the human operator implicitly: each assistive action is like a robot-providing "button" for handling the corresponding situation; within the proposed probabilistic situation identification framework, the robot is able to implicitly switch between the "buttons" to assist the human operator correspondingly, while the human operator is not required to do the switch manually. The implementation of the proposed reactive shared autonomy method is detailed below.

3.2.1 Method Implementation

This sub-section begins with the introduction of the definitions of the situations and their corresponding assistive actions, then the probability model used to identify the situations will be presented in detail. In the framework of the proposed method, the robot is required to equip with sensors capable of producing a 2D depth map of obstacles surrounding the robot. The most common kinds of such sensors are sonar and 2D LRF. Since the proposed

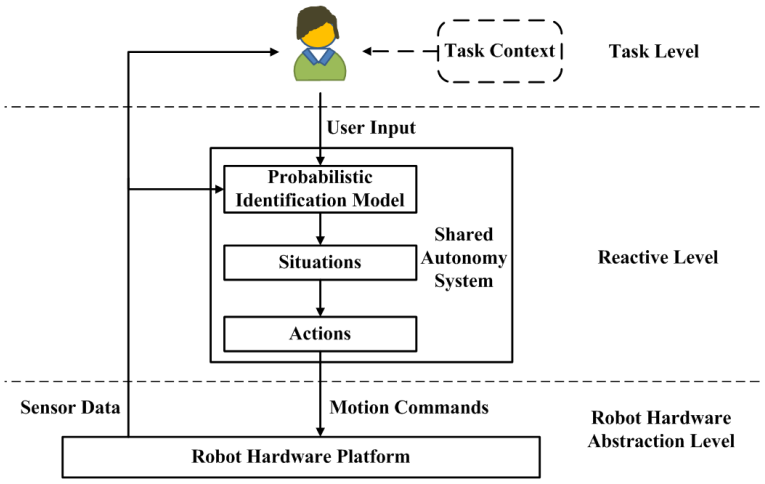


Fig. 3.1 Architecture of the reactive shared autonomy system

method is reactive, the process of situation identification and corresponding action execution is repeated for each sensor update without considering the past information.

3.2.1.1 Set of situations and corresponding actions

The *situated-action* paradigm is used to design the reactive shared autonomy method. This paradigm is based on defining a set of situations that describe the relative state of the shared autonomy entities (*i.e.* the robot, the obstacle distribution, and the user input), and the actions associated with each of the candidate situations. During the execution phase, the environment information and the user input are captured to identify the current situation, then the associated action is carried out.

Three situations are considered: Free Travel (FT), Object Approach (OA) and General Obstacle Avoidance (GOA). They are among the most possible situations encountered by teleoperating a mobile robot without manipulation capability. The symbols introduced in Sec. 3.1 are also used to describe the corresponding actions in this part.

FT: The system is in FT when there are no obstacles within a security zone around the robot bounds. Then the user input will be fully executed by the robot, as the robot assumes that it is safe enough to do so in this situation. The corresponding action can be described as:

$$\mathbf{v}_t = \mathbf{u}_t . \tag{3.5}$$

OA: The system is in OA when the user controls the robot towards an object that is within the security zone of the robot. The robot should keep a safe distance from the facing object when being controlled to approach it. The corresponding action is described as:

$$\mathbf{v}_t = \alpha_t \mathbf{u}_t, \tag{3.6}$$

where:

$$\alpha_t = \begin{cases} \exp(-\frac{1}{\mathrm{d}_t - \mathrm{d}_{\mathrm{safe}}}) & \text{if } \mathrm{d}_t > \mathrm{d}_{\mathrm{safe}}; \\ 0 & \text{otherwise}, \end{cases} \tag{3.7}$$

is the attenuation coefficient for decreasing the motion command of the robot, d_t is the closest distance between the robot and the facing obstacle measured by the range finder of the robot at time t, and $\mathrm{d}_{\mathrm{safe}}$ is the predefined safe distance that the robot must keep away from the approaching object.

GOA: The system is in GOA when there are obstacles within the security zone of the robot, and the user operates the robot in the free area among the obstacles. The corresponding action is realized by using the Dynamic Window Approach (DWA) [57], which is a reactive obstacle avoidance strategy for mobile robots. The action can be concisely described as:

$$\begin{aligned} \mathbf{v}_t =& \mathbf{g}_t \\ =& \mathrm{DWA}(\mathbf{u}_t, \mathbf{z}_t) \end{aligned} \tag{3.8}$$

The OA mainly aims to handle the situation when the user drives the robot to approach a certain object and stop at a certain distance from it, *e.g.* for inspection. The GOA, however, deals with the situation that the user considers the surrounding objects as obstacles, and demands to bypass them to avoid the potential collisions actively. Meanwhile, the normal reactive controller, *e.g.* DWA, tries to find a free movement direction among the obstacles to control the robot to bypass the obstacles safely, but does not allow the robot to go nearer towards the obstacle. Therefore it is reasonable to distinguish the two situations and address them separately in a shared autonomy system design.

The overview of the BN used for the probabilistic situation identification will be introduced in the next part.

3.2.1.2 BN for the probabilistic identification

From the view of the human operator, the candidate situation is decided by the user action according to the environment observation, and the candidate situation equals to its corresponding action in the *situated-action* paradigm. Therefore, the probabilistic situation identification is based on the action recognition, meaning that, when a certain action is recognized, its corresponding situation is identified.

The BN used for the probabilistic situation identification is shown in Fig. 3.2. It is assumed that the human operator takes the action[1] by observing the environment from the robot sensor capture ($\mathbf{z}_t$), then the user issues the control command ($\mathbf{u}_t$) to the robot based on both the action (decides what to do) and the sensor observations (decides how to do).

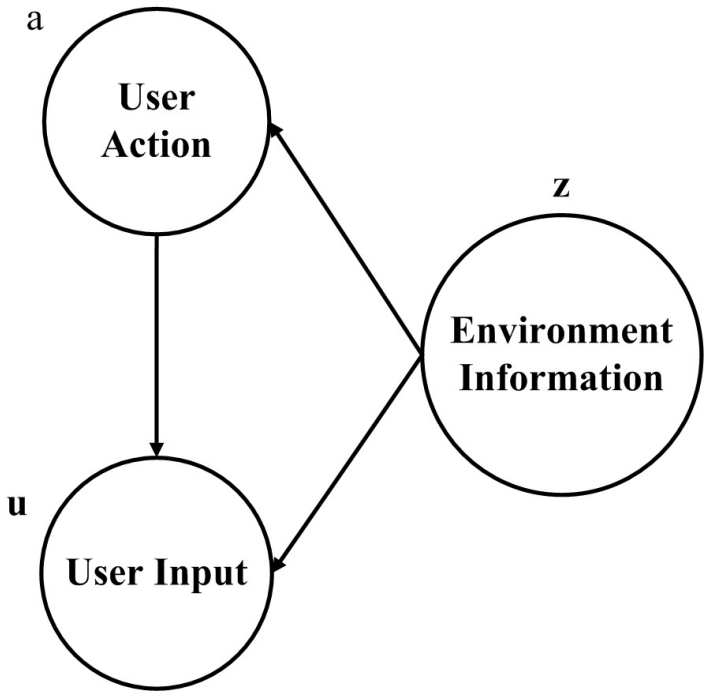


Fig. 3.2 The Bayesian Network (BN) for the situation identification

According to the *Bayes Rule* and the BN (Fig. 3.2), given the environment observation $\mathbf{z}_t$ and the user input $\mathbf{u}_t$, the formula for the situation identification is:

$$p(a_t|\mathbf{u}_t,\mathbf{z}_t) = \eta\, p(\mathbf{u}_t|a_t,\mathbf{z}_t)p(a_t|\mathbf{z}_t), \tag{3.9}$$

[1] An action taken at time t is denoted with the symbol $a_t \in A$, where A represents a set of candidate actions for execution, *i.e.* FT, OA and GOA.

where $\eta = \frac{1}{p(\mathbf{u}_t|\mathbf{z}_t)}$ is the normalization factor. The recognized action a_t^r is selected through the Maximum a Posteriori (MAP) method:

$$a_t^r = \arg\max_{a_t \in A} p(a_t|\mathbf{u}_t, \mathbf{z}_t). \tag{3.10}$$

Consequently, following Eq. (3.2) to get the final motion command, from the proposed reactive shared autonomy system, the arbitration factor β_t is obtained:

$$\beta_t = \max p(a_t|\mathbf{u}_t, \mathbf{z}_t), \tag{3.11}$$

and the robot motion command $\mathbf{g}_t$ generated from the recognized action a_t^r.

The technical details of the environment model and the user input model used in the BN will be presented in the following parts.

3.2.1.3 Environment model

The node *Environment Information* of the BN (Fig. 3.2) is modelled with a local occupancy grid map centered at the robot. The map is updated using the range finder data based on the method described in [183]. The angle range of the range finder is divided equally into N sectors, and each sector is set to cover 5°. The j-th sector is denoted as s_j. Within the border of the occupancy grid map, each angle sector s_j covers a set of occupancy grid cells $M_j = \{m_{jk}\}$, where the index k denotes the k-th grid cell of the sector. Let S denote the local occupancy map, then $S = \{s_j\}$.

For each occupancy grid cell m_{jk}, it has two states: Occupied (including the unknown state) and Free. Each state will be assigned a cost value:

$$f(m_{jk}) = \begin{cases} 1 \text{ if } m_{jk} = \text{Occupied } or \text{ Unknown} \\ 0 \text{ if } m_{jk} = \text{Free} \end{cases} \tag{3.12}$$

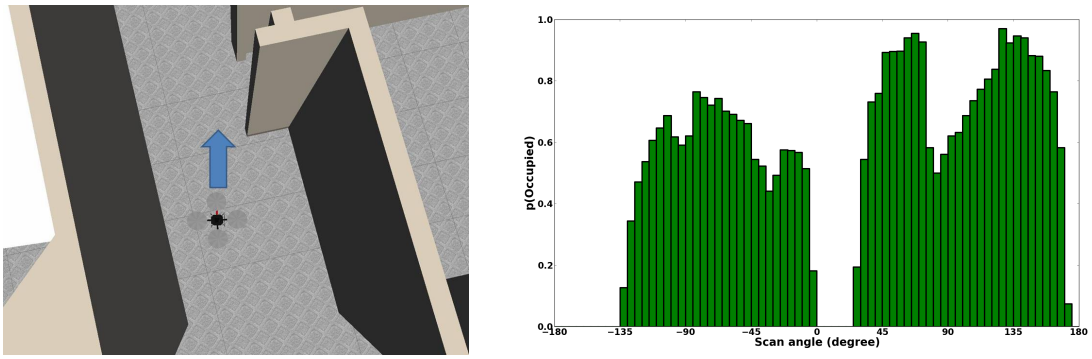
It is assumed that the sectors are independent from each other respecting the measurements of the range finder. Therefore, the probability of the occupied situation of each angle sector can be calculated as:

$$p(s_j = \text{Occupied}) = \frac{\sum_k f(m_{jk})}{|M_j|} \tag{3.13}$$

The probability $p(s_j = \text{Occupied})$ encodes the belief of the robot about the movement safety in the direction indicated by the angle sector s_j. $p(S = \text{Occupied})$ represents the probability of the existence of the obstacles around the robot within the update radius of the map:

$$p(S = \text{Occupied}) = \max_j p(s_j = \text{Occupied}) \tag{3.14}$$

An example of the occupied probability $p(\text{Occupied})$ is illustrated in the following figures. Fig. 3.3(a) shows a captured scene of a simulated quadrotor equipped with a 2D LRF facing several obstacles. Fig. 3.3(b) exhibits the occupied probability $p(\text{Occupied})$ calculated from this scene. In Fig. 3.3(b), the X-axis represents the angle range of the LRF, where the 0° represents the straight ahead direction of the LRF (and also the robot, indicated by the arrow in Fig. 3.3(a)), and the positive angles result from the counter-clockwise rotation centered at the LRF. Each bar denotes the occupied probability of the corresponding angle sector. Note that in Fig. 3.3(b), the occupied probabilities of all angle sectors are displayed together for convenience, although the angle sectors are independent from each other respecting their occupied states.



(a) (b)

Fig. 3.3 Example of the occupied probability: (a) The virtual quadrotor is facing obstacles around it; (b) The occupied probability of each angle sector.

The next part will describe the implementation of the user input model of the BN.

3.2.1.4 User input model

The user input is in the form of an angle calculated from the joystick displacement. This angle can be transformed into the velocity commands along x and y directions of the local

coordinate system fixed on the robot center to drive the robot. Hence the user input is expressed as a *scale* instead of a vector in the following parts of this sub-section.

During mobile robot teleoperation, it is normally struggling for the human user to precisely issue the displacement corresponding to his/her operational intention by pushing a mechanical joystick, while the human user is focusing on the GUI to compute appropriate robot action other than joystick operation. Meanwhile, due to possibly imprecise calibration, the displacement is usually measured with errors by the sensor of the joystick. To account for these uncertainties, the user input is modelled with a normal distribution in the proposed approach:

$$p(\theta_u) = \varepsilon \mathcal{N}(\theta_u; \theta^*, \sigma^2), \tag{3.15}$$

where ε is the normalization factor, θ^* is the real user input, and θ_u is the output of the joystick that results from θ^* with the influence of noise σ.

For each angle sector s_j of the local occupancy map S, let θ_{istart} and θ_{iend} denote the start angle and the end angle of it respectively, and it is assumed that $\theta_{\text{istart}} \leqslant \theta_{\text{iend}}$. The probability of a given user input u pointing at a certain angle sector s_j is calculated as:

$$p(u_t \in s_j) = \int_{\theta_{\text{istart}}}^{\theta_{\text{iend}}} p(\theta_u), \tag{3.16}$$

where $p(\theta_u)$ is defined by Eq. (3.15). Therefore, the probability of a given user input pointing at an occupied angle sector s_j is defined by:

$$\begin{aligned} &p(u_t \in s_j, s_j = \text{Occupied}) \\ &= p(u_t \in s_j) p(s_j = \text{Occupied}), \end{aligned} \tag{3.17}$$

where $p(u_t \in s_j)$ is defined by Eq. (3.16), and $p(s_j = \text{Occupied})$ is defined in Eq. (3.13). Then the probability of the user input pointing at a free angle sector s_j is defined by:

$$\begin{aligned} &p(u_t \in s_j, s_j = \text{Free}) \\ &= p(u_t \in s_j)[1 - p(s_j = \text{Occupied})] \end{aligned} \tag{3.18}$$

The final part of this sub-section will detail the probabilistic model for each candidate situation, *i.e.* the *user action* node of the BN.

3.2.1.5 Probabilistic models of the situations

The FT is identified when it is highly probable that there is no obstacle within a predefined security zone of the robot (the a-priori probability of the action), and the user input points at a probably free angle sector (the likelihood probability of the action). The probability of no obstacle surrounding the robot is the negate of the occupied probability $p(S = \text{Occupied})$, therefore, based on Eq. (3.9), the probability model of the FT is:

$$\begin{aligned} p(a_t = \text{FT}|u_t, \mathbf{z}_t) &= \eta\, p(u_t|a_t = \text{FT}, \mathbf{z}_t) p(a_t = \text{FT}|\mathbf{z}_t) \\ &= \eta \max_j [p(u_t \in s_j, s_j = \text{Free})] \cdot [1 - p(S = \text{Occupied})] \end{aligned} \tag{3.19}$$

The OA is generated when it is highly probable that there are obstacles within the security zone of the robot, and the user input points at a probably occupied angle sector. The probability model of the OA is:

$$\begin{aligned} p(a_t = \text{OA}|u_t, \mathbf{z}_t) &= \eta\, p(u_t|a_t = \text{OA}, \mathbf{z}_t) p(a_t = \text{OA}|\mathbf{z}_t) \\ &= \eta \max_j [p(u_t \in s_j, s_j = \text{Occupied})] \cdot p(S = \text{Occupied}) \end{aligned} \tag{3.20}$$

The GOA is captured when it is highly probable that there are obstacles within the security zone of the robot, and the user input points at a probably free angle sector. The probability model of the GOA is:

$$\begin{aligned} p(a_t = \text{GOA}|u_t, \mathbf{z}_t) &= \eta\, p(u_t|a_t = \text{GOA}, \mathbf{z}_t) p(a_t = \text{GOA}|\mathbf{z}_t) \\ &= \eta \max_j [p(u_t \in s_j, s_j = \text{Free})] \cdot p(S = \text{Occupied}) \end{aligned} \tag{3.21}$$

3.3 Discussion

The proposed reactive shared autonomy system considers three situations for assistance during operation, and identifies them with a BN in a probabilistic manner to account for the uncertainty in the data acquiring and processing, then provides motion assistance to the recognized situation with the corresponding reactive controller. It realizes the preliminary strategy for designing an adaptive shared autonomy system to assist mobile robot teleoperation. However, its major technical limitations are two folds. Firstly, the employed

BN is manually designed, including the edges (the probabilistic relationships) between the nodes within the network, which heavily hinders the scalability of the proposed framework for further extension, *i.e.* to include more situations, since for each new situation to be considered, an estimation model has to be manually indicated correspondingly. Secondly, the provided instantaneous/reactive action mostly pertains to either *obstacle avoidance* or *collision stopping*, as the proposed system relies on a naive environment model to generate safe motions by distinguishing simply *obstacle* from *free space* in the environment. This is far from efficiently describing the *motion patterns* of the human operator controlling the robot for various task executions, since such task patterns usually comprise a spatial-temporal sequence associated with certain semantic component of the environment, *e.g.* doorway to cross, object to inspect, wall segment to follow, *etc.* To improve this, it is supposed to recognize and assist *action primitives* of the human operator with the contextual information. Such contextual action primitives are more flexible and meaningful in describing the human motion intentions than reactive actions. In addition to this, it might also be needed to employ data-driven approaches to learn motion policies for describing various task executions from human demonstrations in an intuitive way instead of manually deriving them. This is essential for the flexibility and scalability of the system.

3.4 Summary

This chapter firstly presented the mathematical description of the proposed contextual-task aware adaptive shared autonomy system, and formalized the research problem of this thesis from the mathematical description. After that, a reactive shared autonomy approach was reported as a preliminary implementation of the proposed shared autonomy system. The situated-action paradigm and the “divide and conquer" strategy were employed to design the system. Three situations were defined, *i.e.* Free Travel, Object Approach and General Obstacle Avoidance. During the operation, one of the pre-defined situations is identified based on the sensor information of the robot and the user input, then the corresponding reactive controller generates the motion commands to move the robot safely. To account for the uncertainties of acquiring and processing user inputs and environment perceptions, a BN was applied to identify the situations in a probabilistic manner, which enables the proposed system to assist the human operator implicitly. The probability models of the environment, the user input and the candidate situations were presented and implemented to realize the proposed system. Finally, the major technical limitations of the proposed system were discussed, inspiring the author of this thesis to further propose an alternative approach

to recognize and assist human action primitive based on contextual information instead of instantaneous/reactive action to address the problem in a data-driven manner. This approach will be introduced in the next chapter.

Chapter 4

Multiple Contextual Task Recognition Using GMR and RBF

This chapter presents a novel multiple contextual task recognition approach, to address the research problem formalized in the previous chapter. The approach proposed in this chapter focuses on recognizing the intentional task the human operator performs to complete an action primitive with the contextual information, which comprises a spatial-temporal sequence with uncertainty for data processing. A RBF is adopted to smoothly and robustly handle this. To implement the RBF, a data-driven approach is employed, *i.e.* GMR model to learn the motion patterns the human operator executes various task types from demonstrations, rather than manually devising the models for describing various task executions from scratch.

This chapter is structured as following. The proposed approach is concisely introduced in Sec. 4.1 with more details. Sec. 4.2 describes the task features obtained from the contextual information, with the aim of encoding the motion patterns the human operator executes various task types, to model the motion policy with the data-driven approach from the human demonstrations afterwards. Sec. 4.3 details the RBF, which is used as the base framework in the proposed approach to smoothly and robustly estimate the contextual task the human operator executes over time. The RBF consists of the *User Model* and the *State Transition Model* after derivation, and these two components need to be implemented respectively. Sec. 4.4 reports the implementation of the User Model by employing the *batch* GMR model to learn the motion patters from human demonstrations, including the model selection approach, which is the cornerstone of the proposed framework. Sec. 4.5 presents the realization of the State Transition Model, including the important assumptions made in the proposed approach.

Sec. 4.6 introduces how to integrate the proposed task recognition approach within the proposed shared autonomy system by arbitrating motion commands from the human operator and the robot based on task estimation.

Moreover, in order to favour an online learning scenario to enhance the flexibility of the proposed approach in real applications, a fast online GMR approach is proposed in Sec. 4.7 to encode and recognize various human motion patterns, to replace the batch GMR model in the User Model of the RBF. With such method, the robot is capable of adapting to the new task types on the fly while being interactively taught by the demonstrator. Finally, this chapter is summarized with the discussions regarding the proposed approach in Sec.4.8.

4.1 Approach Overview

The proposed approach aims to recognize multiple contextual tasks during mobile robot teleoperation with simple and/or noisy input devices[1]. It considers the setting of a human operator using joystick to control the robot in an indoor environment. By observing the information of the robot and the environment displayed in a GUI, the user issues motion commands which vaguely indicate the user intentions for operating the robot, such as avoiding obstacles. As clarified in Chapter 1, a *task* refers to a metric representation of the user intention for a robot to complete an *action primitive*, such as crossing a doorway, inspecting an object, or following a wall segment. Correspondingly, a *task type* indicates a group of *tasks* which share similar action targets or patterns. Therefore, a shared autonomy system integrating the proposed task recognition approach can properly assist the human operator by estimating the task the user performs. This chapter focuses particularly on *contextual tasks* which rely on the information of related objects in the environment. Following the symbols used in [82] and Chapter 3, $i_t \in 1,...,m$ denotes a task type at time t. An instance of a task type (called a *task* in this chapter) is described by a *task feature* $\boldsymbol{\phi}_t \in \mathbb{R}^n$. In this chapter, without the loss of generality of the proposed approach, $m = 4$ task types are considered: *Doorway Crossing*, *Object Inspection*, *Wall Following* and *Robot Docking*. These four task types are representative and important action primitives for teleoperating a mobile robot without manipulation capabilities.

[1] As indicated in Sec. 3.2.1.4, the noise of an input device, *e.g.* a mechanical joystick, results from two major aspects: 1) it is normally challenging for the human user to precisely issue the command corresponding to his/her operational intention via the input device, while the human user is focusing on the GUI to compute appropriate robot action other than manipulating the device during mobile robot teleoperation; 2) due to possibly imprecise calibration, the user command is usually measured with errors by the sensor of the input device.

The user input $\mathbf{u}_t$ from a mechanical joystick is in the form of velocities along x and y directions in the robot's local coordinate frame $\mathbf{u}_t = (v_{x,t}, v_{y,t})$. The input velocity value of each direction is normalized to the range of $(-1, 1)$, where the positive sign indicates that the input is along the positive direction of the corresponding axis.

The proposed task recognition approach infers a time series of the task types (i_t) with the corresponding task features $(\boldsymbol{\phi}_t)$ from a streaming series of the user inputs $\mathbf{u}_t, t = 1, 2, \ldots$. It adopts a RBF to estimate a distribution over candidate tasks. The RBF is implemented by exploiting the GMR technique, which uses Gaussian Mixture Model (GMM) to learn and represent the implicit user model required by the RBF. In the following parts of this chapter, the definitions of the task features used in each task type are firstly be described, then the technical details of the proposed multiple contextual task recognition framework are covered respectively.

4.2 Task Feature

The task feature $\boldsymbol{\phi}$ embodies an instance of a specific task type with the task descriptors obtained from the contextual information. The selected task descriptor should be intuitive and representative to describe a task type with the context information. In this section, a set of simple, compact but highly distinctive task features will be introduced to describe each task type.

If not specified otherwise, the term *distance* used in the following part of this section refers to the *Euclidean distance*, and all the feature descriptors are calculated in the local coordinate frame fixed on the robot center.

4.2.1 Doorway Crossing

The center point of the doorway q_c is considered as the key element to describe a doorway, and the following three variables according to q_c are chosen as the feature descriptors for such task type (Fig. 7.2(a)): 1) the distance d_q between the robot center and q_c; 2) the angle ψ_q between the robot heading and q_c; 3) the angle θ_q between the user input vector $\mathbf{u}$ and the vector $\mathbf{r}_q$ from the robot center to q_c. ψ_q implies whether the human operator rotates the robot to face the doorway, which is usually observed to preliminarily indicate whether

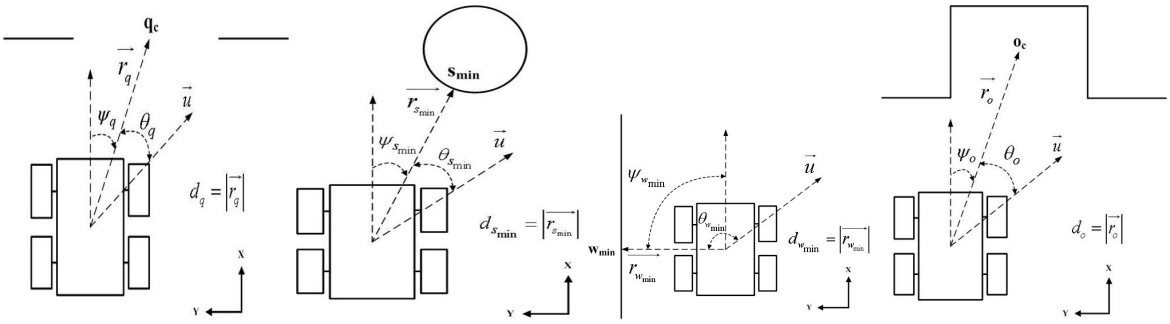


Fig. 4.1 The task features for each of the four task types: (a) Doorway Crossing; (b) Object Inspection, an circular object is used here as an example; (c) Wall Following; (d) Robot Docking.

a certain doorway is the semantic target. θ_q represents whether the user moves the robot towards a doorway in reality.

4.2.2 Object Inspection

The object considered in this task type is segmented from a 2D occupancy grid map of the environment. For each candidate object which is extracted, the surface points of it are maintained:

$$S = \{s | s \in ((x_1, y_1), ..., (x_n, y_n))\}. \tag{4.1}$$

The nearest surface point of an object to the robot center $s_{\min}$ during operation is employed as the key element to describe this task type, since the surface points of an object have already implicitly encoded the shape information of the object.

The following three variables based on $s_{\min}$ are chosen as the descriptors for this task type (Fig. 7.2(b)): 1) the distance $d_{s_{\min}}$ between the robot center and $s_{\min}$, which indicates the relevance of a certain object as the inspection target; 2) the angle $\psi_{s_{\min}}$ between the robot heading and $s_{\min}$, since a holonomic mobile robot will be used to evaluate the proposed approach (see Chapter 7), the robot is reasonably assumed to be *facing* the target object during inspection, thus this angle implies such action; 3) the angle $\theta_{s_{\min}}$ between the user input vector $\mathbf{u}$ and the vector $\mathbf{r}_{s_{\min}}$ from the robot center to $s_{\min}$. $d_{s_{\min}}$, $\psi_{s_{\min}}$ and $\theta_{s_{\min}}$ together indicate whether the user drives the robot to approach an object and then moves it closely

around the object for inspection (in either clockwise or counter-clockwise direction), which is observed to be the usual motion pattern of this task type.

4.2.3 Wall Following

The wall is also segmented from a 2D occupancy grid map of the environment. For each segmented wall, the surface points of it are maintained:

$$W = \{w | w \in ((x_1, y_1), ..., (x_n, y_n))\}. \tag{4.2}$$

The nearest surface point of a wall to the robot center $w_{\min}$ during operation is used as the key element to describe this task type, and the following three variables according to $w_{\min}$ is selected as the descriptors for this task type (Fig. 4.1(c)): 1) the distance $d_{w_{\min}}$ between the robot center and $w_{\min}$; 2) the angle $\psi_{w_{\min}}$ between the robot heading and $w_{\min}$, since when following a wall, the robot moves approximately along the surface of the wall, this angle ought to be around 90°; 3) the angle $\theta_{w_{\min}}$ between the user input vector $\mathbf{u}$ and the vector $\mathbf{r}_{w_{\min}}$ from the robot center to $w_{\min}$. $d_{w_{\min}}$, $\psi_{w_{\min}}$ and $\theta_{w_{\min}}$ together indicate whether the user drives the robot to approach a wall and then moves the robot approximately along the wall to follow it (on either left or right side), which is observed to be the usual motion pattern of this task type.

4.2.4 Robot Docking

The user usually executes this type task to drive the mobile robot to dock into a recharging station, or a table for further inspection, which can be quite difficult for the mobile robot teleoperation, especially when the gap of the docking area is narrow. The center point of a docking area o_c is used as the key element to describe this task type, and the following three variables based on o_c are employed as the descriptors for this task type (Fig. 4.1(d)): 1) the distance d_o between the robot center and o_c; 2) the angle ψ_o between the robot heading and o_c; 3) the angle θ_o between the user input vector $\mathbf{u}$ and the vector $\mathbf{r}_o$ from the robot center to o_c. ψ_o indicates whether the human operator rotates the robot to face a docking target. θ_o represents whether the user moves the robot towards a docking target in reality.

Although *Robot Docking* and *Doorway Crossing* share the similar motion pattern, the former one has different application cases and requires different contextual information, thus it is considered as a separate task type to recognize besides the latter one in this chapter.

Based on the definitions of the task features for each task type, the following part describes the RBF and the GMR implementation of the filter.

4.3 RBF for Task Recognition

The task recognition problem is to estimate the distribution over the task parameters i_t and $\boldsymbol{\phi}_t$ with the user input $\mathbf{u}_t$. To achieve this, a RBF is employed to filter this distribution over time, with the task parameters being its state variable: $\mathbf{x}_t = (i_t, \boldsymbol{\phi}_t)$. A RBF is a first-order Markov model, whose belief of the state variable $b_t = P(\mathbf{x}_t|\mathbf{u}_1, ..., \mathbf{u}_t, \mathbf{z}_1, ..., \mathbf{z}_t, \mathbf{m})$ is maintained during operation, where $\mathbf{m}$ stands for the map of the environment, and $\mathbf{z}_t$ represents the sensor measurement vector at time t. Such a model is shown in Fig. 4.2.

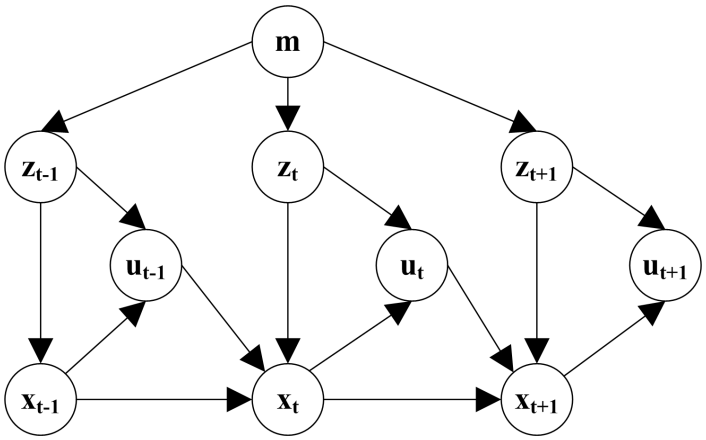


Fig. 4.2 Graphical model of the contextual task recognition system

The RBF is updated to derive the belief over x_{t+1} using the recursive manner as follows:

$$\begin{aligned} b_{t+1}(\mathbf{x}_{t+1}) =& p(\mathbf{x}_{t+1}|\mathbf{u}_1, ..., \mathbf{u}_{t+1}, \mathbf{z}_1, ..., \mathbf{z}_{t+1}, \mathbf{m}) \\ =& \eta \cdot p_{\text{user}}(\mathbf{u}_{t+1}|\mathbf{x}_{t+1}, \mathbf{z}_{t+1}, \mathbf{m}) \\ & \cdot \int_{\mathbf{x}_t} p_{\text{process}}(\mathbf{x}_{t+1}|\mathbf{x}_t, \mathbf{u}_t, \mathbf{z}_{t+1}, \mathbf{m}) b_t(\mathbf{x}_t) d\mathbf{x}_t, \end{aligned} \tag{4.3}$$

where:

1. p_{user} is the *user model*, which describes the likelihood that the user issues the motion command $\mathbf{u}_{t+1}$ to execute a specific task indicated by the state variable $\mathbf{x}_{t+1} = (i_{t+1}, \boldsymbol{\phi}_{t+1})$ and the sensor measurement $\mathbf{z}_{t+1}$. It is the cornerstone of the proposed approach, since it encodes the implicit task execution process of the user.
2. p_{process} is the *state transition model*, which predicts the distribution over $\mathbf{x}_{t+1}$ based on $\mathbf{z}_{t+1}$ and the prior information of the system $\mathbf{x}_t$, $\mathbf{u}_t$.
3. η is the scale factor to normalize the probability distribution.

Since the model is a first order Markov chain, it is reasonable to assume that: the task feature $\boldsymbol{\phi}$ encodes the necessary information from the sensor measurements $\mathbf{z}$ for the task inference, and the state variable $\mathbf{x}$ is complete to represent the user intention in the context of task inference. Therefore, the sensor measurements $\mathbf{z}$ and the map $\mathbf{m}$ will be left out both in the user model and the state transition model for notation simplicity:

$$\begin{aligned} b_{t+1}(\mathbf{x}_{t+1}) =& \eta \cdot p_{\text{user}}(\mathbf{u}_{t+1}|\mathbf{x}_{t+1}) \\ & \cdot \int_{\mathbf{x}_t} p_{\text{process}}(\mathbf{x}_{t+1}|\mathbf{x}_t, \mathbf{u}_t) b_t(\mathbf{x}_t) d\mathbf{x}_t. \end{aligned} \tag{4.4}$$

The implementation of the GMR based user model and the state transition model are introduced in the following two parts respectively.

4.4 GMR based User Model

The derivations of GMR from GMM are common knowledge in machine learning[178]. To smooth the presentation of the proposed approach in this chapter, the symbols and derivation results in [82] are employed to briefly introduce the knowledge in below. For more technique details of GMM and GMR, the readers are kindly referred to [82] and [178]. In the following part of this section, $\mathbf{x}$ and $\mathbf{y}$ denote two random vectors, and they are assumed to jointly distribute to a GMM with h components. Given $\mathbf{x}$, $p(\mathbf{y}|\mathbf{x})$ is a GMM by applying the Gaussian conditioning operation to each component in the GMM, and re-weighting the components accordingly. The final model is a GMR model:

$$\text{GMR}(\mathbf{y}|\mathbf{x}) = \frac{1}{z} \sum_{c=1}^{h} \omega_{c,\mathbf{x}} \mathcal{N}(\mathbf{y}; \boldsymbol{\mu}_{c,\mathbf{y}|\mathbf{x}}, \boldsymbol{\Sigma}_{c,\mathbf{y}|\mathbf{x}}) \tag{4.5}$$

where z is a normalization factor and $\omega_{c,\mathbf{x}}$ is the recalculated weight for the component c:

$$\omega_{c,\mathbf{x}} = \pi_c p(\mathbf{x}|c) \tag{4.6}$$

where π_c is the weight for the component c in the joint GMM, and

$$p(\mathbf{x}|c) = \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_{c,\mathbf{x}}, \boldsymbol{\Sigma}_{c,\mathbf{x}}) \tag{4.7}$$

is the marginal probability that $\mathbf{x}$ is drawn from the component c in the joint GMM. Each component $p(\mathbf{y}|\mathbf{x}, c)$ in the GMR model has the mean $\mu_{c,\mathbf{y}|\mathbf{x}}$ and the covariance $\boldsymbol{\Sigma}_{c,\mathbf{y}|\mathbf{x}}$ respectively:

$$\begin{aligned} \boldsymbol{\mu}_{c,\mathbf{y}|\mathbf{x}} &= \boldsymbol{\mu}_{c,\mathbf{y}} + \boldsymbol{\Sigma}_{c,\mathbf{yx}} \boldsymbol{\Sigma}_{c,\mathbf{x}}^{-1} (\mathbf{x} - \boldsymbol{\mu}_{c,\mathbf{x}}) \\ \boldsymbol{\Sigma}_{c,\mathbf{y}|\mathbf{x}} &= \boldsymbol{\Sigma}_{c,\mathbf{y}} - \boldsymbol{\Sigma}_{c,\mathbf{yx}} \boldsymbol{\Sigma}_{c,\mathbf{x}}^{-1} \boldsymbol{\Sigma}_{c,\mathbf{yx}} \end{aligned} \tag{4.8}$$

which are determined using the Gaussian conditioning operation from the joint GMM.

For different task types i_t, a separate GMR model is used for each of them. Therefore, the user model p_{user} is represented as:

$$\begin{aligned} p_{\text{user}}(\mathbf{u}_t|\mathbf{x}_t) &= p_{\text{user}}(\mathbf{u}_t|i_t, \boldsymbol{\phi}_t) \\ &= \text{GMR}_i(\mathbf{u}_t|\boldsymbol{\phi}_t). \end{aligned} \tag{4.9}$$

The task features encode the contextual information from both environment and the user input, and it is assumed that the GMR model will encode the high variance in the angle parameters seen when facing and approaching the target. To leverage this in the probability computation, d is used as the input data, and ψ and θ as the output data for regression[2] within different task types i to further implement the *User Model* of the RBF:

$$\begin{aligned} p_{\text{user}}(\mathbf{u}_t|\mathbf{x}_t) &= p_{\text{user}}(\mathbf{u}_t|i_t, \boldsymbol{\phi}_t) \\ &= \text{GMR}_i(\psi_t^i, \theta_t^i|d_t^i). \end{aligned} \tag{4.10}$$

For training a GMM, the Expectation-Maximization (EM) algorithm [44] is used. To decide the optimal number of components h of a GMM, the Bayesian Information Criterion (BIC) [166] is employed for its simplicity and satisfying performance. Specifically, multiple models are firstly estimated with an increasing number of components, and the BIC score of

[2]For Doorway Crossing, $d = d_q$, $\psi = \psi_q$, $\theta = \theta_q$; for Object Inspection, $d = d_{s_{\min}}$, $\psi = \psi_{s_{\min}}$, $\theta = \theta_{s_{\min}}$; for Wall Following, $d = d_{w_{\min}}$, $\psi = \psi_{w_{\min}}$, $\theta = \theta_{w_{\min}}$, and for Robot Docking, $d = d_o$, $\psi = \psi_o$, $\theta = \theta_o$, respectively.

each model is calculated to select the best among candidates:

$$s_{\text{BIC}} = -l + \frac{n_p}{2}\log(k) \tag{4.11}$$

where k is the number of $d-$dimensional datapoints, $l = \sum_{j=1}^{k} log(p(\boldsymbol{\xi}_j))$ is the log-likelihood of the model using the demonstration data $\{\boldsymbol{\xi}_j | j = 1,...,k\}$ as the test set, and n_p is the number of free parameters required for a GMM with h components and full covariance matrix:

$$n_p = (h-1) + h\left(d + \frac{d(d+1)}{2}\right). \tag{4.12}$$

As can be seen from its definition, the BIC score trades off between optimizing the model's likelihood and minimizing the number of parameters required to encode data.

4.5 State Transition Model

For the state transition model p_{process}, since the human operator issues the input based on the contextual task to be executed, the evolution of the user intention x is independent from the user input:

$$p_{\text{process}}(\mathbf{x}_{t+1}|\mathbf{x}_t, \mathbf{u}_t) = p(\mathbf{x}_{t+1}|\mathbf{x}_t). \tag{4.13}$$

During the operation, it is assumed that before an intentional task is completed, the user will not switch to other tasks, which means that the user has only one clear intention in mind during the task execution. Therefore, the state transition model can be described as follows[3]

$$p_{\text{process}}(\mathbf{x}_{t+1}|\mathbf{x}_t) = \begin{cases} \sigma & \text{if } \mathbf{x}_{t+1} = \mathbf{x}_t = \gamma \\ 1-\sigma & \text{if } \mathbf{x}_{t+1} \neq \mathbf{x}_t = \gamma, \end{cases} \tag{4.14}$$

where γ represents an intentional contextual task, and σ is the transition coefficient, which is a time scale factor and decides how fast the intention distribution switches back to the uniform distribution. Considering the above assumption and to get a balance performance, σ is set to 0.95 in this work.

[3]The equality between two contextual tasks performed sequentially ($\mathbf{x}_t$ and $\mathbf{x}_{t+1}$) indicates that they both: 1) belong to the same task type, *i.e.* "cross a doorway"; 2) are executed with the same semantic target, *i.e.* "cross the doorway in front" (instead of other doorways in the scenario).

Combined with the user model stated above, the complete task recognition framework (Eq. (4.4)) proceeds as follows: each time a set of candidate tasks is firstly predicted using p_{process}, then the observed user inputs are applied to update the belief distribution with p_{user} and the Bayes rule.

The next section introduces how to integrate the proposed contextual task recognition approach into the proposed shared autonomy system.

4.6 Motion Command Arbitration

Based on the filtered belief over task variables $b_t(i_t, \boldsymbol{\phi}_t)$, the most probable on-going task can be determined using the MAP method:

$$(i_{t,\max}, \boldsymbol{\phi}_{t,\max}) = \arg\max_{(i,\boldsymbol{\phi})} (b_t(i_t, \boldsymbol{\phi}_t)). \tag{4.15}$$

The corresponding probability of this task is denoted as $p_{\max}$.

Hence the arbitration factor: $\beta_t = p_{\max}$ is obtained. The robot motion command $\mathbf{g}_t$ can be generated using the state-of-art motion controllers with the recognized task type and the associated task features: $(i_{t,\max}, \boldsymbol{\phi}_{t,\max})$. Then the user input is combined with the robot motion command according to the arbitration factor (see Eq. (3.2) in Chapter 3) to be sent for execution, and in this way the proposed task recognition approach is integrated within the proposed shared autonomy system.

To enhance the flexibility of the proposed system by enabling it to learn new tasks on the fly while being interactively taught by the demonstrator, the next section reports the fast online GMR algorithm and its application in the proposed RBF framework.

4.7 Fast Online and Incremental GMR Algorithm

To provide demonstration online, the human operator firstly inputs the task type and selects the associated semantic target (*e.g.*, a doorway or an object) to demonstrate via a graphical user interface, then the operator drives the robot to complete the demonstration, finally the demonstration data of each task type i are stored in the form of the corresponding task

features (denoted as $\mathbf{D}_i$). This framework is inspired by [78], which favours an incremental adaptation of robot action through demonstration on the fly.

During task recognition, the task feature of each candidate semantic target $\boldsymbol{\phi}_t$ (see Sec. 4.2 of this chapter) is firstly obtained based on the current information of the environment, robot and user command, then the FANN search algorithm presented in [128] is applied to the demonstration dataset of the corresponding task type i to extract a local database $\mathbf{D}_i(\boldsymbol{\phi}_t)$ consisting of the k points closest to $\boldsymbol{\phi}_t$. $\mathbf{D}_i(\boldsymbol{\phi}_t)$ is now used by the EM algorithm to train the GMM[4] with a very few number of the mixture components (denoted as h, which is usually 2 or 3).[5] Finally, an online GMR model is derived accordingly from the trained GMM, which is employed together with $\boldsymbol{\phi}_t$ by the *User Model* (Eq. (4.10)) to predict the likelihood of the corresponding candidate task type and the associated semantic target. An overview of the proposed fast online GMR algorithm and its application in the proposed RBF framework can be seen in Alg. 1.

The FANN algorithm has already shown to be quite efficient in updating the retrieval of high-dimensional computer vision data. Through some off-line tests, it is noticed that it costs just a few milliseconds for the EM algorithm to build a GMM consisting of 2 or 3 mixture components with around 25 points on a standard PC, which is enough for the application of this thesis, and the process speed of the proposed approach will be further investigated in the experimental section. Since the motion pattern of each task type is supposed to be distinctively described with the corresponding task feature (will be evaluated in Chapter 7), h and k need not to be altered when the new demonstration is introduced, thus it can always quickly build the model online and on-demand without heavy re-computations within this framework.

4.8 Summary

This chapter reported a multiple contextual task recognition approach to address the research problem of this thesis. To overcome the challenge of modeling the implicit way the user executes a certain task and incorporate the uncertainty of the user intention, a batch GMR model combined with a RBF was adopted to infer the on-going tasks based on the raw user inputs and the environmental information, *i.e.* the contextual information. The proposed

[4] In order to simplify the computation, a diagonal matrix is employed to implement the covariance matrix of each Gaussian component of the model.

[5] k is typically larger than h multiplied by the state dimensionality of the GMM.

Algorithm 1 Fast Online GMR Algorithm for the User Model of the RBF

for each task type $i \in 1, ..., m$ **do**
  Given incrementally accumulated demonstration dataset $\mathbf{D}_i$
  Given user input $\mathbf{u}_t$ at time t
  for each corresponding candidate semantic target $o \in O$ **do**
    $\boldsymbol{\phi}_t \leftarrow \mathbf{u}_t$ and o following Sec.4.2
    $\mathbf{D}_i(\boldsymbol{\phi}_t) \leftarrow$ by applying FANN algorithm to $\mathbf{D}_i$ with $\boldsymbol{\phi}_t$
    $\text{GMM}_i \leftarrow$ by applying EM algorithm to $\mathbf{D}_i(\boldsymbol{\phi}_t)$
    $\text{GMR}_i \leftarrow \text{GMM}_i$ according to Eq. (4.5)
    GMR_i is used by the User Model of the RBF in Eq. (4.10) to predict the likelihood of i and o at time t
  end for
end for

approach was applied to the problem of operating a mobile robot in the context of four representative task types: doorway crossing, object inspection, wall following and robot docking. A set of simple, compact yet highly distinctive task features were introduced to describe the motion patterns of the human operator executing different contextual task types. Such task features scale with multiple candidate targets and properly encode the user intentions from demonstrations. The task features are adopted by a GMR model combined with a RBF to infer the most probable task the human operator executes across multiple candidates during operation in a batch way. The BIC was employed to select the best GMR model. To realize the RBF, the State Transition Model of the RBF was implemented, and the associated assumptions to simply the implementation were reported. After this, this chapter showed how to integrate the proposed task recognition approach within the proposed shared autonomy system by motion arbitration with the task estimation results.

In the final part of this chapter before conclusion, a fast online GMR algorithm was introduced to render the proposed approach possible to learn and recognize contextual tasks online and incrementally. The demonstration data for different task types were assumed to be stored incrementally in the form of task features when the user introduced new demonstrations on the fly. During the task recognition, the FANN algorithm was firstly employed to obtain a small training dataset which was closest to the current robot state, then a small GMR model was quickly built by the EM algorithm with the dataset. The proposed fast online GMR algorithm can be seamlessly integrated with the proposed RBF to stably and smoothly infer the most probable contextual task the human operator executes across multiple hypotheses.

To successfully apply the proposed contextual task recognition approach in reality, the critical time scale factor σ in the State Transition Model of the proposed RBF has to be

carefully tuned and decided. To avoid this, the contextual task recognition problem is to be formulated as classification problem, and the classification technique is employed with the feature data to obtain the task recognition result. This will be presented in the next chapter.

Chapter 5

Sparse Contextual Task Learning and Classification using SOGP

This chapter formulates the task recognition problem as a classification problem, and reports to use GP, a Bayesian non-parametric model, to learn and classify human motion patterns performing various contextual task types from demonstrations. Due to Gaussian predictive posterior distribution, GP provides superior informative uncertainty estimations in predicting class labels over other state-of-art classification techniques, such as SVM, which is probably the most popular approach on this topic to date. Such outstanding *introspective capability* makes GP an appealing approach for a life-long active learning framework [95, 188, 187], where it is not required that all classes are represented beforehand, but instead the system is able to adapt its knowledge as it enters new environments, and learns during operation by actively selecting unknown data to ask human for annotation and adding to the training data, which is significant to fulfil the envision of a life-long adaptive assistive robotic system. Meanwhile, to keep the model sparse to limit the amount of storage and computation required, full GP is approximated with a state-of-art SOGP [37], to maintain scalability to large datasets without compromising classification performance.

GP is gaining popularity on the topic of learning from demonstration in robotics [10]. Grollman *et al.* [79, 80] apply SOGP regression to learn and replay control policies from demonstration for robot soccer. Paul *et al.* [146] employ a multi-class GP classifier to learn to categorize the semantic information of outdoor scenes from 3D point cloud data with the human labels. Berczi *et al.* [13] use a binary GP classifier to learn to assess the traversability of the terrain from human demonstration. However, both works have to approximate the

predictive posterior distribution obtained from GP with *e.g.* Laplace method, since they use non-Gaussian likelihood model, such as *probit regression* model, to *suppress* the discrete labels into continuous function values. In contrast, this chapter directly takes the labels as the function values, and models the classification problem as a GP regression, resulting into a closed-form solution for both the posterior distribution and the marginal likelihood (*i.e.* the evidence of the model). Although Kapoor *et al.* [95] apply GP regression to classify the objects from images, they achieve it with the full GP, where the amount of storage and computation required can become extremely huge in the long run. To improve this, full GP is approximated with a state-of-art SOGP [37] in this chapter, to maintain scalability to large datasets.

The remainder of this chapter is organized as follows. Sec. 5.1 introduces the task features to encode the motion patterns of the human operator executing various task types from the contextual information. With the defined task features, Sec. 5.2 details the proposed approach. Finally, the chapter is summarized in Sec. 5.3.

5.1 Task Feature

Without loss of generality, four contextual task types are considered to illustrate the proposed approach: Doorway Crossing (DC), Object Inspection (OI), Wall Following (WF) and Object Bypass (OB). To employ data-driven approach to learn the task execution patterns from human demonstrations, these contextual task types are encoded with a set of *task features*.

As introduced in Chapter 4, a task feature[1] $\mathbf{q}$ embodies an instance of a specific task type with the task descriptors obtained from the contextual information, and is calculated in the local coordinate frame fixed on the robot center. Fig. 5.1 illustrates concisely the scenarios of the four task types and their task features respectively.

The user input $\mathbf{u}$ is issued from a mechanical joystick (Logitech F710 wireless gamepad), which consists of translational velocities along x and y axes, and rotational velocity around z axis in the robot's local coordinate frame: $\mathbf{u} = (v_x, v_y, v_\omega)$. Each input channel is normalized to the range of $(-1, 1)$, where the positive sign indicates that, for the translational velocities, the input is along the positive direction of the corresponding axis, and for the rotational velocity, it is in the counter-clockwise direction around z axis.

[1] In this chapter, $\mathbf{q}$ is used instead of $\boldsymbol{\phi}$ to denote a task feature, since $\phi(\cdot)$ is adopted to represent the Cumulative Density Function (CDF) of a standard normal distribution.

The environmental information is encoded using the *intentional target point* to define a task feature, which is extracted from the semantic components of indoor scenarios, *i.e.* doorway, object and wall segment. More specifically, for the task type DC, the center point of a doorway is selected as the intentional target point for the robot to reach or to cross. For OI and OB, the nearest surface point of a segmented object to the robot center during operation is chosen to be the intentional target point for the robot to follow, since the surface points of an object implicitly characterize the shape of the object. Likewise, for WF, the intentional target point refers to the nearest surface point of a wall segment to the robot center during operation. In this chapter, a 2D LRF is employed to perceive the environment, hence the intentional target point is denoted by a two-dimensional coordinate in the local frame of the robot: $\mathbf{s} = (x_\eta, y_\eta)$, but it is straightforward to extend the definition to 3D configuration.

According to the above introductions, a task feature can be expressed as $\mathbf{q} = (\mathbf{s}, \mathbf{u})$. The following part of this chapter is going to present in detail the proposed approach to learn and recognize multiple contextual task types using the SOGP classifier.

5.2 Methodology

During operation, together with user input, each candidate environmental target reports a query task feature $\mathbf{q}_*^{(c)}$, where $c \in \{1, ..., C\}$, and C denotes the number of the task types[2]. By following the one-vs-all classification formulation, it is attempted to infer:

$$p(t_*^{(c)} | \mathbf{q}_*, \mathbf{Q}_L, \mathbf{t}_L^{(c)}), \tag{5.1}$$

where $t_*^{(c)}$ and $\mathbf{t}_L^{(c)}$ indicate the predictive label of $\mathbf{q}_*$ (*i.e.* task type), and the labels of the demonstration data $\mathbf{Q}_L$ respectively, with $t^{(c)} \in \{-1, 1\}^n$ representing the observation vector of binary labels for the task type c with the dimension n.

To formulate the multiple contextual task classification problem using GP, a *latent* function $\mathbf{f}$ is employed to generate a discrete label t for a data point $\mathbf{q}$, which is represented by a GP. Under the Bayesian paradigm, it is obtained:

$$p(t_*^{(c)} | \mathbf{q}_*, \mathbf{Q}_L, \mathbf{t}_L^{(c)}) \propto \int_{\mathbf{f}} p(t_*^{(c)} | \mathbf{f}) p(\mathbf{f} | \mathbf{q}_*, \mathbf{Q}_L, \mathbf{t}_L^{(c)}), \tag{5.2}$$

[2]The $*$ mark used in this subsection denotes that the corresponding variable is obtained during operation as the query value, to distinguish it from the training data.

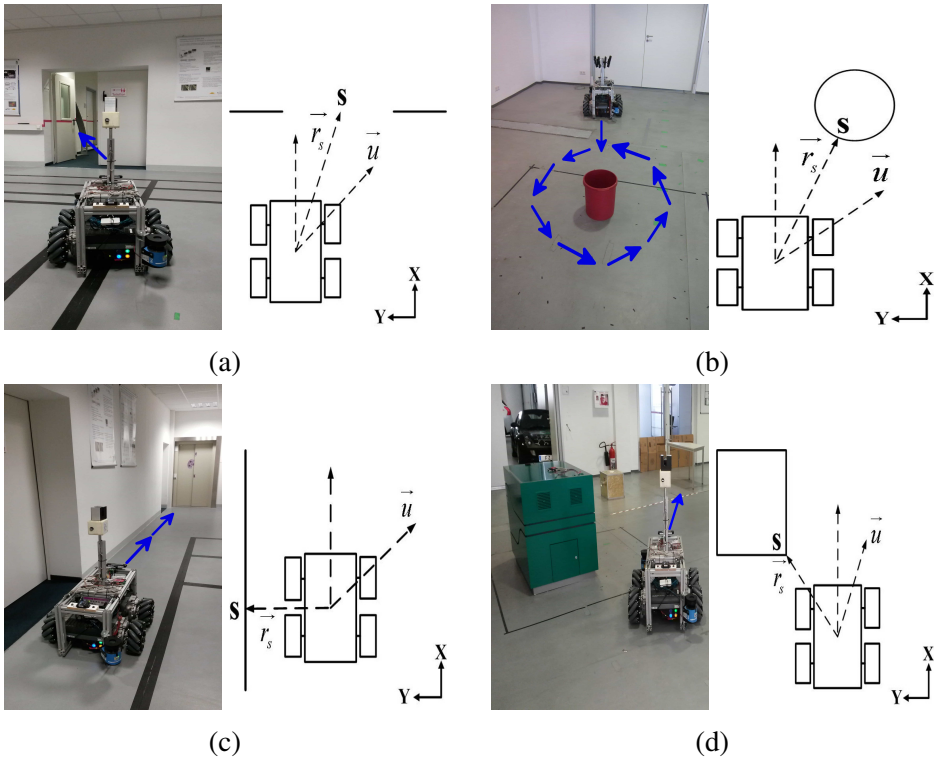


Fig. 5.1 Four contextual task types for learning and recognition, and their corresponding task features ($\mathbf{q}$): (a) Doorway Crossing; (b) Object Inspection; (c) Wall Following; (d) Object Bypass, where $\mathbf{q} = (\mathbf{s}, \mathbf{u})$, $\mathbf{s}$ denotes the intentional target point extracted from the corresponding semantic components, and $\mathbf{u}$ represents the user input issued from a mechanical joystick.

where the latent posterior $p(\mathbf{f}|\mathbf{q}_*, \mathbf{Q}_L, \mathbf{t}_L^{(c)})$ can be written as:

$$p(\mathbf{f}|\mathbf{q}_*, \mathbf{Q}_L, \mathbf{t}_L^{(c)}) \propto p(\mathbf{t}_L^{(c)}|\mathbf{f}) p(\mathbf{f}|\mathbf{q}_*, \mathbf{Q}_L). \tag{5.3}$$

This equation probabilistically combines the smoothness constraints imposed via the GP prior ($p(\mathbf{f}|\mathbf{q}_*, \mathbf{Q}_L)$) and the information provided in the likelihood model ($p(\mathbf{t}_L^{(c)}|\mathbf{f})$). By assuming that the observed values $t_L^{(c)}$ of the latent function $\mathbf{f}$ are corrupted with independent Gaussian noise with variance $\sigma_{(c)}^2$, the Gaussian noise model is applied to compute the likelihood:

$$p(\mathbf{t}_L^{(c)}|\mathbf{f}) = \frac{1}{\sqrt{2\pi\sigma_{(c)}^2}} \exp(-\frac{(\mathbf{t}_L^{(c)} - \mathbf{f})^2}{2\sigma_{(c)}^2}). \tag{5.4}$$

Since this likelihood model is Gaussian, it renders the latent posterior (Eq. (5.3)) Gaussian, and further leads to a closed-form solution for the predictive posterior distribution (Eq. (5.2)) for inference:

$$p(t_*^{(c)}|\mathbf{q}_*, \mathbf{Q}_L, \mathbf{t}_L^{(c)}) = \mathcal{N}(t_*^{(c)}; \mu_*^{(c)}, \sigma_{*(c)}^2). \tag{5.5}$$

Although it is originally developed for regression, Gaussian noise model has proven effective for classification [157, 95], whose performance typically matches the more expensive likelihood models proposed for GP classification, such as *probit* and *logit* models which require approximate inference due to non-Gaussian property.

It is assumed that $\mathbf{f}$ is zero-mean, and it can be obtained:

$$\mu_*^{(c)} = \mathbf{k}_*^T (\mathbf{K} + \sigma_{(c)}^2 \mathbf{I})^{-1} \mathbf{t}_L^{(c)}, \tag{5.6}$$

where $\mathbf{k}_* = \kappa(\mathbf{Q}_L, \mathbf{q}_*)$ and $\mathbf{K} = \kappa(\mathbf{Q}_L, \mathbf{Q}_L)$ denote the kernel values of the training set and the query point, which are computed with the Square Exponential (SE) kernel :

$$\kappa_{\text{SE}}(\mathbf{q}, \mathbf{q}') = \exp(\frac{||\mathbf{q} - \mathbf{q}'||^2}{2l^2}), \tag{5.7}$$

where l is the characteristic length scale (a *hyperparameter* of the model). Hence the final score of the multiple contextual task classifier is achieved by taking the maximal predictive posterior mean across all task types:

$$\mu_*^{\text{mc}} = \max_{c=1...C} \mu_*^{(c)}, \tag{5.8}$$

and returning the corresponding task type[3] c.

As indicated in Eq. (5.5), in addition to the predictive posterior mean $\mu_*^{(c)}$, the posterior variance from the classifier prediction for the task type c can also be computed:

$$\sigma^2_{*(\mathrm{c})} = k_{**} - \mathbf{k}_*^T(\mathbf{K} + \sigma^2_{(c)}\mathbf{I})^{-1}\mathbf{k}_* + \sigma^2_{(c)}, \tag{5.9}$$

where $k_{**} = \kappa_{\mathrm{SE}}(\mathbf{q}_*, \mathbf{q}_*)$. $\mu_*^{(c)}$ is unconstrained continuous value, thus in order to obtain the probability estimation about the predictive label $t_*^{(c)}$ of the query point $\mathbf{q}_*$, it is needed to consider the value $p(t_*^{(c)} \geqslant 0)$, since the sign of $t_*^{(c)}$ decides the binary label: $t_*^{(c)} \rightarrow \pm 1$. GP classification provides with both the posterior mean as well as the posterior variance when making prediction, hence these two values can be combined to compute the probability estimation as:

$$p(t_*^{(c)} \geqslant 0) = \phi(\frac{\mu_*^{(c)}}{\sigma_{*(\mathrm{c})}}), \tag{5.10}$$

where $\phi(\cdot)$ denotes the CDF of a standard normal distribution.

Regarding embedding the SOGP classifier into the proposed shared autonomy system to assist mobile robot teleoperation: from Eq. (5.8) and Eq. (5.10), it is able to obtain the probability of the maximal a posterior task type. This probability computed at each sampling cycle of the robot corresponds to the arbitration factor β_t. Thus the user input and the robot motion command are blended according to β_t to achieve motion assistance within the proposed shared autonomy system, as described in Chapter 3.

Although GP is non-parametric, *i.e.* there is no explicit computation of model parameters in the training step compared with the parametric models, *e.g.* GMM, it is still needed to select appropriate set of the hyperparameters, *e.g.* the characteristic length scale for each input dimension of the SE kernel (Eq. (5.7)), and the noise variance of the model, to improve the performance of the GP classifier. Let Θ_c denote the hyperparameters of the binary classifier for the task type c following the one-vs-all formulation. Θ_c is tuned based on the concept of *Empirical Bayes* [157], where Θ_c are determined to maximize the *marginal likelihood* or the *evidence* of the model: $\hat{\Theta}_c = \arg\max_{\Theta_c} \log(p(\mathbf{t}_L^{(c)}|\mathbf{Q}_L, \Theta_c))$. Thanks to the

[3] If there exist multiple semantic components, the SOGP classifier returns both the task type label and the associated semantic component with the highest score for each query point.

Gaussian noise model, $\log(p(\mathbf{t}_L^{(c)}|\mathbf{Q}_L,\Theta))$ can be computed in closed form:

$$\log(p(\mathbf{t}_L^{(c)}|\mathbf{Q}_L,\Theta_c)) = -\frac{1}{2}(\mathbf{t}_L^{(c)})^T(\sigma_{(c)}^2\mathbf{I}+\mathbf{K})^{-1}\mathbf{t}_L^{(c)} - \frac{1}{2}\log|\sigma_{(c)}^2\mathbf{I}+\mathbf{K}| - \text{Const.} \tag{5.11}$$

In this work, gradient-descent is used to maximize Eq. (5.11) to perform the model selection.

The full GP classification requires $O(kN^2)$ memory and $O(kN^3)$ time to perform inference [157], where k is the number of the classes, and N is the size of the training datasets. These high costs make it infeasible for the long-term robot learning scenario. To limit the amount of storage and computation required, full GP is approximated with SOGP [37], where only a subset of training samples, termed Basis Vectors (BVs), are selected to minimize the Kullback-Leibler (KL)-divergence between the full GP, and one based on the BVs. With suitable enlargement and deletion procedures, the maximum size of BVs (refers to the *capacity* the SOGP) is maintained fixed, hence the required storage and computation resources for processing the GP (for training and inference) are limited, which is significant to scale with large datasets in the long run. Please refer to [37] for extensive technique details of the algorithm.

5.3 Summary

This chapter formulated the contextual task recognition problem as one-vs-all classification problem. The SOGP classifier, a Bayesian non-parametric model, was presented to learn and classify human motion patterns performing various contextual task types from demonstrations, due to its superior introspective capability and sparsity, which favours a lifelong active learning scenario. Four representative contextual task types for mobile robot teleoperation, *i.e.* Doorway Crossing, Object Inspection, Wall Following and Object Bypass, were considered to explain the proposed approach, and the task features were defined to encode the motion patterns of the human operator performing them, which are extracted from the environmental information and the user inputs. To classify the query task feature with the demonstration data during operation, the Gaussian noise model was employed to compute the likelihood of the GP, resulting into the closed-form solutions of the predictive posterior distribution and the model evidence of the GP. By maximizing the closed-form model evidence, the hyper-parameters of the model can be optimized. To measure the similarity between the

query feature data and the training data, the SE kernel was used to compute the covariance matrix of the GP. To keep the model sparse to limit the amount of storage and computation required, the full GP was approximated with the SOGP algorithm, to maintain scalability to large datasets without compromising classification performance.

Chapter 6

Unsupervised Contextual Task Learning and Recognition with DPGMM and SOGP

This chapter will propose a novel approach to learn and recognize multiple contextual tasks in an unsupervised manner. Since the way the human operator performs a contextual task to complete an action primitive is implicit, the supervised learning approaches were employed in the previous two chapters to derive the human motion patterns for various task executions from labeled demonstrations. However, it is difficult for the human expert to manually segment a demonstration into meaningful action primitives for the robot to learn, and in the long run, the manual annotation will be error-prone to limit the applicability of the system, when demonstration data for more and more task types need to be labeled.

To scale with such situation, an unsupervised contextual task learning and recognition approach is reported, consisting of two phases. In the first step, DPGMM is used to cluster the motion patterns of task executions from demonstrations without labels. The major advantage of applying DPGMM for clustering is that the number of possible motion modes (*i.e.* motion clusters) is inferred from the data itself instead of being manually specified *a priori* or determined through model selection, which is required by using *e.g.* GMM and K-Means on this topic. Moreover, it is able to discover both overlaps and distinctions of the task execution patterns through clustering, and this can be used as the knowledge base for interpreting the query patterns later on. Post clustering, a sparse non-parametric Bayesian method, *i.e.* the SOGP classifier (introduced in Chapter 5), is used to classify the query feature point with

the learned motion clusters during operation. Fig. 6.1 illustrates the concept of the proposed approach.

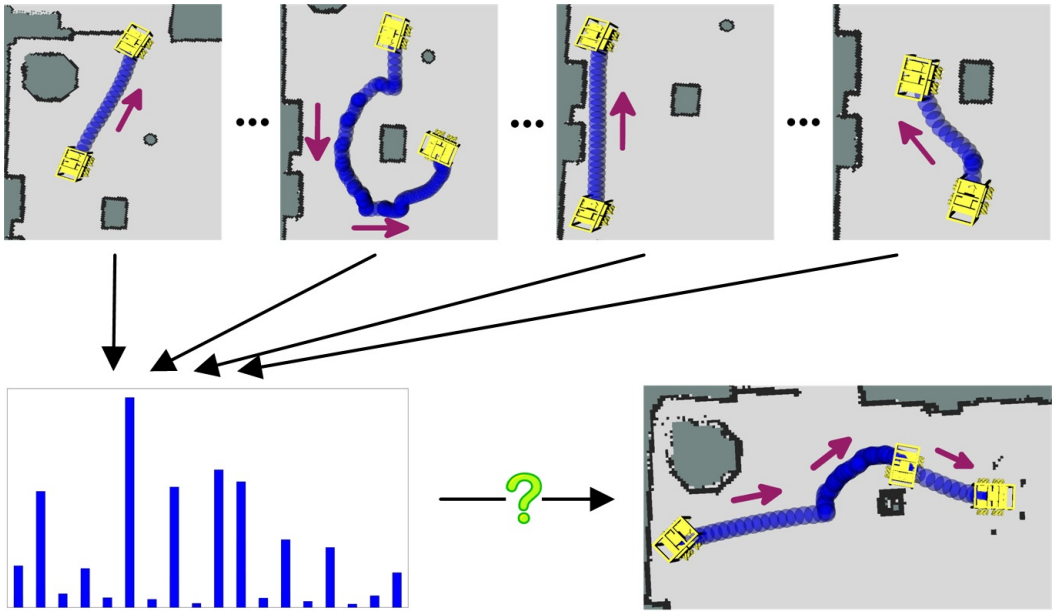


Fig. 6.1 By manually operating a holonomic mobile robot, the demonstration trajectories of the human operator performing various contextual tasks without labels for task types (top) are collected. The demonstrations are transformed into a set of *task features* and provided to DPGMM for clustering to represent the knowledge of the robot regarding the human task execution patterns (bottom left). Then with the SOGP classifier trained using the learned motion clusters, the aim is to interpret the motion patterns of the human operator performing certain tasks during operation (bottom right). Example demonstrations for Doorway Crossing, Object Inspection, Wall Following and Object Bypass are shown respectively (top), with the arrows indicating the movement direction of the robot.

Furthermore, this chapter introduces an algorithm to predict local trajectory the human operator intends to follow in the short term to perform corresponding contextual task in a data-driven manner. Each motion cluster implicitly represents certain motion pattern of the human operator for contextual task execution from demonstration. The proposed algorithm works by utilizing this knowledge from the classified motion cluster, to formulate motion assistance. It applies the proposed fast online GMR approach (please refer to Chapter 4) to classified motion cluster with recognized contextual information. To regulate the prediction accuracy, the *Mahalanobis distance* of each estimated trajectory way point is computed, since each motion cluster is *normally* distributed. By thresholding the distance, the trajectory prediction can be achieved within a predefined tolerance bound regarding the regression outliers. The predicted trajectory for the corresponding contextual task execution is *local*, because the user intention is modeled in this thesis as an *action primitive* to perform a

contextual task, with the aim of providing immediate efficient motion assistance to the human operator *locally*.

This algorithm bridges the task recognition part and the task motion assistance part in the proposed shared autonomy framework, since the predicted trajectory is to be employed as the *reference model* by the state-of-art mobile robot motion controller to generate safe motion commands, which are blended with the user inputs according to the classification confidence, to assist the human operator to carry out the intentional tasks *actively* and *appropriately*. Fig.6.2 briefly illustrates this concept.

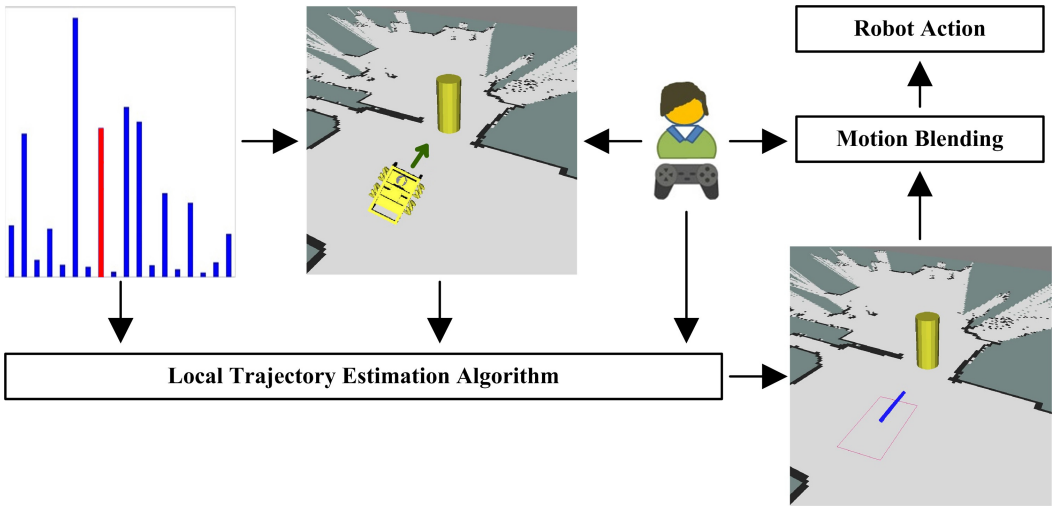


Fig. 6.2 In an example scenario, the robot is being operated to cross a doorway in front (middle), where the target doorway is recognized with high confidence indicated by the bar height. The proposed approach does regression on the classified motion cluster (the red bar) with the user input and the recognized semantic target to predict the local trajectory (in blue) the human operator intends to follow in the next steps for the corresponding task execution (right). The estimated trajectory is to be employed by the state-of-art mobile robot motion controller to generate safe motion command, which is blended with the user input according to the classification confidence to achieve an adaptive task-aware motion assistance.

Regarding the application of unsupervised learning approaches on the topic of robot learning from demonstration [10], Krishnan *et al.* [106] propose to use hierarchical DPGMM to segment the demonstration trajectories from robot-assisted surgeries in an unsupervised manner. Later Murali *et al.* [129] extend this work by employing DPGMM and Deep Learning to leverage video and kinematic data for task-level segmentation. Niekum *et al.* [136, 135] report to apply the Beta-Process Autoregressive Hidden Markov Model (BP-AR-HMM) to segment subtasks from demonstrated finite state machines without labeling them beforehand, aiming to replay complex, multi-step manipulation tasks from demonstrations. This chapter

focuses on clustering and recognizing human motion patterns performing contextual task types from unlabeled demonstrations, where DPGMM is used to achieve clustering without manually indicating the number of possible task types *a priori* or determining it via model selection. With the learned motion clusters, SOGP is employed to classify the query feature point during operation, taking advantage of its superior introspective capability and scalability to large datasets. Regarding the intentional trajectory prediction, although there are state-of-art studies dealing with trajectory inference in the context of assisting wheelchair user by sharing control [40, 41, 87, 88], they assume a set of candidate trajectories to reach certain targets in the environment *beforehand*, and the inference is fixed to these candidates. In contrast, the algorithm proposed here infers the intentional trajectory totally from the human demonstration data with the contextual information, without making any assumptions regarding the property of the trajectory *a priori*.

The remainder of this chapter is organized as follows. Sec. 6.1 describes the task features to encode the human motion patterns for various task executions from the contextual information. Sec. 6.2 introduces the proposed methodology in details. Finally, the chapter is summarized in section 6.3.

6.1 Task Feature

Following the introduction and symbols in 5.1, a task feature $\mathbf{q}$ used in this chapter is built with the intentional target point $\mathbf{s}$ and the user input vector $\mathbf{u}$. In addition to $\mathbf{s}$ and $\mathbf{u}$, the angle θ between the partial user input vector $\mathbf{u}_{xy} = (v_x, v_y)$ and the vector $\mathbf{r}_s$ from the robot center to $\mathbf{s}$ is also computed to be part of a task feature. θ represents the user input direction, hence the movement direction of the robot, relative to $\mathbf{s}$, which bridges two sources of contextual information: environmental perception and user input, and vaguely indicates the user intention for operating the robot regarding the corresponding semantic components. Based on the above introduction, a task feature can be expressed as $\mathbf{q} = (\mathbf{s}, \theta, \mathbf{u})$.

Although just doorway, object and wall are considered to construct task feature here, it is intuitive to obtain task feature from more types of semantic components of the environment, such as docking place, where the intentional target point is the center point of the docking area, and the human target, *e.g.* when the task is to follow a human during telepresence, the intentional target point of which can be the position of the detected human.

The following section of this chapter details the proposed unsupervised approach to learn and recognize multiple contextual tasks with the introduced task features.

6.2 Methodology

6.2.1 Motion Clustering with DPGMM

During demonstration, the task features $\mathbf{q}$ are computed from the target semantic components of the scenario, which are described with no specific task type, *i.e.* unlabeled.

Traditional clustering approaches, such as GMM and K-Means, require that the number of clusters must be specified *a priori*, or chosen via model selection, which is prone to underfitting or overfitting. This severely limits their usefulness when dealing with unstructured data. However, recent work in Bayesian nonparametrics offers a principled way to overcome these limitations.

To discover possible clusters of motion patterns (*i.e.* modes) from the unlabeled demonstration data, the Dirichlet Process (DP) prior is used on the dataset, which allows an infinite collection of modes, and an appropriate number of modes is inferred directly from the data in a fully Bayesian way, without the need for manual specification or model selection [16]. Mathematically, the DP is described with the *stick-breaking* process:

$$
\begin{aligned}
G &\sim DP(\alpha_0 H), \\
G &\triangleq \sum_{k=1}^{\infty} \lambda_k \delta_{\phi_k}, \\
v_k &\sim \mathrm{Beta}(1, \alpha_0), \\
\lambda_k &= v_k \prod_{l=1}^{k-1} (1 - v_l).
\end{aligned}
$$

Where G is an instance of the DP consisting of an infinite set of clusters/mixture components, and λ_k denotes the mixture weight of the component k. Each data item n chooses an assignment according to $\mathrm{w}_n \sim \mathrm{Cat}(\lambda)$, and then samples observations $\mathbf{q}_n \sim F(\phi_{\mathrm{w}_n})$. Since $\mathbf{q}_n$ is multi-dimensional real-valued data, F is taken to be Gaussian. ϕ_k is the data-generating

parameter for the component k, which is drawn from the *normal-Wishart* distribution H with natural parameters ρ_0, facilitating the full-mean, full-covariance analysis.

At the heart of DPGMM is the inference technique, whose goal is to recover stick-breaking proportion v_k and data-generating parameters ϕ_k for each mixture component k, as well as discrete cluster assignment $\mathrm{w} = \{\mathrm{w}_n\}_{n=1}^{N}$ for each observation from the demonstration dataset, which maximizes the joint distribution:

$$p(\mathbf{Q}, \mathbf{w}, \phi, v) = \prod_{n=1}^{N} F(\mathbf{q}_n|\phi_{\mathrm{w}_n})\mathrm{Cat}(\mathrm{w}_n|\lambda(v)) \prod_{k=1}^{\infty} \mathrm{Beta}(v_k|1,\alpha_0)H(\phi_k|\rho_0). \tag{6.1}$$

A variational Bayesian variant inference algorithm, named *Memoized Online Variational Inference* [86], is employed to infer the posterior (Eq. (6.1)). It scales to large yet finite datasets while avoiding noisy gradient steps and learning rates together, and allows non-local optimization by developing principled birth and merge moves in the online setting. For more details regarding the algorithm, please refer to [86].

Each learned motion cluster is considered as an *action primitive*, it can be used with the estimated semantic target to interpret the motion patterns of the human operator performing certain contextual tasks in the form of trajectory the human operator intends to execute (will be detailed in the last part of this section). Hence the robot can efficiently help with the task execution by assisting the human operator to safely follow the intentional trajectory in remote. From this perspective, it is supposed to classify the query task features obtained from multiple candidate semantic components to the learned motion clusters, in order to find the most probable cluster and the associated semantic component during operation. SOGP classifier is employed to achieve this, and it will be covered in the following part.

6.2.2 Motion Classification with SOGP

To recognize which motion patterns (including the associated semantic targets) the human operator executes, the SOGP classifier is employed to classify the query task features to the learned motion clusters. How to adapt it to the application of this chapter will be briefly introduced in this subsection. For more technical details regarding the SOGP classifier, please refer to Chapter 5.

Specifically, by following the one-vs-all formulation, it is attempted to infer:

$$p(t_*^{(c)}|\mathbf{q}_*, \mathbf{Q}_L, \mathbf{t}_L^{(c)}), \tag{6.2}$$

where $t_*^{(c)}$ and $\mathbf{t}_L^{(c)}$ indicate the predictive label of a query task feature $\mathbf{q}_*$ and the labels of the motion cluster data $\mathbf{Q}_L$ respectively, with $t^{(c)} \in \{-1, 1\}^n$ representing the observation vector of binary labels for cluster $c \in \{1, ..., M\}$, where M is the number of the clusters. With this problem formulation, the classification can then be achieved by following Eq. (5.6) - (5.8).

Although the motion clusters are not semantically labeled, they implicitly represent the human motion patterns for performing certain contextual tasks from demonstrations. Meanwhile, to formulate motion assistance for task execution, a reference model is needed by the state-of-art mobile robot motion controller to command the robot to carry out the desired task efficiently. Since it is *non-trivial* to *manually* devise reference model for each candidate task, especially for those which are not yet considered, a data-driven approach will be introduced in the following part to solve this issue, in order to bridge the task recognition module and the task motion assistance module within the proposed shared autonomy framework.

6.2.3 Trajectory Prediction with Fast Online GMR

Although the fast online GMR approach was initially presented in Chapter 4, it was aimed for contextual task recognition in a supervised learning framework instead of trajectory estimation by regression. In this part, the approach is applied to predict local trajectory the human operator intends to follow in the short term for corresponding task execution by regression with associated contextual information, to utilize the motion pattern knowledge from the classified motion cluster. Moreover, the fast online GMR approach is extended by considering the regulation of the prediction accuracy to suit the application. Such predicted trajectory is to be used as the reference model by the state-of-art mobile robot motion controller (*e.g.* work done in [160]) to formulate the corresponding motion assistance by generating efficient motion commands.

During operation, a query task feature is computed with each candidate semantic component in the environment. By applying the proposed SOGP classifier to classify each query task feature into the learned motion clusters, the most probable query task feature is obtained together with the assigned motion cluster $c_{\max}$ and the associated semantic component, by

taking the maximal predictive posterior mean across all clusters and query task features (Eq. (5.8)).

For the purpose of presentational completeness, the fast online GMR approach is briefly summarized as follows. Given the learned training dataset $\mathbf{D}$ and the query value of the regressors $\mathbf{x}_t$ at time t, the FANN algorithm is firstly applied to $\mathbf{D}$ with $\mathbf{r}_t$ to obtain a *small* and *local* database $\mathbf{D}(\mathbf{r}_t)$ consisting of the k points closest to $\mathbf{r}_t$. $\mathbf{D}(\mathbf{r}_t)$ is then employed by the EM algorithm to train a GMM with a very few number of the mixture components (usually 2 or 3). Finally, an online GMR model is derived accordingly from the trained GMM to predict the value of the dependent variables $\mathbf{y}_t$. The approach is listed in Alg. 2.

Algorithm 2 Fast Online GMR Algorithm

1: Given learned trained dataset $\mathbf{D}$
2: Given query value of the regressors $\mathbf{x}_t$ at time t
3: $\mathbf{D}(\mathbf{r}_t) \leftarrow$ by applying FANN algorithm to $\mathbf{D}$ with $\mathbf{x}_t$
4: $\text{GMM}_{\mathbf{r}_t} \leftarrow$ by applying EM algorithm to $\mathbf{D}(\mathbf{r}_t)$
5: $\text{GMR}_{\mathbf{r}_t} \leftarrow \text{GMM}_{\mathbf{r}_t}$
6: $\mathbf{y}_t \leftarrow \text{GMR}_{\mathbf{r}_t}(\mathbf{r}_t)$

With the fast online GMR approach, the proposed local trajectory prediction algorithm is listed in Alg. 3. In each iteration of the prediction, the query intentional target point $\mathbf{s}_*$ is firstly computed with the associated semantic component and the robot pose (*i.e.* the way point along the trajectory). Then the fast online GMR approach is applied to $c_{\max}$ with $\mathbf{s}_*$, to obtain the predictive motion command for the robot (*i.e.* the user input $\mathbf{u}_*$). The next way point the human operator intends to drive the robot to reach (*i.e.* the robot pose in the next iteration) is computed by applying the Robot Kinematic Model (RKM) with $\mathbf{u}_*$, the simulation time Δ_t and the current pose of the robot.

Although such extrapolation procedure can be iterated forever, to achieve certain prediction accuracy, an approach is proposed to terminate it reasonably. At each iteration, the corresponding task feature $\mathbf{q}_{\text{estimated}}$ is computed with the query intentional target point $\mathbf{s}_*$ and the estimated user input $\mathbf{u}_*$. Since each motion cluster, which consists of task features $\mathbf{q}$, is obtained with the proposed DPGMM method, it is assumed to be *normally* distributed, thus its mean vector and covariance matrix can be easily computed beforehand. Then the Mahalanobis distance for $\mathbf{q}_{\text{estimated}}$ with respect to the assigned motion cluster $c_{\max}$ at each iteration can be obtained. Use $\bar{\mathbf{q}}_{c_{\max}}$ and $\mathbf{\Sigma}_{c_{\max}}$ to denote the mean vector and the covariance matrix of the assigned motion cluster $c_{\max}$ respectively, the Mahalanobis distance d_{mh} for

$\mathbf{q}_{\text{estimated}}$ is computed according to its definition as:

$$d_{\text{mh}} = \sqrt{(\mathbf{q}_{\text{estimated}} - \bar{\mathbf{q}}_{c_{\max}})^T \cdot \boldsymbol{\Sigma}_{c_{\max}}^{-1} \cdot (\mathbf{q}_{\text{estimated}} - \bar{\mathbf{q}}_{c_{\max}})}. \tag{6.3}$$

Mahalanobis distance [38] is a measure of the distance between a query point and a distribution, which is *unitless* and *scale-invariant*, and takes into account the correlations of the data set. It is a usual measurement to detect outliers in regressions. Hence by thresholding d_{mh}, the trajectory prediction iteration can be terminated within a predefined tolerance bound regarding the regression outliers. The choice of the fast online GMR approach for regression is advantageous because of its outstanding performance over the batch GMR algorithm (will be demonstrated with evaluations in Chapter 7) and simplicity for hyper-parameters tuning.

Algorithm 3 Trajectory Prediction with Recognized Contextual Information

Given the classified motion cluster with the highest score $c_{\max}$
Given the semantic component associated with the most probable query task feature
Given the current robot pose $\mathbf{p}_r$
Given the RKM$(\mathbf{v}_t, \Delta_t, \mathbf{p}_t)$
initialize $\mathbf{p}_{\text{est}} \leftarrow \mathbf{p}_r$
initialize Traj_Pred $\leftarrow \{\}$
initialize $d_{\text{mh}} \leftarrow 0$
while $d_{\text{mh}} < d_{\text{mh\_thres}}$ **do**
    $\mathbf{s}_* \leftarrow$ with the associated semantic component and $\mathbf{p}_{\text{est}}$
    $\mathbf{u}_* = (v_x^*, v_y^*, v_\omega^*) \leftarrow$ by applying the fast online GMR model to $c_{\max}$ with $\mathbf{s}_*$
    $\mathbf{q}_{\text{estimated}} \leftarrow \mathbf{s}_*$ and $\mathbf{u}_*$
    $d_{\text{mh}} \leftarrow$ Eq. (6.3) with $\mathbf{q}_{\text{estimated}}$
    $\mathbf{p}_t \leftarrow \mathbf{p}_{\text{est}}$
    $\mathbf{p}_{\text{est}} \leftarrow \text{RKM}(\mathbf{u}_*, \Delta_t, \mathbf{p}_t)$
    Traj_Pred $\leftarrow$ Traj_Pred $\cup \{\mathbf{p}_{\text{est}}\}$
end while
return Traj_Pred

The confidence of the predicted trajectory is represented by the confidence of the most probable query task feature where the iteration begins, which is computed with Eq. (5.5). Thus the robotic motion commands, which are generated by the state-of-art motion controller using the predicted trajectory as the reference model, are blended with the human inputs according to the classification confidence (please refer to Eq. (3.2) in Chapter 3). In this way, a contextual-task-aware shared autonomy between human and robot is realized during operation.

6.3 Summary

Following the the strategy formalized in Chapter 3, this chapter reported an unsupervised approach to learn and recognize human motion patterns performing various contextual task types from unlabeled demonstrations, to facilitate autonomy sharing to assist mobile robot teleoperation. The motion patterns were described with a set of intuitive, compact and salient task features. The DPGMM was employed to cluster the motion patterns based on the task feature data, where the number of potential motion components was inferred from the data itself instead of being manually specified *a priori* or estimated through model selection. Moreover, both overlaps and distinctions of the task execution patterns can be discovered through clustering, which is used as a knowledge base for interpreting the query patterns later on. Post clustering, the SOGP classifier was used to recognize which motion pattern the human operator executes during operation, taking advantage of its outstanding confidence estimation when making predictions and scalability to large datasets.

By applying the proposed fast online GMR approach to the classified motion cluster, an algorithm was introduced in this chapter to predict local trajectory the human operator intends to follow in the short term to perform corresponding task. In this way, the motion assistance is formulated in a data-driven manner instead of being devised manually. To calculate and threshold the Mahalanobis distance computed with each estimated way point, the trajectory prediction can be achieved within a predefined tolerance bound regarding the regression outliers. The predicted trajectory is to be employed as the reference model by the state-of-art mobile robot motion controller to generate safe motion commands, which are blended with the user inputs according to the classification confidence. In this way, the proposed contextual-task-aware shared autonomy framework is realized.

Chapter 7

Evaluation

The previous chapters presented the proposed framework for sharing autonomy to assist mobile robot teleoperation, and formalized the research problem to be tackled in this thesis. To address the problem, a set of multiple contextual task learning and recognition approaches were introduced, which serve as the major contributions of this thesis. This chapter extensively evaluates these approaches, since they are the cornerstone of the proposed shared autonomy system. The focus is on assessing the task recognition performance of these approaches in comparison with the baseline approaches in both simulation and with real robot, in order to verify the scalability and effectiveness of the proposed approaches in contextual task learning and estimation. The performance of the proposed shared autonomy system in assisting the human operator with remote task execution in motion is also evaluated, to demonstrate the merit of the proposed contributions in practice.

This chapter is structured as follows. Sec. 7.1 introduces the general settings of the evaluations conducted in the following parts of this chapter. Sec. 7.2 evaluates the proposed RBF framework (see Chapter 4) on multiple contextual task recognition with the GMR models learned from human demonstrations in both simulations and real experiments. In the same section, to evaluate the proposed fast online GMR algorithm, its task recognition performance is compared with that of the batch GMR model, and its real-time property is also tested.

In Sec. 7.3, the classification performance and the scalability to large datasets of the proposed SOGP classifier (see Chapter 5) are investigated over the baseline classification methods with real data. Meanwhile, the introspective capability of the proposed SOGP

classifier in detecting potentially unmodeled classes in the test data is also examined. This capability is critical for an active life-long learning scenario.

Sec. 7.4 reports the motion clustering and classification results with the proposed DPGMM-SOGP approach (see Chapter 6) in real experiments, in order to evaluate its effectiveness in learning and recognizing various contextual tasks in an unsupervised manner. The same section also evaluates the performance of the proposed algorithm in estimating the local trajectory the human operator intends to follow in the short term for corresponding task executions.

Sec. 7.5 evaluates the proposed reactive shared autonomy approach (please refer to Chapter 3) with a small user study in simulation, in order to demonstrate its effectiveness in safely assisting the human operator with teleoperating a simulated quadrotor to navigate in an indoor scenario.

In Sec. 7.6, a carefully controlled and repeatable user study with a significant number of human test participants is conducted, in order to evaluate the performance of the proposed contextual-task aware adaptive shared autonomy system over the baseline approaches in assisting mobile robot teleoperation in a cluttered indoor scenario, and verify the necessity and effectiveness of the most important concept proposed and implemented in this thesis: providing *proactive* motion assistance by recognizing the motion intention and the associated semantic components in a probabilistic way. Finally, this chapter is closed with summary in Sec. 7.7.

7.1 Experiment Settings

The software of the proposed approaches is implemented within the Robot Operating System (ROS) framework due to its modularity. For simulation, the Gazebo Simulator was employed. It is a 3D simulator with physics engines, and provides convenient interfaces to the ROS framework. For real experiments, a holonomic mobile robot named Adaptive Shared Autonomy Platform (ASAP) was used. As shown in Fig. 7.1, the robot consists of two Segway-Omni 50 platforms, and employs two SICK LRFs for perceiving environments. The on-board PC of the robot has an Intel Core i7 M620 Dual-Core processor, 4GB of RAM, and is running Ubuntu 14.04, ROS (version: Indigo), Open Source Computer Vision library (OpenCV), and the Point Cloud Library (PCL). A Logitech F710 wireless gamepad was employed in both simulation and real experiments to issue user inputs and control the robot.

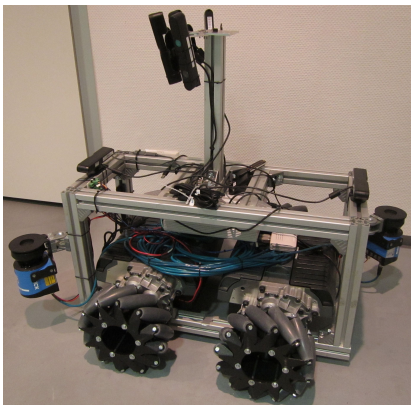


Fig. 7.1 The test mobile platform ASAP

To facilitate providing demonstrations and evaluating the proposed contextual task recognition approaches without the loss of generality of them, the registered information was used throughout their evaluations in this chapter: the maps of the involved test scenarios were built beforehand with the state-of-art Simultaneous Localization and Mapping (SLAM) algorithm implementation [105] within ROS framework, since the robot adopts 2D LRFs for perceiving environments, and it has already been open-sourced in the ROS framework. Then the global positions of the robot in these maps were obtained with the state-of-art localization technique available in the ROS framework.

Regarding the registered information of the candidate semantic components of the scenarios, *e.g.* the global positions of doorways, objects and wall segments, there is unfortunately no open-source solution for extracting and registering them during the SLAM process, neither is such simultaneous mapping and semantic component localization the research topic of this thesis. Therefore, the methods described in Appendix 8.2 were used to annotate the semantic components of the scenarios afterwards. The proposed annotation approach treats the constructed 2D occupancy grid map as an image, and applies OpenCV functions to process such *map image* to propose the candidate semantic components for further annotations, to reduce the manual workload to annotate a scenario from scratch. For more technique details regarding these algorithms to process the map image, please refer to Appendix 8.2. Please notice, the proposed contextual task learning and recognition approaches do not assume to only work with the registered information.

Meanwhile, to evaluate the contextual task recognition performance of the proposed approaches from learning the human demonstrations, the raw data were collected from the human operator manually driving the robot to execute various predefined (yet unknown to

the robot) contextual tasks with random initial poses, and the recorded data were processed in a *post-experimental stage*, *i.e.* to analyse the data *off-line*.

During operation, each semantic component was associated with a candidate task to be estimated in the proposed approaches, and in real applications, there usually exists multiple candidate tasks, *e.g.* multiple doorways to cross, or multiple objects for inspection. Accordingly, the proposed contextual task recognition approaches need to be capable of robustly handling multiple hypotheses when inferring the on-going task the human operator is performing. Consequently, a correct recognition of the contextual task refers to the correct estimations of both the task type and its associated semantic components when evaluating the task recognition performance of the proposed approaches in the following parts of this chapter.

The next section focuses on evaluating the RBF and GMR based multiple contextual task recognition framework proposed in Chapter 4 in both simulations and real experiments.

7.2 Evaluation of RBF Framework combined with GMR on Multiple Contextual Task Recognition

To comprehensively evaluate the performance of the proposed RBF framework combined with batch and online GMR models on multiple contextual task recognition, this section starts with the experiments conducted in simulation with a sensor-equipped quadrotor, then reports the evaluations with the holonomoic mobile robot ASAP in real scenarios. In the end, the proposed fast online GMR algorithm is evaluated with the proposed RBF framework in a variety of real tests.

7.2.1 Evaluation of RBF with Batch GMR in Simulation

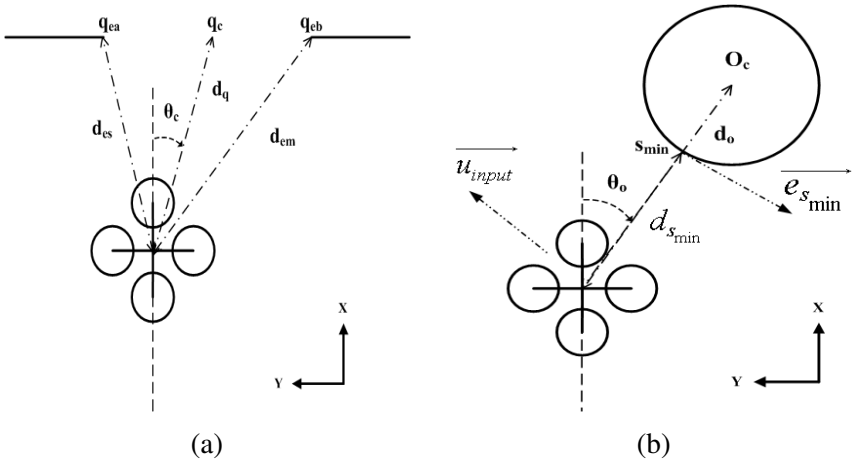
A simulated quadrotor [125] was employed within the Gazebo Simulator to collect data for further analysis. Apart from the basic sensors, *e.g.* an Inertial Measurement Unit (IMU) and a sonar height sensor, the simulated quadrotor is also equipped with a 2D LRF, and a 2D camera facing forward. The robot was controlled by the human operator with a joystick. Two task types were considered, *i.e.* Doorway Crossing and Object Inspection, to evaluate the proposed approach in simulation.

As explained in Sec. 7.1, the maps of the involved test scenarios were built with the state-of-art SLAM algorithm, and their semantic components, *e.g.* the doorways and the object segments, were extracted and registered before the tests.

For simulation tests, more complex task features are adopted for describing the two task types compared with those introduced in Chapter 4. To smooth the statements made in this subsection, the employed task features for the two task types are concisely introduced in the following part respectively.

7.2.1.1 Task Features for the Two Task Types adopted in Simulation

If not specified otherwise, the term *distance* used in the following part refers to the *Euclidean distance*.



(a) (b)

Fig. 7.2 The task feature for each of the two task types: (a) Doorway Crossing; (b) Object Inspection, an circular object is used here as an example, note that o_c and $s_{\min}$ are collinear under this situation.

A. Doorway Crossing. The position of the center point of a doorway in the local coordinate frame of the robot is denoted as $q_c = (x_c, y_c)$, and q_{ea}, q_{eb} represent the two end points of the doorway. The task feature for this type of task contains (Fig. 7.2(a)):

1. d_q: the distance between the robot and q_c.
2. θ_q: the angle between the robot heading and q_c.
3. d_{es}: the smaller distance between the robot and q_{ea}, q_{eb}.

4. d_{em}: the larger distance between the robot and q_{ea}, q_{eb}.

B. Object Inspection. The related object for this task type is processed in 2D configuration. For each candidate object which is segmented, its surface points are maintained:

$$S = \{s | s \in ((x_1, y_1), ..., (x_n, y_n))\}. \tag{7.1}$$

The center point $o_c = (x_c, y_c)$ of the object is calculated as:

$$o_c = \frac{1}{n} \sum_{i=1}^{n} s_i, s_i \in S. \tag{7.2}$$

For each surface point $s \in S$, the unit tangential vector $\mathbf{e}$ at it can be decided by firstly finding a straight line with s and its neighbouring surface points (four neighbouring points were applied in the implementation) using Random Sample Consensus Algorithm (RANSAC) [53], then taking the unit direction vector of the straight line as $\mathbf{e}$. A task feature for this task type consists of (Fig. 7.2(b)):

1. d_o: the distance between the robot and the o_c.
2. θ_o: the angle between the robot heading and the o_c.
3. $d_{s_{\min}}$: the distance between the closest surface point of the object $s_{\min}$ and the robot.
4. $v_{s_{\min}}$: which is determined as:

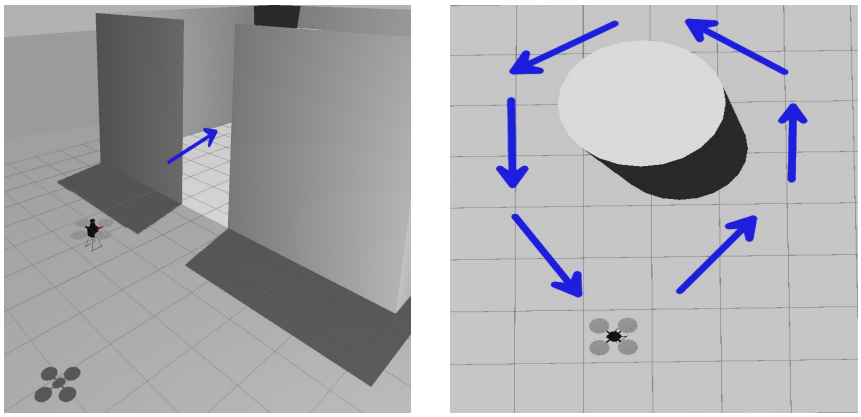
$$v_{\text{smin}} = |\mathbf{e}_{s_{\min}} \cdot \mathbf{u}|, \tag{7.3}$$

where $\mathbf{u}$ is the normalized vector of the user input, $\mathbf{e}_{s_{\min}}$ is the unit tangential vector at the closest surface point $s_{\min}$. In practice, to obtain a smooth result and compensate for the possible sharp change of the direction of the unit tangential vector, $\mathbf{e}_{s_{\min}}$ was computed by averaging the unit tangential vectors at $s_{\min}$ and its neighbouring points (four neighboring points were used in the implementation).

7.2.1.2 Training Data Gathering and Model Selection

For each task type, there is a corresponding environment scenario to provide demonstrations, which is depicted in Fig. 7.7. The robot was controlled to execute a specific task type.

For each task type, the demonstrations were done for eight times starting and ending with different poses. The task features and the user inputs were collected to assemble into a single training dataset for the corresponding task type: $(i, \boldsymbol{\phi}, \mathbf{u})$. A joint GMM is learned offline for each task type, and the user model GMR_i (Eq. (4.9) in Chapter 4) for the task type i is derived accordingly, which is then applied online to update the RBF (Eq. (4.4) in Chapter 4).



(a) (b)

Fig. 7.3 The demonstration scenarios for two task types: (a) Doorway Crossing; (b) Object Inspection, one possible demonstration is illustrated with the blue arrows.

For model selection, a set of candidate GMMs are computed with up to ten components using the demonstration data, and the model with the best score are chosen according to Eq. (4.11) in Chapter 4. Finally, the optimal model for the task type Doorway Crossing has five components, while the one for the Object Inspection consists of four components.

7.2.1.3 Testing in More Complex Scenarios

To verify the effectiveness of the learned models, several more complex test scenarios are applied. They contain more candidate targets compared to only one target in each of the demonstration conditions. Due to its intuitiveness and simplicity, an exponential based confidence function similar to [24] is adopted as the baseline method to compare with the proposed approach (although in [24] the authors do not provide the evaluation results regarding the intention recognition):

$$c = e^{-k_\alpha \alpha \cdot k_\beta \beta}. \tag{7.4}$$

For the task type[1] Doorway Crossing, $\alpha = d_q$, $\beta = |\theta_q|$; for Object Inspection, $\alpha = d_{s_{\min}}$, $\beta = |1 - v_{s_{\min}}|$.

Five volunteers operated the robot to execute the task with the pre-specified goal(s) (unknown to the robot) for two times in each of the test scenarios starting and ending at random poses. To facilitate a clear intention expression, the test participants were provided with the global view of the simulated world during the tests as shown in the following scenario captures. The trajectory, the corresponding task features and user inputs along were recorded during each test. Each user test trajectory is taken, and each of the two estimation methods is applied with the corresponding task features and user inputs at every point along, in a post-experimental stage. The percent of time (*success rate*) is measured across each trajectory when the correct goal(s) is identified, obtaining ten test results for each of the scenarios. The Wilcoxon matched-pairs signed rand test is chosen for the comparison of the test data pairs without normal distribution.

After being tuned according to the experiment data with reasonable efforts, the scaling factor k_α of the baseline method is set to 0.05 for the doorway crossing task, and 0.3 for the object inspection task, while k_β is set to 2.0 and 0.2 respectively.

A. Doorway Crossing with Two Doorways.

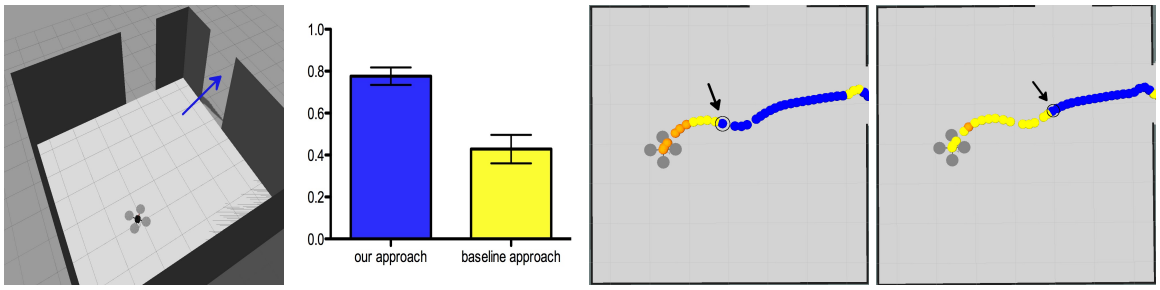
This test scenario consists of two doorways, and the robot was controlled to cross one of them (the goal doorway), as illustrated in Fig. 7.4(a).

Fig. 7.4(b) shows the means and standard errors of the success rate for the proposed approach vs. the baseline method. The Wilcoxon matched-pairs signed rank test confirms that the success rate of the proposed approach is significantly higher than that of the baseline approach ($P < 0.0039$).

Fig. 7.4(c) and Fig. 7.4(d) compare the two methods on one of the trajectories in this scenario, colored in each case based on if the inference is correct (blue spheres), incorrect (orange spheres) or unknown (yellow spheres). As can be seen from the figure, the proposed approach is able to predict the goal earlier than the baseline method.

B. Object Inspection with Two Candidate Objects.

[1] d_q, θ_q, $d_{s_{\min}}$ and $v_{s_{\min}}$ are defined in the task features of the corresponding task types in the subsection 7.2.1.1 of this chapter.



(a) (b) (c) (d)

Fig. 7.4 The doorway crossing test with two possible doorways: (a) The scenario of the test, the goal doorway is marked with the blue arrow; (b) Percent of time the estimation is successful; (c) The recognition result of applying the proposed approach to one test trajectory, correct estimations are depicted by blue spheres, incorrect situations by orange spheres and unknown ones by yellow spheres respectively, the arrow indicates the point where the successful recognition starts; (d) The recognition result of applying the baseline approach to the same trajectory.

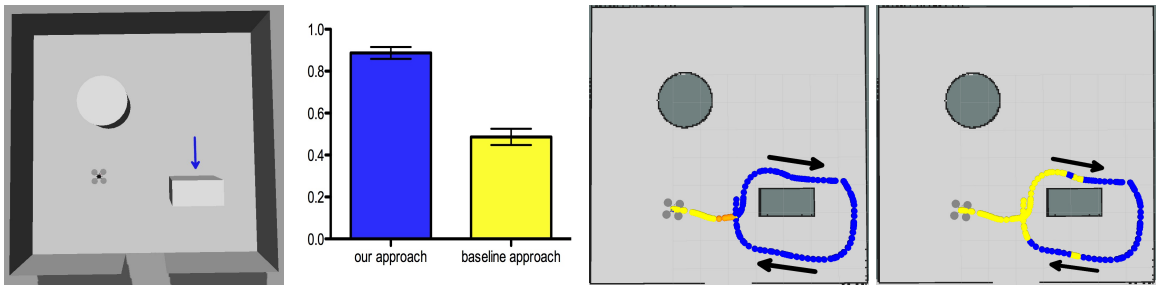
Similarly, in this test scenario there are two candidate objects (a circle and a rectangle), and the robot was operated to move around one of them (the goal object), as shown in Fig. 7.5(a).

Fig. 7.5(b) shows the means and standard errors of the success rate for each of the two approaches. The Wilcoxon matched-pairs signed rank test verifies that the correct estimation rate of the proposed approach is significantly higher than that of the baseline approach ($P < 0.002$).

Fig. 7.5(c) and Fig. 7.5(d) illustrate one of the trajectories in this scenario, colored in the same way as the previous test scenario. As can be viewed, the proposed method can recognize the correct goal earlier and much more stably than the baseline method.

C. Task with both Doorway Crossing and Object Inspection.

This test scenario aims to investigate whether the proposed approach can properly estimate the user intention, when both task types: *Doorway Crossing* and *Object Inspection* are possible. The robot was operated to firstly move around the object, then cross the doorway, as depicted in Fig. 7.6(a). Therefore, the groundtruth of this test is *Object Inspection* at first, then *Doorway Crossing*, and the test data collected along each trial trajectory were marked in this way for the success rate calculation.



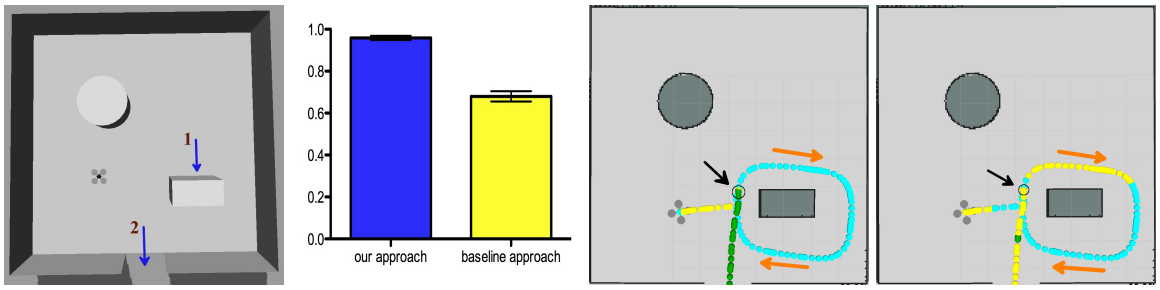
(a) (b) (c) (d)

Fig. 7.5 The object inspection test with two candidate objects: (a) The scenario of the test, the goal object is indicated with the blue arrow; (b) Percent of time the estimation is correct; (c) The recognition result of applying the proposed approach to one test trajectory, correct inferences are depicted by blue spheres, incorrect estimations by orange spheres and unknown ones by yellow spheres respectively, the arrows display the movement direction of the robot in this test trajectory; (d) The recognition result of applying the baseline approach to the same trajectory.

Fig. 7.6(b) shows the means and standard errors of the success rate for the proposed approach vs. the baseline method. The Wilcoxon matched-pairs signed rank test acknowledges that the success rate of the proposed approach is significantly higher than that of the baseline approach ($P < 0.002$).

Fig. 7.6(c) and Fig. 7.6(d) display one of the test trajectories in this scenario, colored in each case based on whether a certain task is clearly recognized (*Object Inspection*: the cyan spheres; *Doorway Crossing*: the green spheres), or not (yellow spheres). The black arrows illustrate the start of the task *Doorway Crossing*, while the orange ones display the movement direction of the robot. As indicated in the figures, the proposed method considerably outperforms the baseline method along this trajectory.

Considering the above results and discussions, and the fact that the baseline method requires different parameters for different task types resulting from a fine tuning of the scaling factors, the proposed approach can obtain an effective model with a relative small number of demonstrations (eight per configuration), and provide the estimations in a unified framework. Moreover, the results verify that the proposed approach fulfills the design goal for the system. This further motivate the author of this thesis with high confidence to implement and evaluate it on a real robot platform, which is reported in the next subsection.



(a) (b) (c) (d)

Fig. 7.6 The combined test of the doorway crossing and the object inspection: (a) The scenario of the test, the sequence of the task execution is marked with the blue arrows and the numbers beside; (b) Percent of time the inference is correct; (c) The recognition result of applying the proposed approach to one test trajectory, a cyan sphere represents the detection of the object inspection task, a green sphere stands for the doorway crossing task, while a yellow one means that no task is identified, the black arrow shows the split point of the two tasks in this trajectory, while the orange ones display the movement direction; (d) The recognition result of applying the baseline approach to the same trajectory.

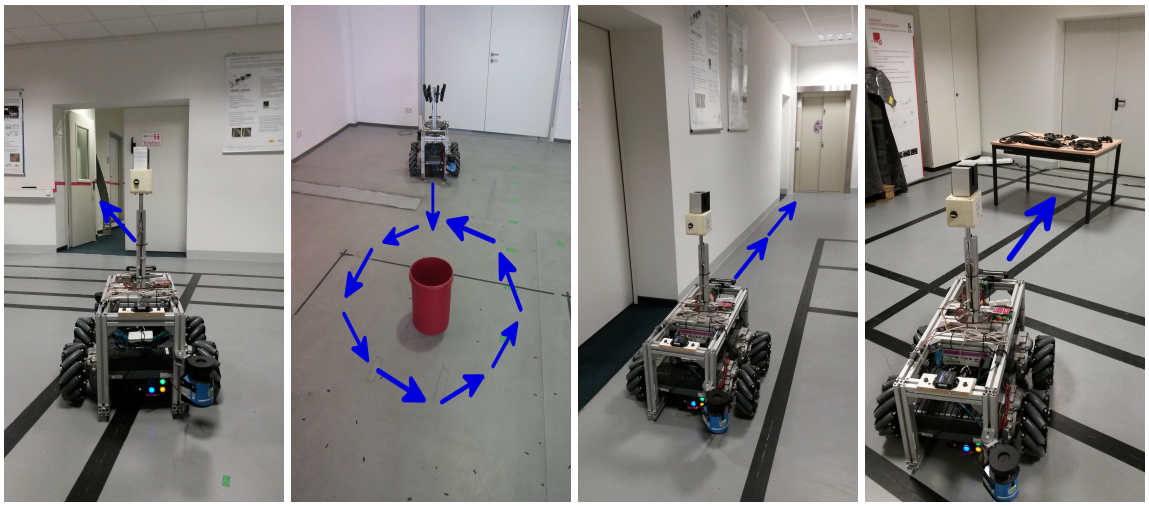
7.2.2 Evaluation of RBF with Batch GMR using Real Robot

This subsection evaluates the proposed approach in a set of real scenarios with the holonomic mobile robot ASAP introduced in Sec. 7.1. In addition to Doorway Crossing and Object Inspection, two more task types were considered, *i.e.* Wall Following and Robot Docking, for evaluation. More compact task features are adopted to describe the four task types, as presented in Chapter 4. The involved scenarios were mapped and the corresponding semantic components were extracted beforehand.

Similarly to simulation tests, for each task type, there is a corresponding environment scenario for providing demonstrations, which is depicted in Fig. 7.7. The robot was controlled to demonstrate each task type for five times starting from different poses. The models are trained with the batch mode as introduced in Chapter 4.

To verify the learned models, the proposed approach was evaluated in a different scenario, as illustrated in Fig. 7.14. It has one doorway, one table for docking, three objects for inspection[2] and two walls around the scenario (another side of the scenario is a long slope), as indicated by the corresponding arrows respectively. It is a cluttered scenario and poses

[2]Note that these objects are different from the one (a dustbin) used to demonstrate the object inspection task type in both shape and size.



(a) (b) (c) (d)

Fig. 7.7 The demonstration scenarios for the four contextual task types: (a) Doorway Crossing; (b) Object Inspection; (c) Wall Following; (d) Robot Docking. One possible demonstration for each task type is illustrated with the blue arrows respectively.

challenge for estimating the intentional task of the human operator, since the recognitions of the four contextual task types can confuse each other in this scenario.

(a) (b)

Fig. 7.8 The scenario used for the evaluation, which is captured from two angles. The blue arrows indicate the candidate objects for inspection, the green one shows the table for docking, the red one represents the doorway, and the violet ones denote the orientations of the two walls. Note that another side of the scenario is a long slope.

Similar to the subsection 7.2.1, an exponential based reactive confidence function introduced in [24] was employed as the baseline method to compare with the proposed approach:

$$C = e^{-k_\alpha \alpha \cdot k_\beta \beta}. \tag{7.5}$$

For[3] *Doorway Crossing*, $\alpha = d_q$, $\beta = |\psi_q|$; for *Object Inspection*, $\alpha = d_{s_{\min}}$, $\beta = |\psi_{s_{\min}}|$; for *Wall Following*[4], $\alpha = d_{w_{\min}}$, $\beta = ||\psi_{w_{\min}}| - 90°|$; for *Robot Docking*, $\alpha = d_o$, $\beta = |\psi_o|$. The baseline approach is tuned experimentally with the reasonable efforts to achieve its optimal performance.

During the tests, each of eight volunteers operated the robot to execute one task with the pre-specified goals (unknown to the robot) in the test scenario (Fig. 7.14) starting from random poses. The required task for each test participant was a sequence of two different task types, *e.g.*, to firstly inspect an object, then cross the doorway, or to firstly follow a wall, then dock into the table, *etc*. The trajectory, the corresponding task features and user inputs along were recorded during each trial, and finally eight groups of the test data had been recorded.

Each user test trajectory is taken, and each of the two estimation methods is applied with the corresponding task features and user inputs at every point along, in a post-experimental stage. The following values across each test trajectory are measured as the criteria to comprehensively judge the recognition performance of different approaches: 1) *Success Rate*: percent of time when the correct goal(s) is identified; 2) *Error Rate*: percent of time when the wrong goal(s) is inferred; 3) *Null Rate*: percent of time when no target(s) is recognized. The Wilcoxon matched-pairs signed rank test is employed to compare the success rates (the most important criterion) of different approaches. It is appropriate for the comparison of the test data without normal distribution.

Fig. 7.15 presents respectively the means and the standard deviations of the measured three criterion values for the proposed approach against the baseline approach over all the test trajectories. These results prove that the proposed approach has achieved the stable and good performance to recognize all the required tasks in a cluttered scenario with multiple candidates. The Wilcoxon matched-pairs signed rank test confirms that the success rates of the proposed approach are significantly higher than those of the baseline approach (Eq. (7.5)) in all the test trajectories ($P < 0.0006$).

In the following parts of this subsection, to provide a qualitative comparison of the performances of the two methods, the representative recognition results along four test trajectories (which include all the four task types) by applying the two methods will be illustrated respectively as examples, due to the page limits.

[3] d_q, ψ_q, $d_{s_{\min}}$, $\psi_{s_{\min}}$, $d_{w_{\min}}$, $\psi_{w_{\min}}$, d_o and ψ_o are defined in the task features of the corresponding task types in Chapter 4 respectively.

[4] When this task type is executed, $|\psi_{w_{\min}}|$ is supposed to be around 90°.

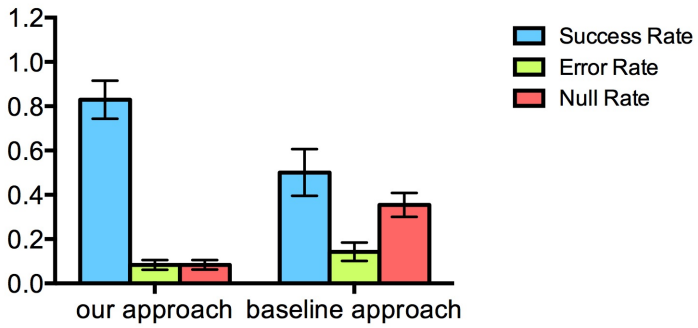


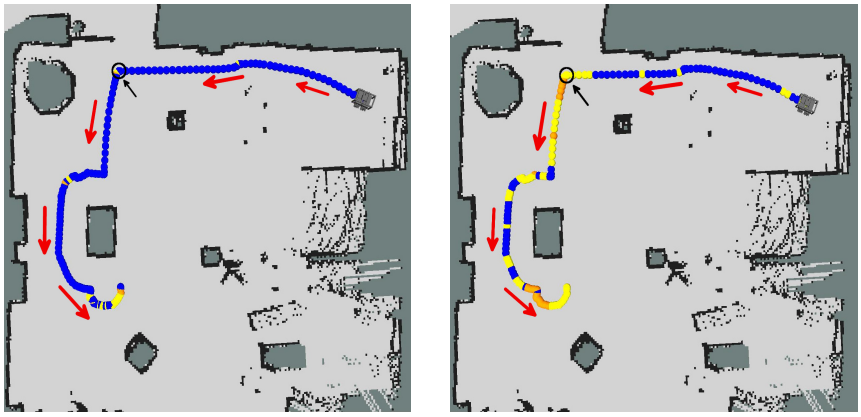
Fig. 7.9 This figure illustrates the means and the standard deviations of the three criterion values for the proposed approach against the baseline approach across all the test trajectories.

As shown in Fig. 7.10(a) and Fig. 7.10(b), the two methods are compared on one test trajectory, where the robot was operated to firstly inspect an object, then cross the doorway, colored in each case based on if the estimation was correct (blue spheres), incorrect (orange spheres) or unknown (yellow spheres). The black arrows indicate the start point of the task type *Doorway Crossing*, while the red ones show the movement direction of the robot. As can be viewed from the figures, the proposed approach considerably outperforms the baseline approach along this trajectory.

(a) (b)

Fig. 7.10 The recognition results of applying the proposed approach (a) and the baseline approach (b) to one test trajectory, where the robot was moved to firstly inspect an object, then cross the doorway. Correct estimations are illustrated by blue spheres, incorrect recognitions by orange spheres and unknown ones by yellow spheres respectively. The black arrows show the split point of the two task types in this trajectory, while the red ones denote the movement direction of the robot along this trajectory.

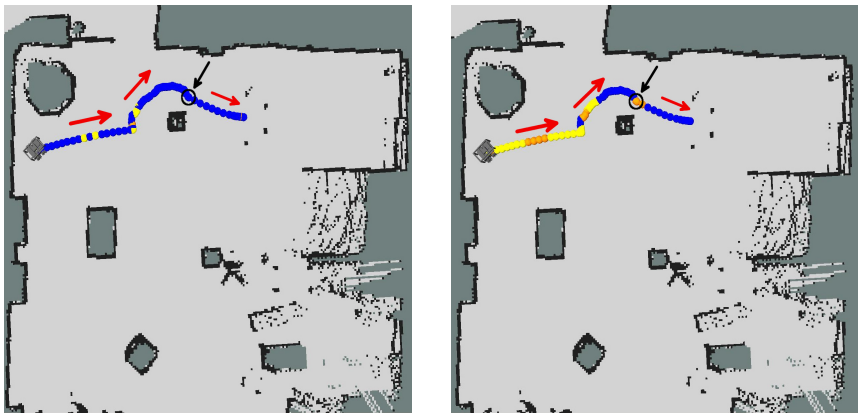
Fig. 7.11(a) and Fig. 7.11(b) illustrate respectively the recognition results along one test trajectory by applying the two approaches, where the robot was controlled to firstly follow wall, then inspect an object, colored in the same way as Fig. 7.10. As shown in the two figures, the proposed method can correctly recognize the intentions of the human operator with much shorter prediction time and better stability than the baseline method along this trajectory, although at the very end of the trajectory the estimation performance of the proposed approach declines due to the strong motion drifts of the robot when moving sideways on the uneven floor of the scenario. Meanwhile, note that during the object inspection phase, the inspection direction of the robot (in counter-clockwise direction) is different from that depicted in Fig. 7.10 (in clockwise direction) when the robot was being moved around the object, and the proposed approach can still correctly infer the action.



(a) (b)

Fig. 7.11 The recognition results of applying the proposed approach (a) and the baseline approach (b) to one test trajectory, where the robot was operated to firstly follow wall, then inspect an object. The color coding is the same as Fig. 7.10.

Fig. 7.12(a) and Fig. 7.12(b) display respectively the estimation results along one test trajectory by employing the two methods, where the robot was operated to firstly inspect an object, then dock into the table, colored in the same way as Fig. 7.10. As can be viewed from the figures, the proposed approach considerably surpasses the baseline approach with shorter prediction time and better stability along this trajectory. Note that, although the sizes and shapes of the objects in Fig. 7.10 and Fig. 7.11 are different from the one (a dustbin) used to demonstrate the object inspection task type (Fig. 7.7(b)), the proposed approach still recognizes the action correctly and stably. This verifies the definition of the task features for this task type.

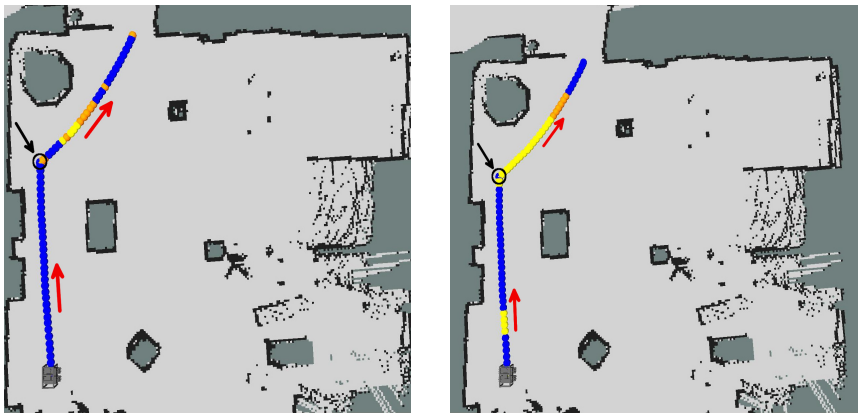


(a) (b)

Fig. 7.12 The recognition results of applying the proposed approach (a) and the baseline approach (b) to one test trajectory, where the robot was moved to firstly inspect an object, then dock into the table. The color coding is the same as Fig. 7.10.

Fig. 7.13(a) and Fig. 7.13(b) show respectively the recognition results along one test trajectory by applying the two approaches, where the robot was controlled to firstly follow wall, then cross the doorway, colored in the same way as Fig. 7.10. Although the proposed approach performs better than the baseline approach along this trajectory, it does not recognize the doorway crossing task type (*i.e.* the second part of the trajectory) as stably as along the trajectory depicted in Fig. 7.10(a). The reason for this issue is that, in this part of the trajectory, the robot was being moved along a big object which is so close to the wall that it is perceived as part of the wall, thus the proposed approach recognizes such movement more as wall following than doorway crossing. When the robot was being moved away from the object, the proposed approach makes the correct estimations again. Meanwhile, note that during the wall following phase, the robot followed the wall on the left side, while in Fig. 7.11(a) it followed the wall on the right side, and the proposed approach can correctly recognize the action in both situations. This validates the definition of the task features for this task type.

In summary, the above results verify the effectiveness of the proposed RBF framework combined with the batch GMR model for multiple contextual task recognition. It is trained with a relative small number of demonstrations (five per task type), and considerably outperforms the baseline approach which requires a fine tuning of the parameters for recognizing each task type, in a cluttered and challenging indoor scenario with a variety of real tests done by different test participants.



(a) (b)

Fig. 7.13 The recognition results of applying the proposed approach (a) and the baseline approach (b) to one test trajectory, where the robot was moved to firstly follow wall, then cross the doorway. The color coding is the same as Fig. 7.10.

The next subsection focuses on investigating the performance of the proposed RBF framework combined with the proposed fast online GMR algorithm on multiple contextual task recognition.

7.2.3 Evaluation of RBF with Fast Online GMR

The ASAP was employed to continue the experiments in this subsection. Three task types were considered, *i.e.* Doorway Crossing, Object Inspection and Wall Following in this subsection for evaluation. The same sets of task features as the previous subsection are used to describe them. The involved scenarios were mapped and the corresponding semantic components were extracted in advance.

The scenario used to evaluate the proposed approach is illustrated in Fig. 7.14. It contains two candidate doorways, three candidate objects with different shapes and sizes, and two walls around the scenario (another side of the scenario is a long slope), as indicated by the corresponding arrows respectively. The scenario is cluttered and hence challenging for estimating the intentional task of the human operator, especially considering that in this scenario the inferences of the three task types can potentially confuse each other.

During the experiments, each of nine volunteers operated the robot to execute four tasks with the pre-specified goal(s) (unknown to the robot) in the scenario (Fig. 7.14) starting from random poses in such sequence: 1) to cross a doorway; 2) to inspect an object (in

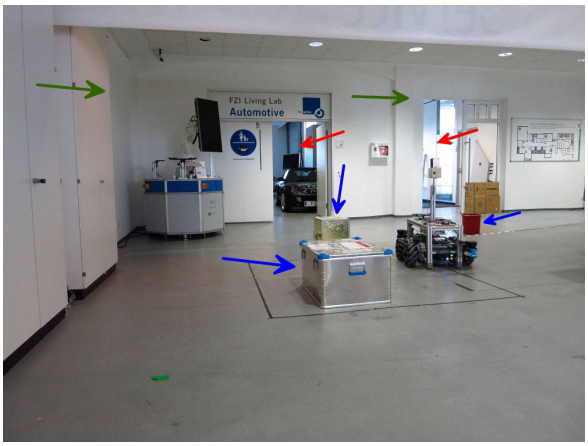


Fig. 7.14 The scenario used for the evaluation. The blue arrows indicate the candidate objects for inspection, the red one represents the doorways, and the green ones denote the orientations of the two walls. Note that another side of the scenario is a long slope.

random directions[5]); 3) to follow the wall (on either left or right side); 4) to firstly follow the wall (on either side), then inspect an object (in random directions), finally cross a doorway. The trajectories, the corresponding task features and the user inputs along were recorded during each trial, and finally for each of the four tasks, nine groups of data had been recorded respectively. The average speed of the robot during the experiments was approximately $0.3m/s$. To compare with the proposed approach, the same method used in 7.2.2 is employed as the baseline approach, and its parameters are tuned manually with reasonable efforts based on the experiment data to achieve the optimal performance.

One group of the recorded data (till four groups) is randomly selected from each of the first three tasks respectively as the demonstration data for the corresponding three task types (all the recorded data of the fourth task are used for testing). Each of the remaining data from the first three tasks after the demonstration selection and all the recorded data from the fourth task are employed for testing, and each of the estimation methods is applied with the corresponding task features at every point along, in a post-experimental stage[6]. The same criteria as in the subsection 7.2.2 are adopted to comprehensively judge the recognition performance of different methods, *i.e. Success Rate*, *Error Rate* and *Null Rate*. During each incrementation, the *Selection-Testing* procedure is repeated three times.

Fig. 7.15 presents respectively the means and the standard deviations of the measured three criterion values for the proposed approach[7] against the baseline approach when execut-

[5] In clockwise or counter-clockwise direction after approaching the target object
[6] Before each test, the system is reset to eliminate the memory of the previous test.
[7] With the parameters: $h = 3$, $k = 24$, please refer to Chapter 4 for their definitions.

ing the required four tasks, as the demonstration is incrementally added (from one to four). As is shown, the proposed approach is able to incorporate new demonstrations incrementally and correctly. These results also prove that, the proposed approach has achieved the stable and good performance to recognize all the contextual tasks in a cluttered scenario with multiple candidates, especially when more demonstrations are added, its performance has no trend of degradation. The Wilcoxon matched-pairs signed rank test is employed to compare the success rates (the most important criterion) across different approaches during each incrementation. It is appropriate for the comparison of the test data without normal distribution. It confirms that the success rates of the proposed approach are significantly higher than those of the baseline approach across all the four tasks during each incrementation ($P < 0.001$). Moreover, as is indicated in Fig. 7.15, by adopting only one demonstration, the proposed approach has already obtained the high recognition scores for the four tasks. This affirms the effectiveness of the task features for the corresponding task types, and also means that the proposed approach is able to be quickly deployed, which is an important and valuable advantage of the proposed approach for real applications.

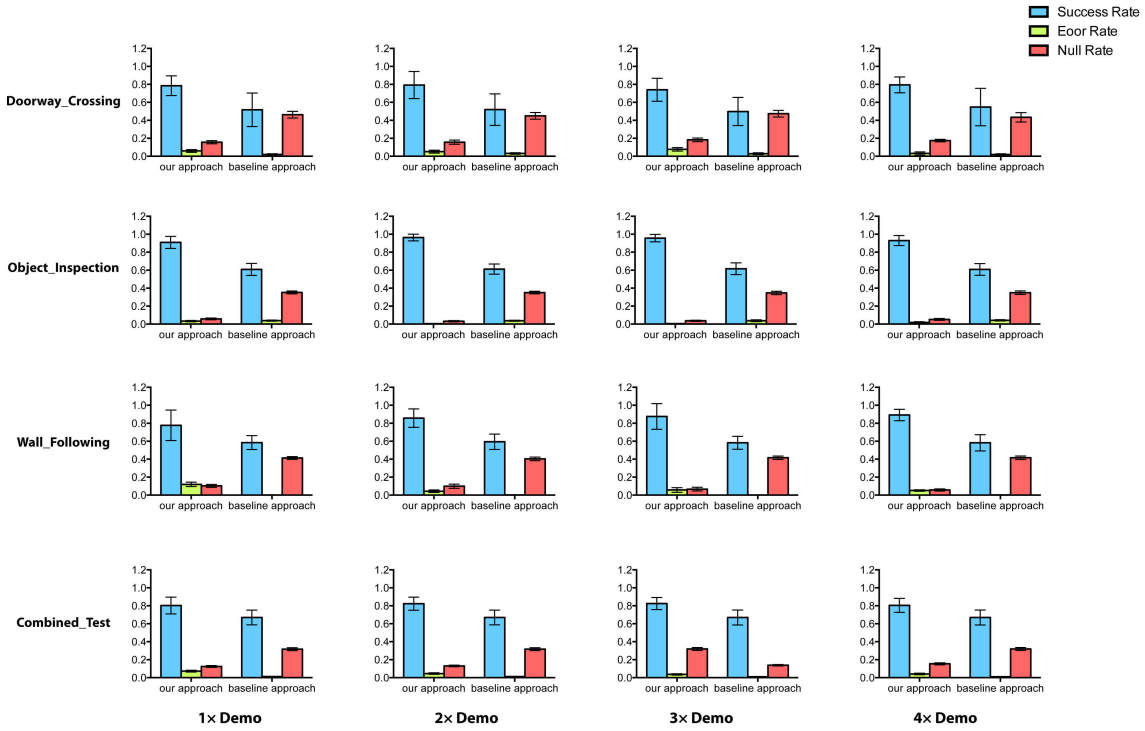


Fig. 7.15 This figure illustrates respectively the means and the standard deviations of the three criterion values for the proposed approach against the baseline approach when executing the required four tasks, as new demonstrations were added incrementally (from one to four). During each incrementation, the procedure of randomly selecting demonstration and testing was repeated three times.

Table 7.1 Average data process speed of the proposed online GMR algorithm during each incrementation, together with the average numbers of the demonstration samples for all the task types and the test samples over the four tasks respectively

Iteration	Average Numbers of Demonstration Samples	Average Numbers of Test Samples	Average Calculation Time per Test Point (ms)	Maximal Calculation Time of All Test Points (ms)
1x Demo	1893	27272	2.17618	18.3762
2x Demo	3266	25899	2.30633	16.4556
3x Demo	5347	23818	2.20749	16.6908
4x Demo	6833	22332	2.47746	17.1376

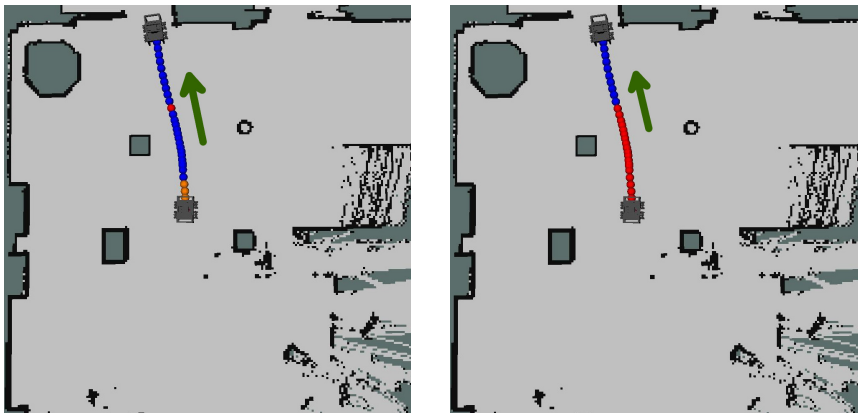
To investigate the process speed of the proposed approach, Table 7.1 reports during each incrementation respectively the average numbers of the demonstration samples for all the task types and the test samples over the four tasks, and the corresponding average calculation time[8] per test point and the maximal calculation time spent on all test points of the proposed online GMR approach, since the Selection-Testing procedure is repeated three times within each iteration. As is indicated, the proposed approach is able to be built and filter the candidate tasks quite fast (averagely less than 2.5ms per test point, maximally below 18.5ms on one test point). This means that it is highly suitable for real time applications.

In the following parts of this subsection, to provide a qualitative comparison of the performances of the two methods, the representative recognition results along one of the test trajectories by applying the two methods after adopting four demonstrations will be illustrated for each of the four tasks respectively as examples, due to the stable performance of the proposed approach and the page limits.

As is shown in Fig. 7.16(a) and Fig. 7.16(b), the two methods are compared on one of the test trajectories in the task of crossing a doorway, colored in each case based on if the estimation was correct (blue spheres), incorrect (orange spheres) or unknown (red spheres). Along the trajectory, the green arrows indicate the movement direction of the robot. As can be viewed from the figure, the proposed approach is able to predict the goal much earlier and more stably than the baseline approach, and it is not confused by the two objects nearby during the recognition of this task, although the trajectory passes between them.

Fig. 7.17(a) and Fig. 7.17(b) illustrate respectively the representative recognition results along one of the test trajectories by applying the two approaches in the task of inspecting the

[8]The calculation time refers to the total computation time spent on each test point to firstly train *small GMR models* around the point (primarily resulting from the FANN and the EM algorithms) then to do the recognitions for all the task types.



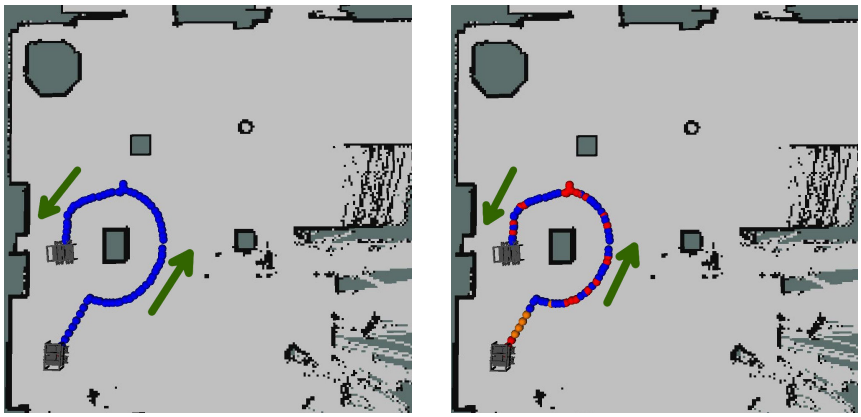
(a) (b)

Fig. 7.16 The representative recognition results of applying the proposed approach (a) and the baseline approach (b) to one test trajectory in the task of doorway crossing with four demonstrations. Correct estimations are depicted by blue spheres, incorrect recognitions by orange spheres and unknown ones by red spheres respectively. The green arrows denote the movement direction of the robot along this trajectory.

target object, colored in the same way as Fig. 7.16. The proposed method can recognize the correct action of the human operator with shorter time and better stability than the baseline method along this trajectory, even when the recorded data are not smooth because of the robot motion and the localization errors.

Fig. 7.18(a) and Fig.7.18(b) depict respectively the representative estimation results along one test trajectory by applying the two approaches, where the robot was controlled to follow the wall, colored in the same way as Fig. 7.16. As shown in the two figures, the proposed method can correctly recognize the intention of the human operator with much higher stability than the baseline method along this trajectory.

Fig. 7.19(a) and Fig. 7.19(b) display respectively the representative recognition results along one of the test trajectories by employing the two methods in the task of firstly following the wall, then inspecting an object, finally crossing a doorway, colored in the same way as Fig. 7.16. Along the trajectory, the black arrows indicate the start points of the following two task types: Object Inspection and Doorway Crossing. As is shown in the two figures, the proposed approach considerably outperforms the baseline approach along this trajectory. Note that during the wall following phase, the robot followed the wall on the left side, while in Fig. 7.18 it followed the wall on the right side, and the proposed approach can correctly recognize the action in both situations. Meanwhile, during the object inspection phase, although the inspection direction of the robot (in clockwise direction) was different from that depicted in



(a) (b)

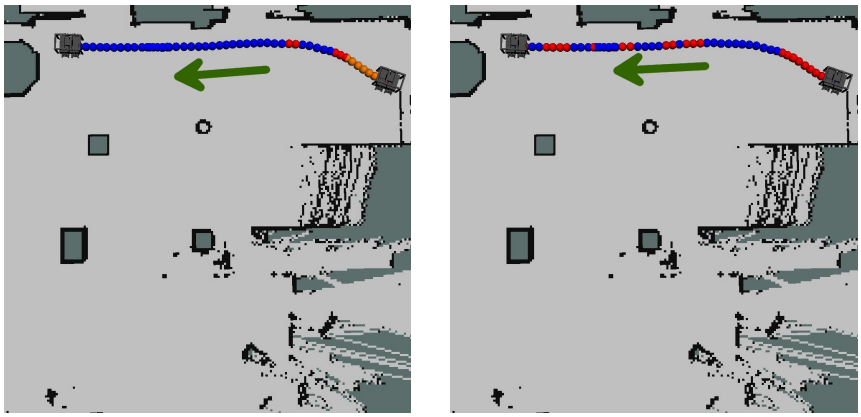
Fig. 7.17 The representative recognition results of applying the proposed approach (a) and the baseline approach (b) to one test trajectory in the task of object inspection after four demonstrations were added. The color coding is the same as Fig. 7.16.

Fig. 7.17 (in counter-clockwise direction) when the robot was moved around the object, and the shapes and the sizes of the two target objects are different from each other, the proposed approach can still correctly infer this action in the two situations.

To further prove the proposed approach, the proposed online fast GMR approach[9] is compared against the batch mode based GMR approach[10] over the dataset of the fourth task assignment, which contains nine test trajectories, since it consists of all the three task types. All the datasets of the first three tasks (nine trajectories per each task) are used as the demonstration data for the corresponding task types. The above introduced three criterion values are measured across each test trajectory to compare the recognition performance of the two methods. Fig. 7.20 presents respectively the means and the standard deviations of the three criterion values for the proposed approach against the batch mode based GMR approach. It is confirmed with the Wilcoxon matched-pairs signed rank test that, the success rates of the proposed approach are significantly higher than those of the batch mode based GMR approach to recognize all the contextual tasks in this test ($P < 0.001$). It indicates that, compared with the global model, the proposed online local model better encodes the motion patterns of the human operator executing various contextual task types from the corresponding demonstrations.

[9] With the parameters: $h = 3$, $k = 24$

[10] Within the batch mode, the GMM is trained and selected in the way introduced in Chapter 4. The trained GMM uses the full covariance matrix.



(a) (b)

Fig. 7.18 The recognition results of applying the proposed approach (a) and the baseline approach (b) to one test trajectory, where the robot was operated to follow the wall, after four demonstrations were added. The color coding is the same as Fig. 7.16.

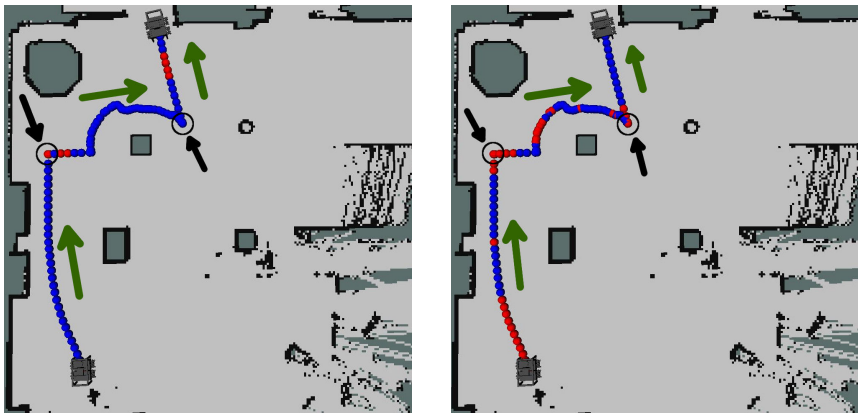
In summary, a set of real tests are performed in a cluttered indoor scenario, and the results verify the effectiveness of the proposed fast online and incremental approach for learning and recognizing multiple contextual tasks the human operator executes.

The next section presents the evaluations of the SOGP classifier proposed in Chapter 5 on classifying multiple contextual tasks within a set of real tests.

7.3 Evaluation of SOGP on Classifying Multiple Contextual Tasks

To evaluate the proposed SOGP classifier, the ASAP was still employed to collect data for post-experimental analysis. Four task types were considered, *i.e.* Doorway Crossing (DC), Object Inspection (OI), Wall Following (WF) and Object Bypass (OB), for evaluations, and the task features introduced in Chapter 5 are adopted to describe them. The involved scenario was mapped and the corresponding semantic components were annotated in advance, which is shown in Fig. 7.26.

Seven volunteers operated the robot manually in the scenario, to execute each of four task types for two times from random initial positions in random order respectively. Finally, 14 datasets (trajectories) were collected for each task type for further analysis, containing totally



(a) (b)

Fig. 7.19 The representative recognition results of applying the proposed approach (a) and the baseline approach (b) to one test trajectory in the task of firstly following the wall, then inspecting an object, finally crossing a doorway, after four demonstrations were adopted. Along the trajectory, the black arrows show the split points of the three tasks. The other color coding is the same as Fig. 7.16.

2512, 9841, 5571 and 3416 way points[11] for Doorway Crossing, Object Inspection, Wall Following and Object Bypass respectively. Please notice, except Doorway Crossing, there are two different movement directions for the other three task types recorded in the datasets: Object Inspection (to inspect the target object in either clockwise or counter-clockwise direction regarding the object), Wall Following (to follow the wall by either left or right side of the robot) and Object Bypass (pass the target object by either left or right side of the robot). This comprehensively represents the features of these task types and increases the complexity of classification. The average speed of the robot during the evaluations was approximately $0.3m/s$.

Apart from the SOGP classifier, SVM with SE kernel, K Nearest Neighbour (KNN) and Random Forest (RF) are employed as the baseline classifiers for comparison. The LIBSVM [33] implementation is used for SVM, and OpenCV implementation for RF. Across all the following evaluations, the capacity of SOGP is chosen to be 500. In the following experiments, two statements are to be verified. First, even being sparsified from the full GP, the SOGP classifier outperforms the other classifiers in classification accuracy on real data, and is capable of real-time update thanks to the adopted sparsity approximation (the subsection 7.3.1). Second, the SOGP classifier provides considerably better uncertainty estimation about the resulting class labels than the other classifiers, when being trained with few classes (in the following evaluations two instead of four), demonstrating that the SOGP

[11] The data sampling rate was 5Hz, which was the sampling rate of the joystick.

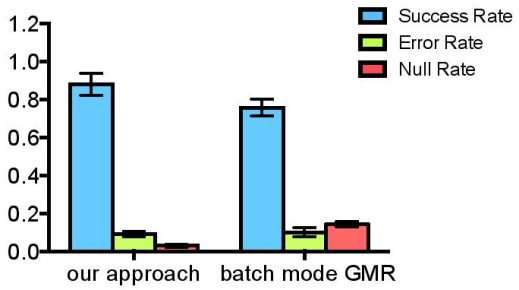


Fig. 7.20 This figure illustrates respectively the means and the standard deviations of the three criterion values for the proposed approach against the batch mode based GMR approach when estimating all the three contextual task types over the nine test trajectories. There were nine demonstrations per each task type, which were adopted from the above introduced experiments.

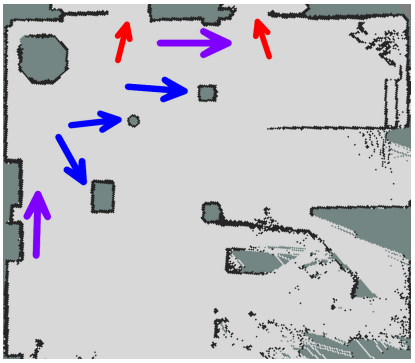


Fig. 7.21 The map of the scenario used for evaluation, where there are two candidate doorways denoted by red arrows, three objects with different sizes and shapes indicated by blue arrows, and two wall segments shown by violet arrows.

classifier is much more useful for detecting unknown classes in the training set, which is promising for a life-long active learning system (the subsection 7.3.2).

7.3.1 Evaluation of Classification Accuracy And Process Speed

This subsection evaluates the classification performance of the four classifiers on the collected datasets about their accuracies, which is the basic requirement for a classifier. Each classifier is evaluated over the collected datasets with a five-fold cross validation, in a post-experimental stage. Please notice, since each candidate semantic component of the scenario reports a corresponding task feature $\mathbf{q}$, a correct classification refers to the correct recognitions of both the task type label c and the associated semantic target in the following evaluations.

Table 7.2 Confusion matrix resulting from the SOGP classifier with the original test results in numbers.

		DC	OI	WF	OB
Groundtruth	DC	1550	102	286	84
	OI	44	7226	2	
	WF	105	18	5020	
	OB	20	475	85	2329

Assigned Labels

In each fold, during training[12], for SVM, KNN and RF, a grid-search is done on their respective parameters over reasonable sets; while for SOGP, before approximation, the (locally) optimized hyperparameters are obtained by maximizing the log evidence of the full GP (Eq. (5.11) in Chapter 5) via gradient-descent.

Table 7.2 shows the confusion matrix of the SOGP classifier with the original test results in numbers after the five-fold cross validation. Fig. 7.22 depicts the confusion matrix of SOGP classifier, where values are normalized along rows. Thus the diagonal values represent per-class recall. Fig. 7.23 visualizes its confusion matrix with values normalized vertically, then the diagonal values represent precision. Overall, the SOGP classifier exhibits good precision performance in the evaluations.

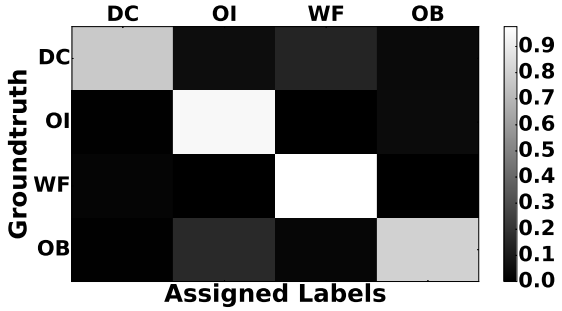


Fig. 7.22 Confusion matrix (normalized) resulting from the SOGP classifier. Recall values appear along the diagonal.

The confusion matrices of SVM, KNN and RF with the original test results in numbers are shown in Table 7.3, Table 7.4 and Table 7.5 respectively. For further comparison, the precision and the recall are combined into a F_β-measure, as expressed in Eq. (7.6), where the parameter β indicates the relative importance assigned to recall performance over precision.

[12]The training data are in the form of a set of task features computed from the groundtruth semantic components.

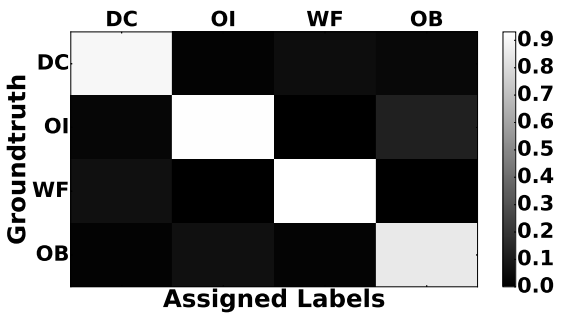


Fig. 7.23 Confusion matrix (normalized) resulting from the SOGP classifier. Precision values appear along the diagonal.

Table 7.3 Confusion matrix resulting from the SVM with the original test results in numbers.

		DC	OI	WF	OB
Groundtruth	DC	1223	520		279
	OI	535	6825	15	233
	WF			4686	457
	OB	247	245	431	2189

Assigned Labels

Since the aim of the proposed approach is to provide motion assistance based on contextual task recognition (*i.e.* the contextual-task aware shared autonomy), a high precision of task classification is more critical for appropriate task motion assistance than a high recall (*i.e.* it is better to provide no assistance than wrong assistance), thus $\beta = 0.5$ is chosen to assign greater importance to precision over recall. Tab.7.6 compares the $F_{0.5}$-measure values of the classifiers after the five-fold evaluation. As can be viewed, the SOGP classifier outperforms the other approaches in classification accuracy even being sparsified from the full GP, meaning that the SOGP classifier is able to scale with large datasets in the long run without compromising classification performance.

Table 7.4 Confusion matrix resulting from the KNN with the original test results in numbers.

		DC	OI	WF	OB
Groundtruth	DC	1280	445	14	283
	OI	478	6900	3	227
	WF		20	4780	343
	OB	347	262	554	1949

Assigned Labels

Table 7.5 Confusion matrix resulting from the RF with the original test results in numbers.

Groundtruth		DC	OI	WF	OB
	DC	1158	474	57	333
	OI	514	6881	12	201
	WF		11	4606	526
	OB	370	174	425	2143

Assigned Labels

Table 7.6 $F_{0.5}$-measure comparison for SOGP, SVM, KNN and RF after 5-fold cross validation.

Classifier	SOGP	SVM	KNN	RF
Doorway Crossing	0.87	0.61	0.61	0.57
Object Inspection	0.93	0.90	0.91	0.91
Wall Following	0.94	0.91	0.90	0.90
Object Bypass	0.84	0.70	0.68	0.67

$$F_\beta = \frac{(1+\beta^2)(\text{precision} \times \text{recall})}{\beta^2 \text{precision} + \text{recall}} \tag{7.6}$$

In addition to the classification accuracy, the speed of the SOGP classifier in data processing is also very important, since it is demanded that the proposed approach possesses real-time performance, in order to employ it in real applications. Towards this aim, 2500 way points for each task type are (randomly) taken from the collected datasets, to incrementally[13] update the SOGP classifier for each task type following the one-vs-all classification formulation (see Chapter 5). The process speed of a SOGP classifier is measured as the average data processed per second, including training and predication of a sample incrementally. The process speed of the SOGP classifier is plotted for each task type in Fig. 7.24. As can be viewed, the process speed of each SOGP classifier keeps above 100Hz, thus the proposed SOGP classifier is demonstrated to possess the real-time performance.

[13] Due to its *Online* property, the SOGP algorithm is famous for the capability of being *incrementally* updated.

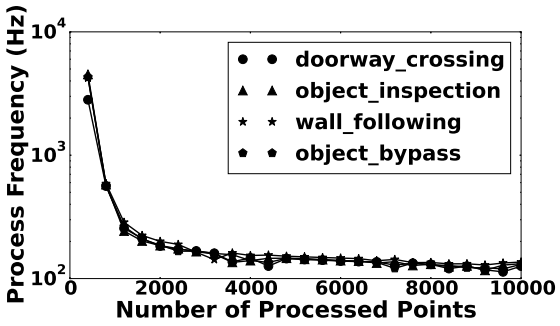


Fig. 7.24 The sequential process speed of the SOGP classifier for each task type. For each task type, there are 2500 positive samples, so there are totally 10000 samples per task type used to update the corresponding SOGP classifier following the one-vs-all classification formulation. The process speed of each SOGP classifier keeps above 100Hz, please note the scale on y-axis.

7.3.2 Evaluation of Classification Uncertainty Estimation

In this subsection, the uncertainty estimates of the probabilistic classification output of the four classifiers to new classes not used in training are compared with each other, since the capability of a classifier to estimate the prediction uncertainty (*i.e.* the introspective capability) is vital for robotic applications involving mission-critical decision making, such as mobile robot teleoperation where this report focuses. It would not be expected that the system is able to provide appropriate motion assistance to a previously unseen motion pattern distinctive from the one used for training.

The classifiers are trained only on datasets from two classes (arbitrarily selected): Doorway Crossing and Object Inspection. Data from Wall Following and Object Bypass are presented for inference, resulting into a classification distribution over binary labels. The normalized entropy value [76] of label distribution over each test point is computed. It measures the uncertainty in the classification decision, hence characterizes the introspective capacity of a classification framework:

$$h_{\mathrm{NE}} = \frac{-\sum_{c \in C} p(\mathbf{q}_* \in c) \log p(\mathbf{q}_* \in c)}{h_{\max}}, \tag{7.7}$$

where $h_{\max} = \log|C|$, denoting the entropy of the $|C|$-dimensional uniform distribution ($|C| = 2$ in this evaluation), and $p(\mathbf{q}_* \in c) = p(t_*^{(c)} \geqslant 0)$. This value ranges between 0 and 1, where a higher value expresses greater uncertainty in the classifier's belief. For each way point along each test trajectory in the following evaluations, the task feature of a way point from the groundtruth semantic target is computed, and the four classifiers with this task

feature are queried to obtain its probability distribution over the binary labels, to facilitate the computation of the normalized entropy value of this point, for ease of the introspection comparison of the four classifiers.

The histogram of the normalized entropy values for each classifier is obtained, as depicted in Fig. 7.25 respectively. As can be noticed, the SOGP classifier assigns clearly much higher normalized entropy for a large majority of the inference points, while the other approaches express comparatively much higher confidence in their classifications, by committing a majority of the *un-modelled* points to one of the known classes with higher certainty (*i.e.* lower normalized entropy). This signifies that, the SOGP classifier exhibits considerably stronger indication of the presence of potentially *un-modeled* classes.

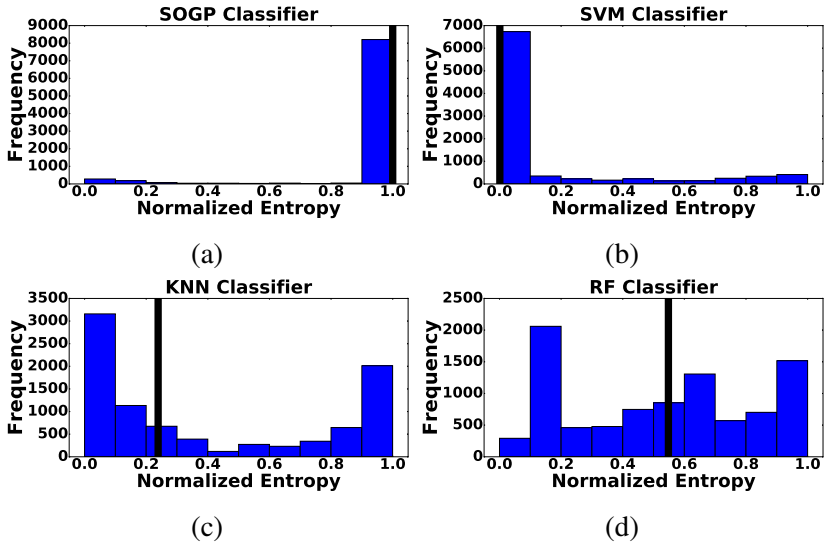


Fig. 7.25 The normalized entropy histograms of the label distribution for SOGP, SVM, KNN and RF respectively, where the black bars indicate the median normalized entropy for the corresponding classifier. Please note the scale on y-axis.

In summary, the experimental results from a set of real tests confirm the statements made in the beginning of this section: first, the SOGP classifier outperforms the baseline classifiers in classification accuracy on the real data, and achieves real-time performance during classification due to the adopted sparsity approximation; second, the SOGP classifier provides considerably better uncertainty estimation over the resulting class labels than the baseline classifiers, when being trained with few classes, affirming that the SOGP classifier is much more useful for detecting potentially unknown classes in the training set, favoring an active life-long learning scenario.

The next section concentrates on investigating the performance of the DPGMM-SOGP approach proposed in Chapter 6 on unsupervised learning and recognizing multiple contextual tasks.

7.4 Evaluation of Unsupervised Learning and Recognition of Multiple Contextual Tasks with DPGMM and SOGP

The ASAP was employed as the robotic platform to evaluate the proposed approach. The average speed of the robot during the evaluations was approximately $0.3m/s$. The involved scenario was mapped and the corresponding semantic components were annotated beforehand.

The following evaluations are made in a post-experimental stage. Since it is concerned about whether the most probable semantic component estimated by the SOGP classifier corresponds to the groundtruth target, the rate of the Correspondence Rate (CR) per test trajectory is computed, and the Average Correspondence Rate (ACR) over all test trajectories is obtained, to characterize the recognition performance of the SOGP classifier in the tests. Additionally, to evaluate whether the most probable motion cluster found by the classifier is able to appropriately interpret the motion patterns of a test trajectory, the average dissimilarity of the most probable task feature to all points in the assigned motion cluster was computed with each way point along each test trajectory, *i.e.* the Intra-cluster Average Dissimilarity (ICAD), which measures the "tightness" of the most probable query task feature to the assigned motion cluster. This metric is meaningful, since the tightness of the classifications measures how well the proposed approach interprets the motion patterns of a test trajectory: a good interpretation of motion patterns requires a low dissimilarity of the most probable query point to the points in the assigned motion clusters. L_2-norm was used as the dissimilarity metric, and the Mean of Intra-cluster Average Dissimilarity (MICAD) over all way points along all test trajectories was obtained, which was utilized together with ACR to evaluate the performance of the proposed approach. Meanwhile, over the following evaluations post clustering, the discriminative SVM classifier with the SE kernel was employed as the baseline approach to compare with the generative SOGP classifier, and the capacity[14] of the SOGP

[14]This choice is to make the SOGP classifier sparser than the SVM whose sparsity is denoted with the number of support vectors after training, which will be shown in the following evaluations.

classifier was set to be 500. The SOGP classifier has already been demonstrated to possess the real-time performance in data processing in the subsection 7.3.1.

In the following evaluations, three statements will be verified to show that the proposed approach serves as a generic framework for representing and exploiting the knowledge of the contextual task executions from unlabelled demonstrations, with the aim of inferring the contextual tasks the human operator performs. First, the proposed approach gives very good recognition results on the test data sampled from the task types used for training in an indoor scenario with multiple candidates, which satisfies the basic requirement for the approach (the subsection 7.4.1). Second, the proposed approach is generalizable to appropriately interpret the motion patterns of new task types not used for training (the subsection 7.4.2). Finally and more importantly, the proposed approach is able to detect unknown motion patterns distinctive from those used in the training set, due to the superior introspective capability of the SOGP classifier (the subsection 7.4.3).

7.4.1 Performance Evaluation with Known Task Types

This subsection aims to evaluate the performance of the proposed approach on recognition of the task types used for training in an indoor scenario with multiple candidates, which is the basic criterion for the proposed approach.

The demonstration data were collected from six volunteers performing each of the four contextual task types: Doorway Crossing (DC), Object Inspection (OI), Wall Following (WF) and Object Bypass (OB) for two times, in an indoor scenario in random order with random starting poses and semantic targets respectively. The map of the scenario is shown together with the annotated candidate semantic components in Fig. 7.26. Totally, 12 trajectories were obtained for each task type, and there were 2254 way points[15] for Doorway Crossing, 6286 points for Object Inspection, 4104 points for Wall Following and 1827 points for Object Bypass, respectively. Firstly, to show the motion clustering result qualitatively, all the collected trajectories were employed as the training data[16] to be clustered by DPGMM, where the inference algorithm was applied iterating the initial number of clusters from 1 to 100, although the clustering results were generally consistent, the one with the highest log likelihood (*i.e.* the evidence) was selected, to ensure good results. Fig. 7.27(a) displays the

[15]The data sampling rate was 5Hz, which was the sampling rate of the joystick.

[16]The trajectories used as the training data were transformed to the sequences of task features computed from the groundtruth semantic components, while possessing no labels for the task types.

discovered motion clusters and the feature data assigned to them in the form of a stacked histogram, colored by the original task types. Moreover, the feature data with their first two components (*i.e.* $\mathbf{s} = (x_\eta, y_\eta)$) are plotted on the joint space and colored according to the original task types and the discovered clusters in Fig. 7.27(b) and Fig. 7.27(c) respectively. As can be viewed, a majority part of Wall Following and Object Bypass feature data are grouped into two sides, representing the motion patterns which are demonstrated in either left or right side of the robot regarding the semantic targets for the two task types respectively. Likewise, a majority part of Object Inspection feature data are assigned to two separate clusters, although not evidently illustrated in in Fig. 7.27(b) and Fig. 7.27(c), corresponding physically to the situations when the robot is demonstrated to inspect the target objects in either clockwise or counter-clockwise direction. Upon consideration, they are reasonable distinctions, and initially not thought of by the demonstrator. This property is critical, since it allows the DPGMM to determine action primitives unknown even to the demonstrator. Meanwhile, most motion clusters consist of a blend of feature data from multiple task types, which represents the overlaps of the motion patterns of them, potentially resulting from that the robot was always operated to firstly align with the target, then approach it during demonstration. On the other hand, the split of the overlap feature data into a series of clusters suggests that the DPGMM is finding too many distinctions, rather than not learning to distinguish.

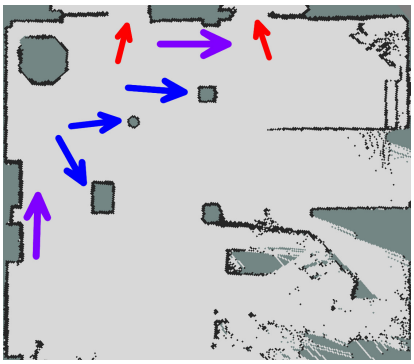


Fig. 7.26 The map of the scenario used for evaluations with different test fractions, and the extracted semantic components, being denoted by arrows with different colors: doorways (red), objects (blue), and wall segments (violet).

Then, to quantitatively evaluate the performance of the proposed approach, the collected dataset was randomly split into test and training with test fractions varying as 0.25, 0.5 and 0.75 based on the trajectory number. In each test phase, the training data were firstly clustered by DPGMM in the same way as above. Post clustering, the SOGP classifier and the SVM classified each way point along each test trajectory with the learned motion clusters.

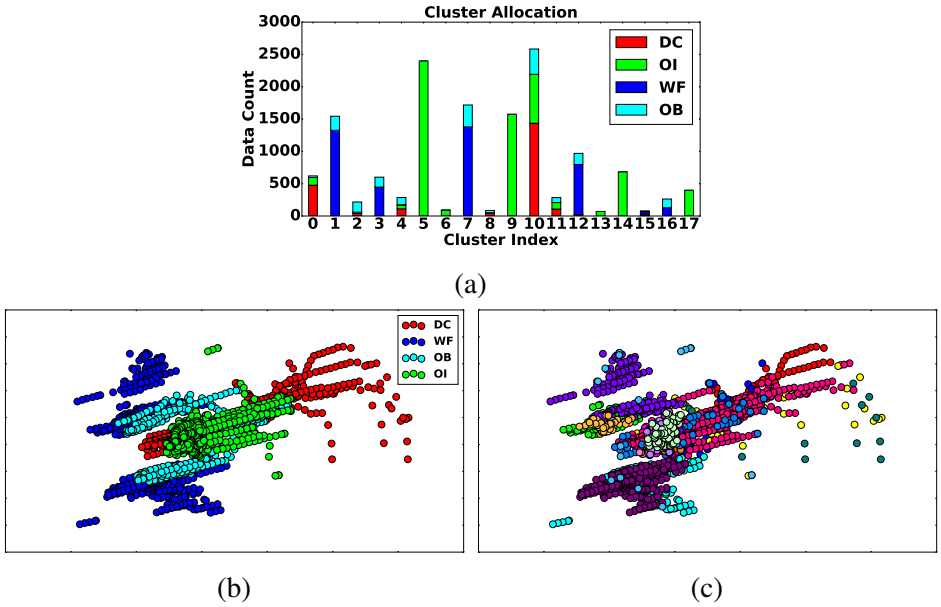


Fig. 7.27 The qualitative result of motion clustering: (a) The stacked histogram shows the discovered motion clusters and the feature data assigned to them, colored by the original task types, please note the scale on y-axis; (b) The feature data, mapped into 2D with their first two components and colored by the original task types; (c) The feature data, mapped into 2D with their first two components and colored by the discovered motion clusters.

To determine the model parameters, for SVM, a grid-search was done on its parameters over reasonable sets, while for SOGP classifier, the (locally) optimized hyper-parameters were obtained by maximizing the evidence of full GP via gradient-descent. The test results within different fractions are displayed in Table 7.7, including the numbers of the training samples, the learned clusters and the support vectors used by SVM, the ACR and the MICAD of the two classifiers, respectively. As can be viewed, the ACR of the SOGP classifier decreases obviously compared between the test fraction 0.25 and 0.5, but maintains stable between the fraction 0.5 and 0.75. The tightness measurements of the SOGP classifier keep stable across the test fractions. In general, the SOGP classifier yielded very good recognition results even being trained with much more clusters than the number of task types used for demonstration. In comparison with SVM, being confirmed by the paired T test for the measurements of CR and ICAD, the SOGP classifier performs considerably better than SVM in all test fractions, even with a sparser representation denoted by the capacity size of the SOGP classifier and the number of the support vectors used by SVM. This verified that the proposed approach is able to correctly recognize the motion patterns it learns from the demonstrations.

Table 7.7 The ACR and the MICAD comparison for the SOGP classifier and the SE SVM post clustering with varying test data fractions, along with the numbers of the training samples and the discovered clusters. The capacity size of the SOGP classifier and the number of support vectors used by SVM are also listed in each test fraction for comparison of the sparsity of the two classifiers.

Experiment	Test Fraction 0.25		Test Fraction 0.5		Test Fraction 0.75	
No. of Training Samples	10284		7718		3540	
No. of Motion Clusters	14		16		12	
Classifier	**SOGP**	**SVM**	**SOGP**	**SVM**	**SOGP**	**SVM**
Sparsity	c: 500	sv: 1894	c: 500	sv: 896	c: 500	sv: 513
ACR	0.93	0.80	0.88	0.74	0.88	0.79
MICAD	0.88	1.11	0.87	1.16	0.87	1.32

7.4.2 Performance Evaluation with New Task Types

Apart from recognizing the learned motion patterns, it is also valuable to investigate whether the proposed approach is generalizable to correctly capture new task types not used for training. Therefore the test data from performing three new contextual task types were collected: Wall Inspection, Robot Docking[17] and Gap Crossing for two times each with random initial poses respectively, in another cluttered indoor scenario whose map and the corresponding semantic components are illustrated in Fig. 7.29. The whole dataset sampled in the subsection 7.4.1, were provided for clustering (see Fig. 7.27) then training the SOGP classifier and the SE SVM in the same manner, and the datasets collected in this subsection were presented for inference. The ACR and the MICAD values of the two classifiers over all test trajectories were computed to compare their recognition performance, which are listed in Table 7.8, together with the numbers of the training samples, the discovered motion clusters and the support vectors used by SVM.

It might be noticed that the ACR of the SOGP classifier achieved in this subsection is obviously lower than that in the subsection 7.4.1. After analysis, it is found that it results from the test trajectories for the task type Gap Crossing, where the groundtruth motion pattern was to operate the robot to cross a small gap between two objects, and the SOGP classifier was interfered by the two nearby objects when estimating the semantic targets along the trajectory. Fig. 7.28 displays the most probable semantic targets estimated by the SOGP classifier along one test trajectory (shown in Fig. 7.35(a)) from performing Gap Crossing, which are denoted with star markers. The groundtruth target was “gap1”, which is indicated

[17]The robot was to be docked into a table.

by a dashed line. As can be viewed, most incorrect estimations are caused by the two objects forming the gap.

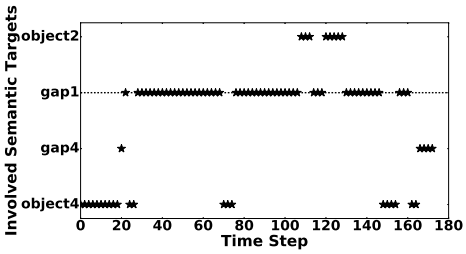


Fig. 7.28 The most probable semantic targets estimated by the classifier along one test trajectory (please refer to Fig. 7.35(a)), which are denoted with the star markers, and the dashed line indicates the groundtruth target.

Overall, as confirmed by the paired T test for the measurements of CR and ICAD, the proposed approach is able to recognize the new task types with considerably better performance on the evaluation metrics using the sparser SOGP classifier than using the SVM.

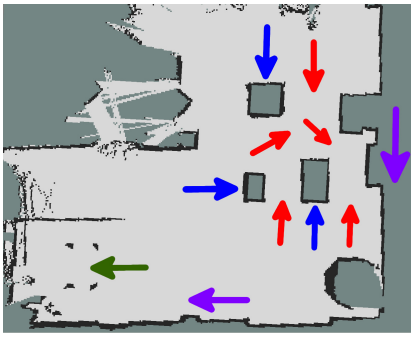


Fig. 7.29 The map of the scenario used for evaluations with three new task types, and the extracted semantic components, being denoted by arrows with different colors: gaps (red), the docking target under the table (green), objects (blue), and wall segments (violet).

7.4.3 Introspection Evaluation with Distinctively Unknown Task Types

In real and long term applications, it is hardly possible to train the robot with all the needed task types before deployment due to time and economic considerations on one hand. On the other hand, as shown in the previous subsection, the robot is supposed to utilize the motion clusters to generate autonomous motion commands, which assists the teleoperation by fusing them with the user inputs based on the probability/confidence of the task recognition, thus it would not be expected that the system can provide appropriate motion assistance to a

Table 7.8 The ACR and the MICAD comparison for the SOGP and the SVM classifiers post clustering on test data collected from performing three new task types, along with the numbers of the training samples and the discovered clusters. The capacity size of the SOGP classifier and the number of support vectors used by SVM are also listed for comparison of the sparsity of the two classifiers.

No. of Training Samples	14471	
No. of Motion Clusters	18	
Classifier	SOGP	SVM
Sparsity	c: 500	sv: 2723
ACR	0.82	0.74
MICAD	0.90	1.11

previously unseen motion pattern distinctive from the one used for training, for example when it is initially trained with Doorway Crossing and later applied to assist Object Inspection[18], even if the system correctly recognizes the corresponding semantic targets. Therefore, for robotic applications involving mission-critical decision making, such as mobile robot teleoperation where this report focuses, it is imperative to investigate a classifier's capability of uncertainty estimation when classifying the motion clusters for a query task feature, *i.e.* the *introspective* capability [76] of the classifier. To characterize the introspective capability of a classifier, the normalized entropy value [76] for each query point was computed based on its discrete probability distribution over the discovered motion clusters. For each way point along each test trajectory used in the following evaluations of this subsection, the task feature of a way point from the groundtruth semantic target was computed, and the SOGP classifier and SVM were queried with this task feature to obtain its discrete probability distribution over the learned motion clusters, to facilitate the computation of the normalized entropy value of this point, for ease of the introspection comparison of the two classifiers.

In this subsection, the dataset collected in the subsection 7.4.1 was used, where two task types were arbitrarily selected for clustering then training the two classifiers: the SOGP classifier and the SE SVM in the same way as the subsection 7.4.1, and the datasets from the other two task types were used for inference, attempting to do the introspection evaluation with distinctive motion patterns. In order to mitigate any influences of the specific training and test data selected, such evaluation procedure was repeated across all possible task type combinations for training, resulting into six groups of the normalized entropy values. The mean and standard deviation normalized entropies of each of the six test groups are listed in

[18]Performing Doorway Crossing means to drive the robot to simply approach the target doorway, while Object Inspection aims to not only approach the target object, but also move around it within certain distance while facing it.

Table 7.9 Mean and standard deviation normalized entropies from six iterations of training and testing, where the datasets from two task types were used for training, and the rest data were presented for inference. The total datasets are collected from performing four task types. The MICAD measurements of the two classifiers in each test iteration are also listed.

Test Task Types	Classifier	Normalized Entropy $\mu \pm$ std.err.	MICAD
OI and WF	SOGP	$\mathbf{0.703 \pm 0.402}$	1.30
	SVM	0.498 ± 0.351	1.52
OB and WF	SOGP	$\mathbf{0.900 \pm 0.232}$	1.44
	SVM	0.151 ± 0.290	2.08
OB and OI	SOGP	$\mathbf{0.796 \pm 0.369}$	1.52
	SVM	0.354 ± 0.215	1.56
DC and WF	SOGP	$\mathbf{0.550 \pm 0.331}$	1.00
	SVM	0.218 ± 0.286	1.05
DC and OB	SOGP	$\mathbf{0.535 \pm 0.367}$	1.01
	SVM	0.304 ± 0.332	1.12
DC and OI	SOGP	$\mathbf{0.857 \pm 0.271}$	1.41
	SVM	0.641 ± 0.263	1.56

Table 7.9 respectively, together with the MICAD measurements of the two classifiers. As confirmed with the paired T test, the mean normalized entropies for the SOGP classifier are considerably higher than those of the SVM classifier, signifying that the former exhibited greater uncertainty in the judgement, indicating strongly the presence of potentially *unmodelled* motion patterns, which is also suggested by the high MICAD values (compared with those in the subsection 7.4.1) across all test iterations, while the latter was extremely confident in its classifications with lower values of the normalized entropy, even though the high MICAD values imply a potential inappropriate interpretation of the motion patterns. In practice, the robot can utilize this outstanding introspective capability of the SOGP classifier to *actively* query for an update of the demonstration data without manual labels to increase its knowledge regarding the uncertain motion patterns, which are potentially distinctive to those already absorbed in its knowledge base. This property is vital to fulfil the vision of a life-long adaptive assistive robot adopted by this thesis.

7.4.4 Evaluation of Local Intentional Trajectory Prediction with Classified Motion Clusters and Recognized Contextual Information

This subsection focuses on evaluating the performance of the local intentional trajectory prediction algorithm proposed in Chapter 6 with the classified motion clusters and recognized contextual information. Regarding the parameters of the proposed algorithm, throughout the following evaluations, the simulation time for extrapolation was set $\Delta_t = 0.05s$ (*i.e.* 20Hz) to be the same as the sampling period of the dataset, and the threshold of the Mahalanobis distance of each predicted way point was set $d_{\text{mh\_thres}} = 3.0$ empirically. The fast online GMR approach used 2 Gaussian components for regression, and employed diagonal covariance matrix for each Gaussian component to simplify computation. Its training data size was 10.

The motion clusters obtained in the subsection 7.4.1 (please refer to Fig. 7.27) were employed to train the SOGP classifier. The trained classifier was used to classify the motion clusters and recognize the associated semantic components along the test trajectories. There were *three* sources regarding the test trajectories, and all the test trajectories were recorded from test participants manually driving the robot for certain predefined (yet unknown to the robot) contextual tasks with random initial poses.

The first source was the *nine* trajectories collected from the human operator sequentially performing three contextual tasks for nine times: firstly following the wall segment, then inspecting an object, finally crossing a doorway. Such source contained the four contextual task types used for clustering, *i.e.* Doorway Crossing, Object Inspection, Wall Following and Object Bypass, and it was sampled from performing the fourth task assignment introduced in the subsection 7.2.3. The second source was the *six* trajectories collected from the human operator performing each of the three contextual tasks: Wall Inspection, Robot Docking and Gap Crossing, for two times, and it was sampled from the evaluations made in the subsection 7.4.2. These three task types were not demonstrated for clustering, but they share motion similarity with the learned ones. Hence the aim of this source was to evaluate the generalizability of the proposed approach. The last source was the *five* trajectories collected in the same scenario as that used in the second source. They were recorded from two volunteers manually driving the robot[19] to execute a set of contextual tasks along the way in the cluttered scenario, *e.g.* crossing a narrow gap, inspecting an object or a wall segment, bypassing an object and docking into a table, with the aim of comprehensively evaluating the proposed algorithm.

[19]Two test trajectories were obtained from one volunteer, while the rest three were collected from the other one.

Totally, there were *twenty* test trajectories, consisting of 15575 way points obtained from different human operators controlling the robot to perform seven contextual task types[20] in different scenarios. Especially, they contained the samples collected from executing certain task types in alternative ways, *e.g.* object inspection (in either clockwise or counter-clockwise direction regarding the target object during inspection), wall following (to follow the wall segment on either side of the robot), and object bypass (to pass the target object by either side of the robot). Thus these twenty trajectories are the appropriate test datasets to evaluate the *scalability* and *effectiveness* of the proposed approach (and the proposed DPGMM-SOGP framework for unsupervised contextual task learning and recognition indirectly).

To comprehensively evaluate the proposed intentional trajectory prediction algorithm (and also the proposed DPGMM-SOGP framework), the following tests are grouped into two parts: the qualitative evaluations and the quantitative ones.

7.4.4.1 Qualitative Evaluations

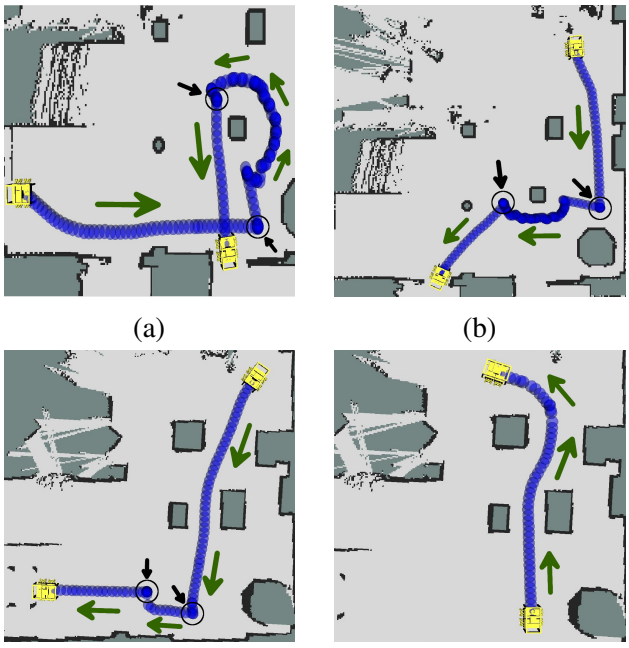
In the qualitative evaluations, the proposed approach was applied to certain way points of several test trajectories, to estimate local trajectories originating from them. The estimated and the groundtruth trajectory segments starting from the same poses and possessing approximately the same length[21] are drawn together, aiming to provide the visual comparisons between the estimations and the groundtruth results. In the following comparison figures, apart from the two trajectory segments with different colors, a bar is put on the most probable semantic target recognized by the classifier at the applied way point, with its height indicating the classification confidence. The footprint of the robot is also denoted with a pink polygon in these figures.

Firstly, the performance of the proposed approach on estimating the four motion patterns (including their possible variations) used for training are investigated. Fig. 7.30 illustrates the four groundtruth trajectories adopted for these qualitative evaluations.

For *Doorway Crossing*, Fig. 7.31 depicts the comparison done at one way point of the trajectory in Fig. 7.30(a). As can be noticed, the estimated trajectory matches the groundtruth one quite well, and the correct semantic target is also recognized with high confidence,

[20] Doorway Crossing, Object Inspection, Wall Following, Object Bypass, Robot Docking, Wall Inspection and Gap Crossing

[21] The length of a trajectory is obtained by accumulating the Euclidean distances (computed with x and y coordinates of the recorded poses of way points) between its ordered pairs of way points.



(a) (b) (c) (d)

Fig. 7.30 The groundtruth trajectories for qualitatively evaluating the proposed approach on estimating the four motion patterns used for training, where the robot was manually operated to execute various contextual tasks in cluttered scenarios. The black arrows denote the manually labeled split points of the sequentially performed tasks (when applicable), while the green ones indicate the movement directions of the robot. (a) A sequence of tasks were executed: the robot followed wall segment on its right side at first, then inspected target object in counter-clockwise direction, finally crossed target doorway. (b) Similarly, the robot followed wall segment on its right side at first, then inspected target object in clockwise direction, finally crossed target doorway. (c) The robot moved among obstacles at first, then inspected wall segment in one direction, finally docked into a table. (d) The robot moved among obstacles to reach the other side.

demonstrating qualitatively the performance of the proposed approach in estimating this motion pattern.

For *Object Inspection*, after approaching the target object, there are two operational directions to inspect it, *i.e.* in either clockwise or counter-clockwise direction regarding the target object. Fig. 7.32(a) compares the groundtruth trajectory segment (from the trajectory in Fig. 7.30(a)) with the one estimated at the way point where the human operator was driving the robot to approach the target object. Since the motion pattern of *approaching object* has been learned from demonstrations, the execution of it can be captured with high

[22]The transparency of the groundtruth segment (in red) is increased to highlight the predicted one (in blue). This practice is also applied to Fig. 7.36 and Fig. 7.37 for reader's better understanding.

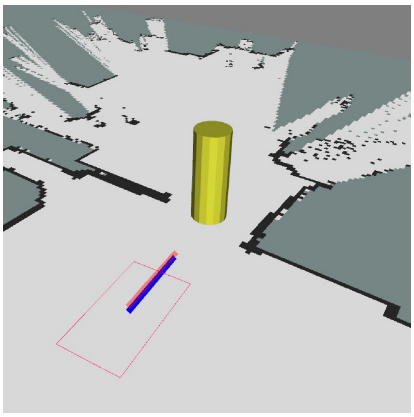
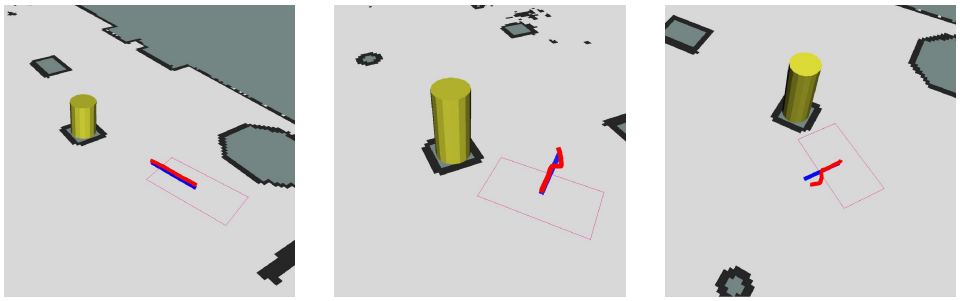


Fig. 7.31 The estimated trajectory (blue) is compared with the groundtruth segment[22] (red), where the robot was being operated to cross doorway. The pink polygon represents the footprint of the robot. A bar is put on the most probable semantic target estimated by the classifier at this way point, with its height indicating the estimation confidence.

confidence and accuracy. With respect to the inspection phase, Fig. 7.32(b) and Fig. 7.32(c) draw the comparisons done at the two way points (from the trajectories in Fig. 7.30(a) and Fig. 7.30(b)) where the robot was moving along the target object while facing it in counter-clockwise and clockwise directions respectively. In both situations, the human motion patterns are correctly interpreted and estimated with high confidence, despite the jitters of the recorded groundtruth way points resulting from the strong motion drifts of the robot when moving sideways on the uneven floors of the scenarios.

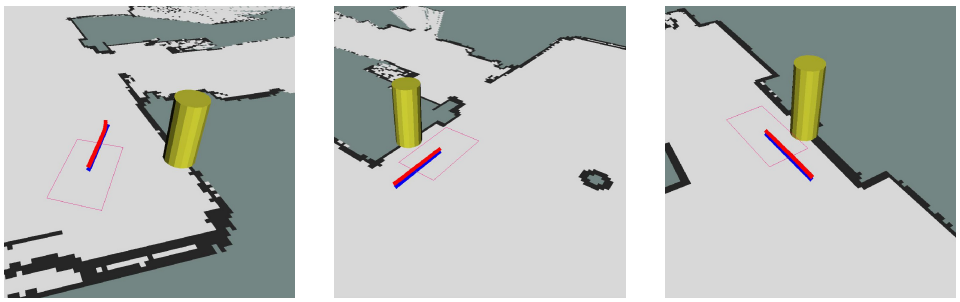


(a) (b) (c)

Fig. 7.32 The estimated trajectories are compared with the groundtruth segments, where the human operator was driving the robot to execute object inspection task. The color coding is the same as Fig. 7.31. (a) The robot was approaching the target object. (b) It was inspecting the object in counter-clockwise direction. (c) It was inspecting the object in clockwise direction.

Regarding *Wall Following*, the robot can follow the target wall segment on either right or left side of it. Fig. 7.33(b) and Fig. 7.33(c) depict the comparisons done at the two way

points (from the trajectories in Fig. 7.30(a) and Fig. 7.30(b)) where the robot was following the target wall segment on its right and left sides respectively. As displayed in both figures, the results of the semantic target recognition and the local trajectory estimation qualitatively demonstrate that, the human motion intentions are correctly interpreted and predicted with high confidence in both situations. When executing Wall Following, the human operator usually controls the robot to approach the target wall segment while aligning the robot with its surface, especially when the robot is noticed to be not close enough to the wall. Such motion pattern is included in the demonstration datasets from performing Wall Following, and is learned by the proposed DPGMM-SOGP framework. Hence its execution can be correctly recognized during operation, and the corresponding local trajectory is estimated with high accuracy, as displayed in Fig. 7.33(a).

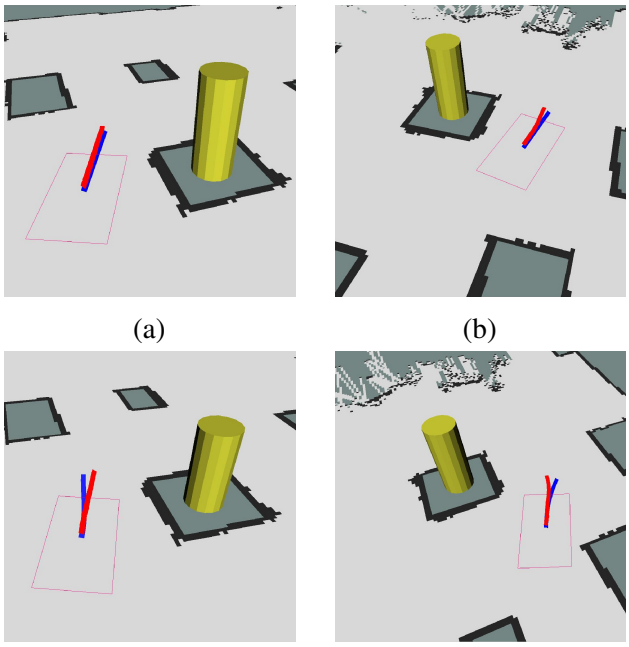


(a) (b) (c)

Fig. 7.33 The estimated trajectories are compared with the groundtruth segments, where the human operator was driving the robot to perform wall following task. The color coding is the same as Fig. 7.31. (a) The robot was aligning with the wall segment. (b) It was following the wall segment on its right side. (c) It was following the wall segment on its left side.

Regarding *Object Bypass*, the human operator can execute it in alternative directions, *i.e.* to pass the target object by either left or right side of the robot. Fig. 7.34(a) illustrates the comparison (applied to the trajectory in Fig. 7.30(c)) where the robot was avoiding the object on its right side, while Fig. 7.34(b) depicts the comparison (applied to the trajectory in Fig. 7.30(d)) where the robot was bypassing the object on its left side. As can be viewed, the task motions of the human operator in both situations are correctly interpreted and predicted with high accuracy. Interestingly, at some way points when performing Object Bypass, the estimated trajectories are noticed to aim to guide the robot further away from the nearby obstacle then the groundtruth trajectory segments, as shown in Fig. 7.34(c) and Fig. 7.34(d), respectively. It is deduced that, by learning from demonstrations, the proposed approach interprets such situations more conservatively than the human operator actually executing this movement from the perspective of robot safety. How such action *deviation* between human

and robot will influence the human-robot interaction within a shared autonomy system in reality remains an attractive point for further investigation.



(a) (b) (c) (d)

Fig. 7.34 The estimated trajectories are compared with the groundtruth segments, where the human operator was controlling the robot to perform object bypass task. The color coding is the same as Fig. 7.31.

The above qualitative evaluation results comprehensively confirm the performance of the proposed approach on estimating the motion patterns (including their possible variations) used for training. The following qualitative evaluations are focused on investigating the performance of the proposed approach on estimating the motion patterns (including their possible variations) not employed for training, yet sharing motion similarity with the learned ones. Fig. 7.35 depicts the four groundtruth trajectories used for the following qualitative evaluations.

For *Gap Crossing*, Fig. 7.36(a) draws the comparison (applied to the trajectory in Fig. 7.35(a)) where the robot was being controlled to cross a gap between two obstacles. As can be seen, at this way point, the classifier correctly recognizes the semantic target with high confidence, and the estimated local trajectory matches the groundtruth one quite well. Fig. 7.36(b) illustrates another comparison applied to the same trajectory. Although the recognized semantic target is incorrect (the object nearby instead of the gap), the estimated trajectory still matches the groundtruth one quite well. The motion cluster assigned to the

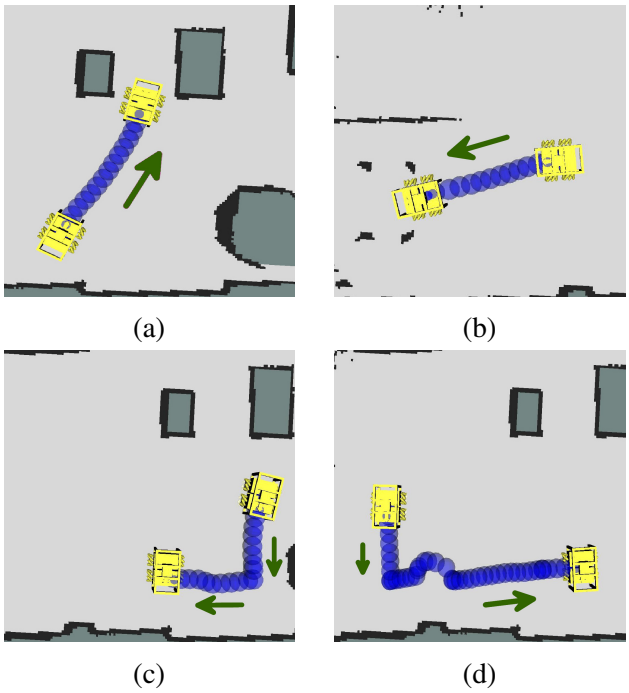
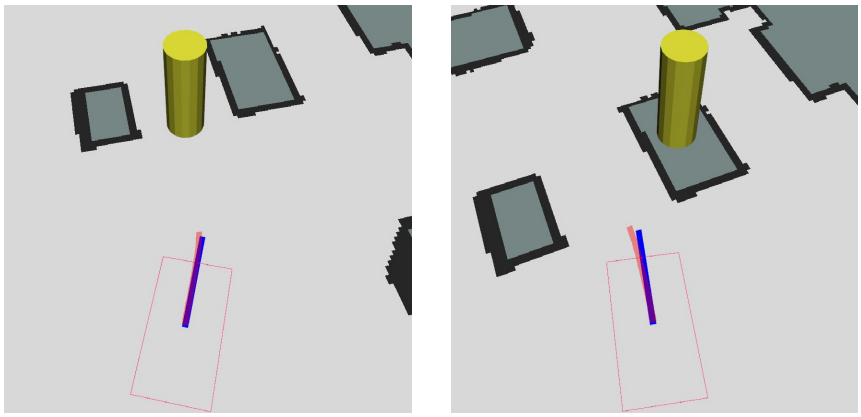


Fig. 7.35 The groundtruth trajectories for qualitatively evaluating the proposed approach on estimating the three motion patterns not demonstrated during training. The color coding is the same as Fig. 7.30. (a) The robot was crossing a gap between objects. (b) The robot was docking into a table. (c) The robot approached wall segment at first, then inspected it in one direction. (d) The robot inspected wall segment in another direction after approaching it.

starting way point is further examined (the 3rd cluster, please refer to Fig. 7.27(a)), and it is noticed that it consists of the demonstrations from Wall Following and Object Bypass, hence it is deduced that the proposed DPGMM-SOGP framework interprets the motion pattern at this way point as to bypass the target object, which is reasonable for this situation.

For *Robot Docking*, Fig. 7.37 depicts the comparison (applied to the trajectory in Fig. 7.35(b)) where the robot was docking into a table in front. The semantic target is correctly recognized with high confidence, and the local trajectory is estimated with high accuracy. Moreover, after checking the motion cluster assigned to the starting way point (the 10th cluster, please refer to Fig. 7.27(a)), it is noticed that it mainly consists of the demonstrations from Doorway Crossing. Therefore, it is posited that the proposed DPGMM-SOGP framework interprets the motion pattern at this way point by generalizing from the learned motion pattern of Doorway Crossing.

Regarding *Wall Inspection*, it can be executed in alternative directions after the robot approaching the target wall segment. The comparisons depicted in Fig. 7.38(b) (applied to



(a) (b)

Fig. 7.36 The trajectories estimated at two way points are compared with the groundtruth segments, where the robot was being operated to cross a gap between obstacles. The color coding is the same as Fig. 7.31.

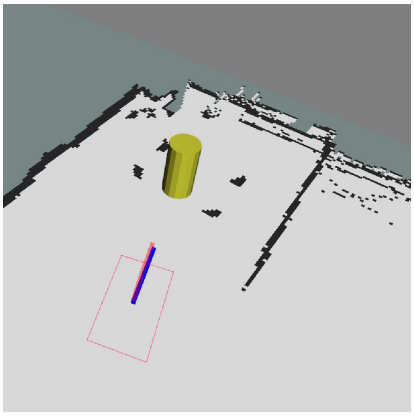
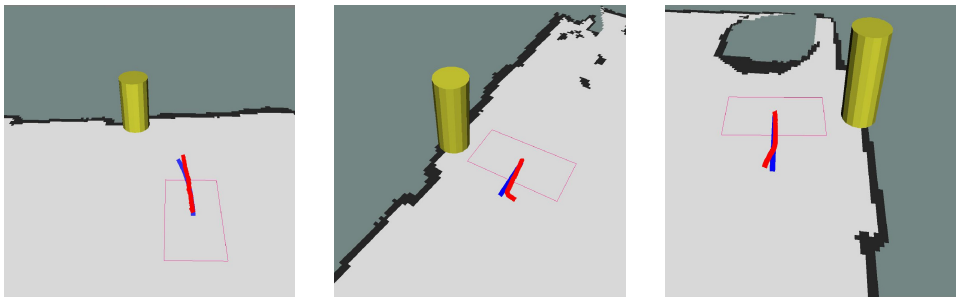


Fig. 7.37 The estimated trajectory is compared with the groundtruth segment, where the robot was being controlled to dock into a table. The color coding is the same as Fig. 7.31.

the trajectory in Fig. 7.35(d)) and Fig. 7.38(c) (applied to the trajectory[23] in Fig. 7.35(c)) verify the very accurate estimation results of the proposed approach in both situations. This means that, the proposed DPGMM-SOGP framework is able to distinguish the two situations, and correctly interprets the human motion patterns and recognizes the associated semantic target. With the information, the proposed intentional trajectory algorithm predicts the local trajectory the human operator intends to follow in future steps with high accuracy. Meanwhile, the human motion pattern of driving the robot to approach the target wall segment is also estimated with high confidence and accuracy, as displayed in Fig. 7.38(a).

[23]Please notice the jitters of the recorded way points in this trajectory due to the strong motion drifts of the robot when moving sideways on the uneven floor of the scenario.



(a) (b) (c)

Fig. 7.38 The estimated trajectories are compared with the groundtruth segments, where the human operator was driving the robot to execute wall inspection task. The color coding is the same as Fig. 7.31. (a) The robot was approaching the target wall segment. (b) It was inspecting the wall segment in one direction. (c) It was inspecting the wall segment in another direction.

The above qualitative evaluation results prove the performance of the proposed approach in generalizing to the task types (including their possible variations) not used for training. In summary, the overall qualitative results in this part verify the scalability and effectiveness of the proposed DPGMM-SOGP framework and the proposed intentional trajectory prediction algorithm. The next part of this subsection will present the quantitative evaluation results with the baseline approaches.

7.4.4.2 Quantitative Evaluations

To further quantitatively evaluate the performance of the proposed algorithm, the batch GMR and Locally Weighted Projection Regression (LWPR) [195] are employed as the baseline regression approaches. LWPR is a popular machine learning algorithm seeking to provide incremental, realtime inference and prediction for high-dimensional input-output function approximation. Sharing the similar aim, it is poised to compete with the fast online GMR approach. The library proposed in [103] is used for its implementation, and its (hyper-)parameters are tuned according to the suggestions introduced in [102].

The batch GMR model uses full covariance matrix, and it is trained by the EM method with up to ten components on each learned motion cluster beforehand. For each motion cluster, the optimal model is selected among the candidate ones based on the BIC. The LWPR model is also trained for each motion cluster in advance, and during training, $\omega_{\text{gen}} = 0.2$, an

initial setting of $d_* = 2.0$ and blending is enabled. The predictions made by both approaches were also thresholded with $d_{\text{mh\_thres}} = 3.0$.

The three approaches are applied to estimate the local trajectory at each way point of all test trajectories respectively. To characterize the estimation error in a straightforward manner[24], at each way point for examination, the *pairwise Euclidean distance* between the estimated trajectory and its corresponding groundtruth one having the same number of way points[25] is computed. Finally, the means and the standard deviations of such error measurement are obtained for the three approaches over all test trajectories respectively. They are shown in Fig. 7.39. The results indicates that the fast online GMR approach outperforms the baseline approaches in this test.

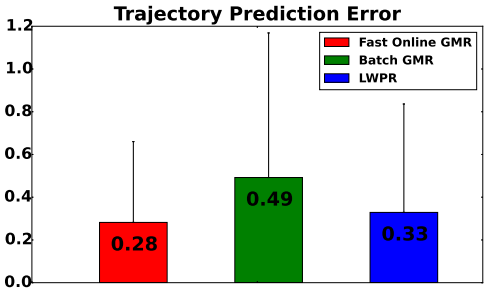


Fig. 7.39 The means and the standard deviations of the trajectory prediction errors for the three regression approaches.

In addition to the prediction accuracy, the prediction speed of the proposed approach is also the concern of this thesis, since it performs the learning and prediction online during operation. In the application of this thesis, the required process speed[26] corresponds to $\sim$ 30Hz. Because in real applications, the estimation is made after the motion cluster classification and semantic target recognition, it is meaningful to evaluate the process speed of the whole pipeline (*i.e.* motion cluster classification, semantic target recognition and local trajectory estimation). To test this, the three regression approaches combined with the SOGP classifier are applied to each way point of all test trajectories respectively.

At each test way point, the process time of each combination consists of the time for motion cluster classification and contextual information recognition by SOGP and the time

[24]For more information regarding trajectory similarity comparison, please refer to [186].

[25]The distance between two way points is computed with their recorded localized poses in the involved scenario, *i.e.* $\mathbf{p}_{\text{wp}} = (x_{\text{wp}}, y_{\text{wp}}, \theta_{\text{wp}})$

[26]At least, it should be higher than the data sampling period which is 20Hz

for trajectory prediction by the corresponding regression approach[27]. Such process time of each combination is then recorded at each test way point respectively. In the end, the means and the standard deviations of the process time of the three combinations per way point over all test trajectories are obtained respectively, as shown in Fig. 7.40. The results show that the combination using the proposed online GMR approach achieves the fastest process speed (approximately 55Hz). Though it is not considerably faster than the baseline approaches, no pre-training is needed to employ the proposed online GMR approach. Such results further verify that the winning combination satisfies the realtime requirement of the proposed application.

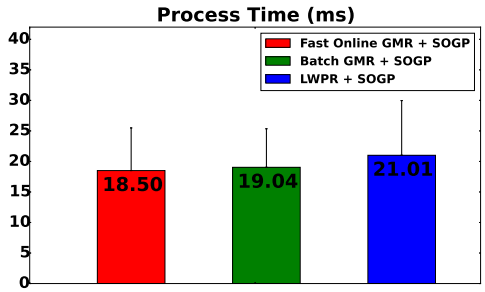


Fig. 7.40 The means and the standard deviations of the process time of the three prediction combinations at each test way point.

To summarize the evaluations made in this section, the experimental results acknowledged that, the proposed DPGMM-SOGP approach serves as a generic framework for representing and exploiting the knowledge of the contextual task executions from unlabelled demonstrations in mobile robot teleoperation, with the aim of recognizing the motion patterns the human operator performs based on the contextual information. Meanwhile, the results from the qualitative and the quantitative evaluations confirmed the effectiveness of the proposed intentional trajectory prediction algorithm, especially its realtime property in data processing to efficiently learn and make predictions with high accuracy. This excellence renders the proposed algorithm feasible for the online use, bridging the task recognition part and the task motion assistance part of the proposed shared autonomy system. The overall evaluation results in this section verified the scalability and effectiveness of the proposed approaches in learning, recognizing and utilizing the motion pattern knowledge of the human operator in an unsupervised data-driven manner from human demonstrations, and the proposed contextual-task aware adaptive shared autonomy system is realized with them.

[27]The SOGP classifier, the batch GMR model and the LWPR model are trained *off-line* on each motion cluster, thus the training time of them is not considered in the process time.

The next section reports the evaluations of the reactive shared autonomy presented in Chapter 3 in simulation, to verify its effectiveness in safely assisting the human operator with the teleoperation task.

7.5 Evaluation of Reactive Shared Autonomy System to Safely Assist Teleoperation in Simulation

To evaluate the proposed approach, a teleoperation task was conducted in a simulated office environment. The same simulated quadrotor as the subsection 7.2.1 was employed for evaluations, and the task was to teleoperate the robot to complete a predefined path in an indoor environment, as shown in Fig. 7.41. During the task, the operator was required to control the robot to cross several narrow doorways and one narrow corridor using a Logitech F710 wireless gamepad. The radius of the robot is $0.39m$, and the controller frequency is 20Hz.

Five subjects (three males and two females, aged 22 to 28) were asked to finish this teleoperation task under two operating modes: Manual mode and Adaptive Shared Autonomy (ASA) mode. Under the Manual mode, there was no support from the robot, while the proposed approach was activated under the ASA mode. To eliminate the biases, the order of the two modes was unknown to the participants. A variety of important data were time-logged during the tests, *e.g.* robot poses provided by the simulator, laser readings, joystick commands *etc.*. They were used for the offline evaluation of the proposed method after the tests.

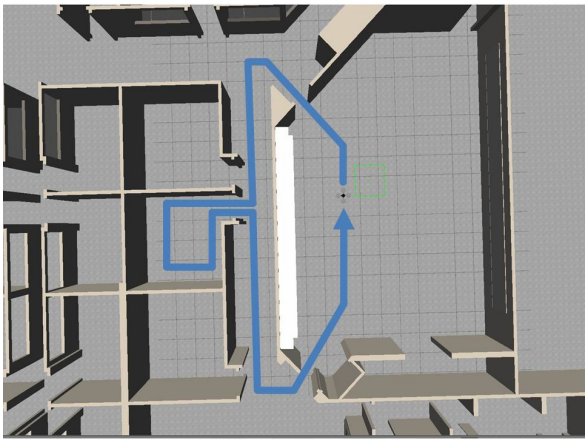


Fig. 7.41 The navigation task in the simulated office environment

The user interface of the simulation include the following components: the laser scan reflections from the LRF with the robot model, as shown in the left part of Fig. 7.42; the front camera image, as shown in the top right part of the same figure; and the tail camera image displaying the back view of the robot, as shown in the bottom right part of Fig. 7.42. The LRF has a 360° field of view. The camera image was not processed. During the task execution, the operator had no access to the global view of the simulated environment, but observed the environment only through the user interface.

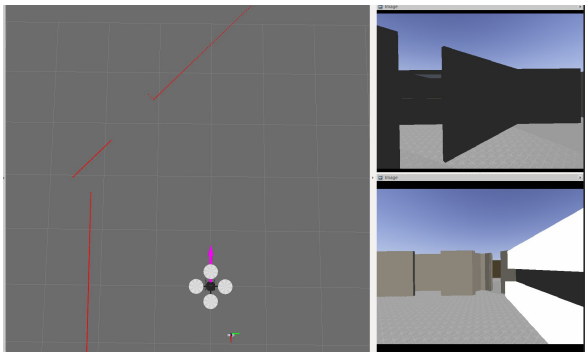


Fig. 7.42 User interface of the simulation in ROS rviz: left: laser scan image, the arrow starting from the robot represents the user input via joystick; top right: front camera image; bottom right: tail camera image.

Fig. 7.43 shows the snapshots of the instant recognition probabilities of the three situations obtained from the logged data of one test under the ASA mode. To provide a simple view of the occupied situation of the environment around the robot, the local occupancy grid map S is thresholded with the iterative dynamic threshold method, and grouped into occupied and free sectors respectively. The thresholded occupancy grid map is used only for the display. The occupied sectors of the map are illustrated with the green blocks in the laser scan image, as shown in Fig. 7.43.

As can be viewed in Fig. 7.43(b), the user input points at an occupied sector of the map, thus the situation Object Approach (OA) has the highest a-posteriori probability according to the definition of the situation OA (Eq. (3.20) in Chapter 3). In Fig. 7.43(c), the user input points at a free sector, therefore the situation General Obstacle Avoidance (GOA) has the highest a-posteriori probability.

The two female subjects did not succeed to finish the task under the Manual mode because of the collision leading to the crash of the quadrotor. All the male subjects finished the task under the Manual mode successfully. This is partly because that the two female subjects have

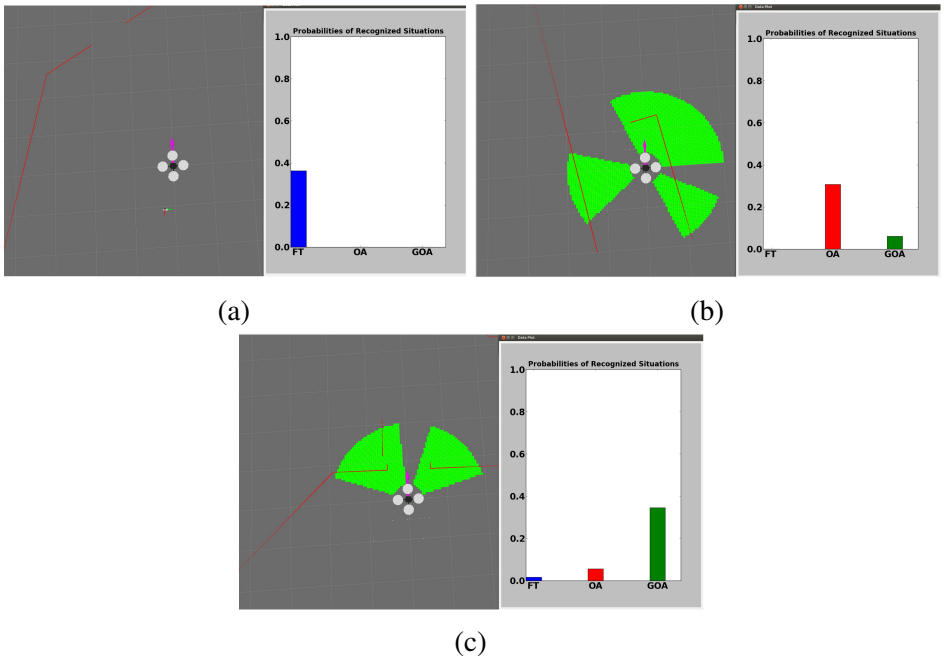
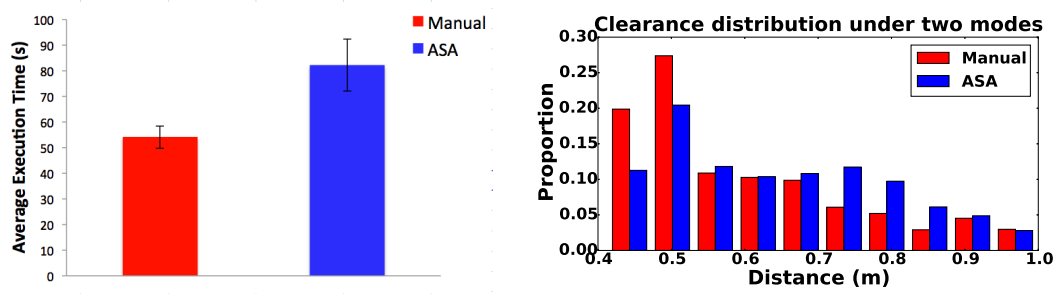


Fig. 7.43 Snapshots of the instant recognition probabilities of the three situations in simulation. The meanings of the bar graph's colors: blue: FT, red: OA, green: GOA. (a) The situation FT has the highest a-posteriori probability; (b) The situation OA has the highest a-posteriori probability; (c) The situation GOA has the highest a-posteriori probability.

little computer game experience compared to the male subjects. All the subjects finished the task successfully under the ASA mode.

When the robot was operated under the ASA mode, execution time was greater than that under the Manual mode. In fact, on average the proposed system operates at approximately 65% the speed of the non-assisted mode, as can be seen in Fig. 7.44(a). It shows the average execution time for all the successful tests (three subjects) under two modes respectively. The main reason for this is that when designing the system, much greater emphasis is placed on safety rather than speed. In practice, this means the robot will behave more cautiously, perhaps slowing down considerably to make safe movements for crossing narrow spaces, whereas a human may not decelerate to such an extent.

Fig. 7.44(b) illustrates the distribution of the minimum distances between the robot and the obstacles under two modes in all tests except the two fail cases under the Manual mode. These distances data were recorded by the LRF. As can be seen from the figure, a large part of the distance records under each mode is around $0.4m - 0.6m$ (approximately 58% under the Manual mode, and 45% under the ASA mode). This results from that the operator was required to control the robot to cross several narrow doorways in the task. Apart from this,



(a) (b)

Fig. 7.44 (a) The average time taken to manually operate the robot to finish the required task, compared with that when using the assistance mode in all the successful tests; (b) Clearance distribution graph under two modes in all tests except the two fail cases under the Manual mode.

there are more data records under the ASA mode than the Manual mode over $0.7m$, which suggests that the ASA mode tends to keep the robot further away from obstacles than the Manual mode. And even in the range of $0.4m - 0.6m$, there are less data records under the ASA mode than the Manual mode. This implies that it is more possible that the ASA mode can prevent the robot from collision when crossing the narrow doorways than the Manual mode.

Table 7.10 Mean and standard deviation of clearance data

Subject No.	Task Clearance (MEAN/STD)	
	Manual	ASA
1	0.622/0.160	**0.688/0.189**
2	0.612/0.199	**0.699/0.215**
3	0/0*	**0.696/0.208**
4	0/0*	**0.676/0.174**
5	0.629/0.184	**0.632/0.158**

* Due to the crash of the robot, the clearance is assumed to be zero.

Table 7.10 shows the means and standard deviations of the recorded clearance data of each subject in all tests under two modes respectively. Note that except two fail cases, the average clearance distances for all the other three subjects are larger under the ASA mode than the Manual mode, indicating that on average, the ASA mode maintains a larger distance from obstacles than the Manual mode. The relative large standard deviations under two modes partly result from the environment of the task, which contains both the narrow doorways and the wide areas. In addition to this, the fact, that the ASA mode confirms the full control authority of the human operator when in the Free Travel situation as expected

from its definition, has also contributed to the large standard deviations under the ASA mode, which implies that the robot assisted the human operator adaptively.

Therefore, the above results demonstrate that, with the assistance from the proposed method, the chance of a collision is smaller than that with no assistance from the robot, and the system is able to assist the human operator implicitly and adaptively.

To further confirm the necessity of the Object Approach situation and the advantages of the proposed method over the pure DWA, an object approach test was explicitly made, where the robot was operated to drive towards an object in front of it with the assistance from only DWA and the proposed method respectively. The trajectories were recorded and shown in Fig. 7.45. As can be seen, the robot tried to bypass the object with only the DWA, while the proposed method stopped the robot at a certain distance from the object by correctly recognizing the Object Approach situation in this test, which is potentially useful for the object inspection task.

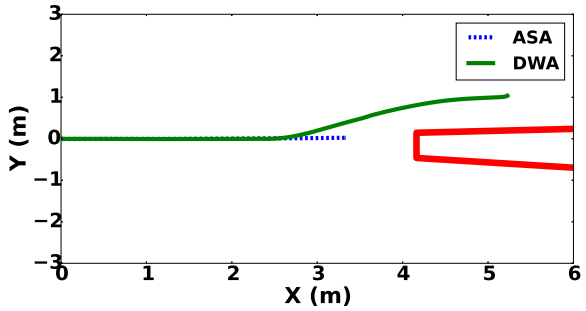


Fig. 7.45 Comparison of trajectories in the object approach test, where the red lines represent the object in this test.

The next section presents the evaluations of the proposed contextual-task aware adaptive shared autonomy system over the baseline approaches in assisting mobile robot teleoperation in a cluttered indoor scenario.

7.6 Evaluation of Contextual-Task Aware Adaptive Shared Autonomy System in Assisting Mobile Robot Teleoperation

This section focuses on evaluating the performance of the proposed contextual-task aware adaptive shared autonomy system in assisting mobile robot teleoperation. Such system is the main contribution of this thesis. To achieve this, a carefully controlled and repeatable user study with a significant number of human test participants was conducted. In this user study, each test subject was primarily required to teleoperate the sensor-equipped holonomic mobile robot (*i.e.* ASAP) to finish a navigation task in a cluttered environment. To increase the cognitive load and degrade the performance of the human operator, a secondary task was added during test to distract the human operator from the primary task, since in real teleoperation applications, the human operator usually needs to concentrate on a secondary task from time to time, thus this configuration aims to further increase the difficulty of teleoperation.

The next part will introduce the design and protocol of the user study in detail.

7.6.1 Experiment Design and Protocol

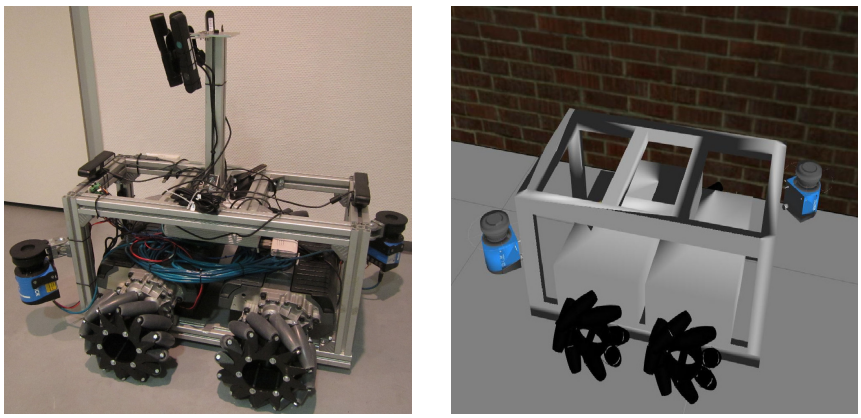
While it is significant to demonstrate new approaches on real hardware, it was observed in several pilot tests with the real robot that, using real hardware in tests can bring up difficult confounding factors, which can further weaken the repeatability and the validity of the experimental data. For example, there was often severe motion drift, when the robot was moved sideways (*e.g.* to inspect an object). Such issue usually failed the test procedure. The top speed of the robot varied slightly between test participants resulting from the different load of battery during tests. This can further undesirably affect a critical evaluation metric of the test, *i.e.* the primary task completion time of a test participant, since this metric is assumed to be dependent solely on the human factors (*e.g.* driving skill, cognitive load, *etc.*) and the interaction between the shared autonomy system and the test subjects. Additionally, as mentioned in [34], the change of daylight levels in the evaluation scenario affected the camera images observed by each test participant, when the tests were made at different times of day. This factor will also introduce undesirable effect on the performance of the test participants.

Because the test focus of this user study is the proposed shared autonomy software system and its interaction with the test participants, to eliminate these and other unnecessary confounding factors to increase the repeatability and the validity of the test, the user study was made in a high-fidelity 3D simulation in this section, and the Gazebo Simulator was employed for the simulation.

The next part will present the simulated robot, the user interface and the evaluation scenario used in the test.

7.6.1.1 Robot, user interface and test scenario

Fig. 7.46(b) shows the simulated mobile robot used in the test. It is equipped with two 2D LRFs to perceive the environments, and a RGB camera to provide images of the surroundings to the human operator[28]. Additionally, it is also installed with an IMU to improve the localization. To emulate the real sensors, Gaussian noise is added to the observations from the LRFs and the IMU during operation. For comparison, Fig. 7.46(a) displays the corresponding real robot. Please notice, the 3D sensors on the real robot were not used in the user study, hence they are not simulated here. During test, the translational speed of the robot was limited to be within $1.0\text{m}/\text{s}$, and the maximal rotational speed of the robot was set to be within $1.2\text{rad}/\text{s}$(*i.e.* around $70.0°/\text{s}$).



(a) (b)

Fig. 7.46 (a) The real robot; (b) the simulated robot model used in the test.

[28]The camera is simulated with a very small black box at the height of around 0.9m to the ground, which is hardly visible in the figure.

The human subject had no direct access to the test scenario in this user study. Instead, they observed the surroundings and the state of the robot only via user interface, as illustrated in Fig. 7.47. The user interface consists of three components, offering the human operator with: 1) camera images; 2) the map of the scenario, the position of the robot, and the laser reflections of the robot against obstacles; 3) the secondary task to complete during operation (will be detailed in the following parts of this section). To control the robot for navigation, the test participants used a Logitech F710 wireless gamepad with the information perceived from the user interface during test.

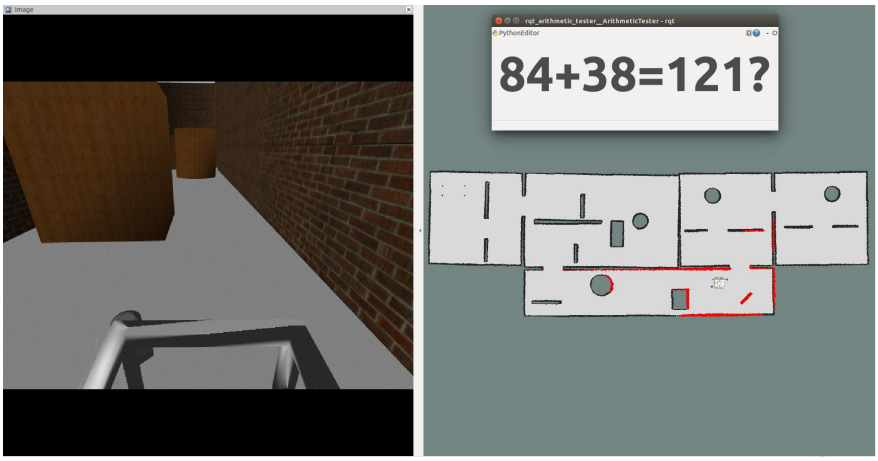
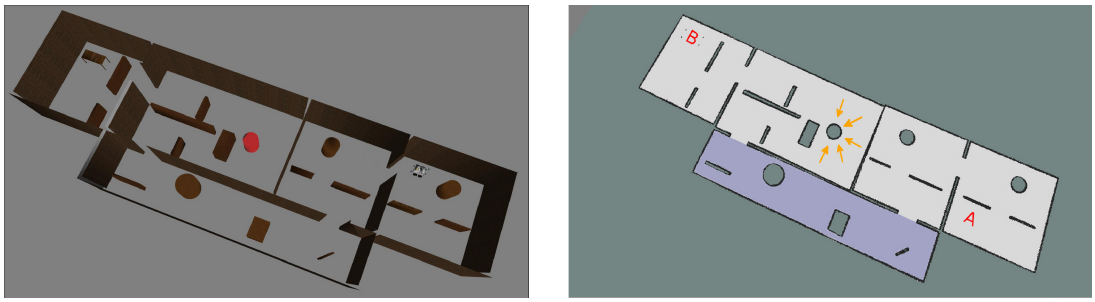


Fig. 7.47 The user interface presented to the test participants. Left: camera image. Right: 2D occupancy grid map of the test scenario, the position of the robot, the laser data reflected from the obstacles (red dots), and the window showing the secondary task.

Fig. 7.48(a) shows the simulated test scenario applied in the user study. It is a cluttered indoor scenario with narrow[29] doorways, gaps between obstacles and corridors. There is also a table for the robot to dock into. To eliminate undesired confounding factors resulting from the environmental model used by the human subject and the robot during test, all human participants were provided with the identical and complete 2D occupancy grid map of the scenario, generated before the trials by manually driving the robot around the scenario with the state-of-art SLAM algorithm in ROS, as depicted in Fig. 7.48(b). Please notice, the noise of the map results from the artificially generated sensor noise during operation.

The primary and secondary tasks in the user study will be detailed in the following part. Additionally, the different control modes employed in the test for comparison will also be introduced.

[29]There is only around 15cm in clearance considering the laser protective field of the robot.



(a) (b)

Fig. 7.48 (a) The simulated test scenario; (b) the created 2D occupancy grid map of the test scenario.

7.6.1.2 Primary and secondary Tasks, and evaluation modalities

As noted in Fig. 7.48(b), the primary task for each test participant was to control the robot to navigate from point A to point B (*i.e.* to dock into the table), then back to point A. The path between point A and point B is only one way available for the robot with no alternative routes. Additionally, each test participant was instructed to inspect the red cylinder by moving the robot sideways while facing the object, when the human subject observed it through the on-board camera during navigation, as noted by the orange arrows in Fig. 7.48(b). Therefore, to navigate through such test path, a test participant has to execute a set of contextual task types, *e.g.* doorway crossing, gap crossing, wall following, object inspection and robot docking, which is supposed to be recognized and appropriately assisted by the proposed contextual-task aware adaptive shared autonomy system.

Whenever the robot entered the predefined area in the scenario, as noted by the blue shaded region in Fig. 7.48(b), each test participant was required to consecutively do 20 secondary tasks in parallel to the primary task, which aimed to increase the cognitive load and degrade the performance of the human operator during test. More specifically, a secondary task was to judge whether an elementary arithmetic *add* or *minus* question with two two-digit numbers was correctly answered, as exampled in Fig. 7.47. The human subject simply pressed one of the two buttons on the joystick to indicate "Yes" or "No" during test. Both question and its answer were randomly created each time, and the random answer was restricted to be very close to the correct one[30] to increase the difficulty of this task. The design of such secondary task is based on the following two major considerations: 1) elementary arithmetic operation is basic knowledge of an educated person; 2) nowadays most

[30]Too extreme answer was observed to over-simplify the task.

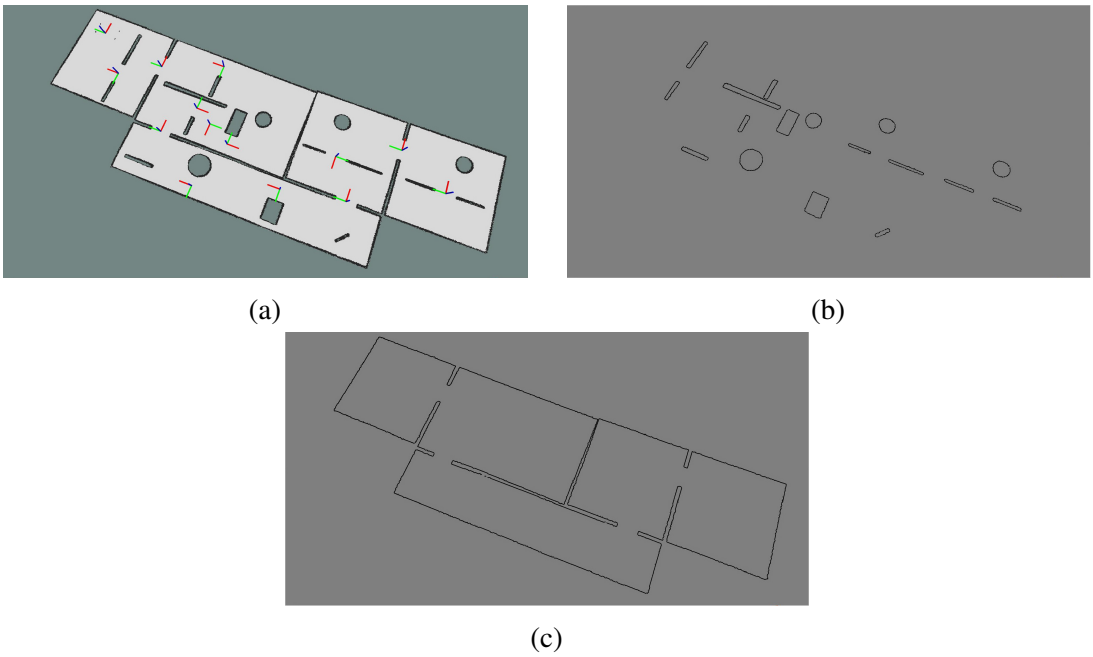
of the time people use calculator to solve such question instead of doing it mentally, hence forcing them to do it mentally can effectively increase their cognitive load and distract them from the primary task.

Each test participant was assisted by the proposed contextual-task aware shared autonomy system during the trial. Fig. 7.49 presents the extracted semantic components of the test scenario to be used by the proposed approach, *e.g.* the positions of doorways, gaps and docking table, and the point clouds of wall segments and objects. The trained DPGMM-SOGP framework (obtained in the subsection 7.4.2) was employed to classify motion pattern and recognize associated semantic component. Upon such information together with user inputs, the local intentional trajectory of the human operator was predicted with the algorithm introduced in Chapter 6. The robot motion command was generated by using the ROS-Navigation framework with the predicted trajectory and the recognized semantic components. The motion assistance was achieved by blending the user input and the robot motion command according to the classification confidence of the motion pattern. To provide safety assistance[31], the motion command sent to the robot is checked and scaled to avoid potential collision in the given motion direction by employing the dynamic window concept presented in [57].

To benchmark the performance, two other control modes were also tested in the user study. In *manual* mode, each test participant operated the robot with joystick to finish the trial without any assistance from the robot. In *reactive shared autonomy* mode, the reactive shared autonomy approach proposed in Chapter 3 was employed to assist the human operator. Such approach is *reactive* in the sense that, it provides only instantaneous motion assistance mostly pertaining to *obstacle avoidance* or *collision stopping*, since it relies on a naive environment model to generate safe motions by distinguishing simply *obstacle* from *free space* in the environment. In contrast, the proposed contextual-task aware shared autonomy approach employs a more complex environment model containing the semantic components of the environment, *e.g.* doorway, object and wall segment, which increases its cognition level. Consequently, it offers *proactive* motion assistance that lasts for a sequence of time steps to appropriately guide the human operator by recognizing the motion intention of the human operator and the associated semantic component.

The next part will introduce the procedure of the whole user study.

[31] This configuration mainly aims to address the safety issue when the recognition confidence of the system is very low, *i.e.* when the system does not yet appropriately capture the user intention.



(a) (b)

(c)

Fig. 7.49 The extracted semantic components of the test scenario: (a) the positions of the doorways, the gaps and the docking table (red: x direction, green: y direction, blue: z direction); (b) the point cloud of the objects contained in the test scenario; (c) the point cloud of the wall segments

7.6.1.3 Test procedure

A total of eighteen human subjects (aged between 24 and 32) were invited to participate into a within-groups experiment, *i.e.* each test participant performed all three trials. All test participants have at least university-level education, and an experience questionnaire before the user study reported that the majority of the human subjects have experience in driving, playing computer games or operating mobile robots.

Before the actual experiment, each test participant was briefed about this user study, *e.g.* the hardware (a laptop and a joystick), the user interface, the simulated robot, the test scenario, and the primary and secondary tasks. Then each test participant had a training course to get familiar with the user study (including the three control modes), ensuring that they possessed a common minimum knowledge regarding the test. When operating the robot to navigate in the scenario during the training course, each test participant had the direct access to the scenario when necessary (to help establish their mental model of the robot driving via user interface), and was presented with the secondary task if entering the predefined area. To proceed to the actual experiment, each test participant was required

to finish a practice obstacle task for one time with each of the three control modes, solely with the user interface, within a specific time limit, without collision and being distracted by the secondary task along the way. In addition to the training course, all human subjects were instructed to do the secondary task (*i.e.* 40 randomly created elementary arithmetic questions) separately without driving, to obtain their baseline performance.

During the actual experiment, the order of the three control modes was rotated between different test participants[32] following six different permutations, to prevent the mode bias from bringing up undesired confounding factors into the test data. All human subjects were instructed to perform the primary task as quickly and safely (*i.e.* try to avoid collision during operation) as possible. When being presented with the secondary task, the test participants were told to do it as quickly and accurately as possible, and they were explicitly instructed to give priority to the secondary task over the primary task, so they were supposed to do the primary task only if their workload supported it. Such configuration aims to prevent them from intentionally ignoring the secondary task to weaken the whole test configuration, as suggested in [34].

During each trial, a set of data and metrics were recorded to facilitate the evaluation of the test performance of the human subjects in a post-experimental analysis, including: the primary task completion time, the number of collisions during navigation, the answer time for each arithmetic question and the number of errors made in the secondary task. Meanwhile, at the end of each trial, each human participant was required to complete a NASA Task Load Index (TLX) questionnaire [81]. This is a well-known, widely-used tool to quantify the subjective workload post test, in order to evaluate a human-related technology or system.

The design and procedure of the user study have been detailed in the above parts. The next part will then present the results of the experiment with the statistical analysis.

7.6.2 Experimental Results

In order to facilitate the discussions afterwards, this subsection concentrates on statistically analyzing the recorded evaluation metrics. Since this user study is a within-groups experiment with the different control modes as the only independent variable, the repeated measures ANOVA is employed for a *post-hoc* analysis. After a *null hypothesis* is rejected by ANOVA, Fisher's Least Significant Difference (LSD) is applied for pairwise comparison to determine

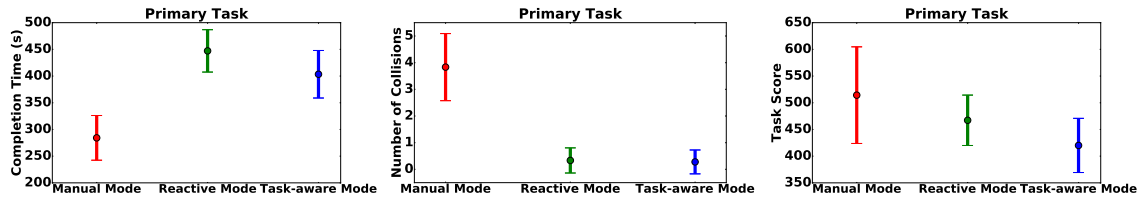
[32]The applied control mode was unknown to the test participants during trials.

the significant conditions in the data group, where the significant level is 0.05 in this statistical analysis.

For the primary task, it is clearly shown by ANOVA that the control modes have significant effects on both the *primary task completion time* ($p < 0.001$, Fig. 7.50(a)) and the *number of collisions* ($p < 0.001$, Fig. 7.50(b)). The pairwise comparison further indicates that, the human participants completed the primary task considerably faster with the manual mode than with the proposed task aware shared autonomy mode ($p < 0.0001$) and the reactive shared autonomy mode ($p < 0.0001$), and the proposed task aware shared autonomy mode performs significantly better than the reactive shared autonomy mode in terms of the primary task completion time ($p < 0.01$). Regarding the number of collisions, as observed from the pairwise comparison, both the proposed task aware shared autonomy mode and the reactive shared autonomy mode perform considerably better than the manual mode ($p < 0.0001$), although no significant difference is confirmed between these two semi-autonomous control modes in terms of safety. It was observed that within the manual mode, the test participants often had collisions when crossing narrow gaps and doorways, and trying to do the secondary task in parallel.

In order to comprehensively evaluate the primary task performance, it is required to obtain a *primary task score*. With such metric, it is supposed to be able to compare the navigation performances between different test subjects with the completion time and the number of collisions, *e.g.* one completes the assigned task very fast but with many collisions, while another one achieves a slower time yet with few collisions. Towards this aim, the primary task score is obtained by adding a time penalty, *i.e.* 60 seconds for each collision in this experiment (emphasizing the safety of the platform in remote), to the task completion time of each test subject, following the similar practice adopted in [34]. The control modes have been confirmed by ANOVA to have significant effects on the primary task scores ($p < 0.001$, Fig. 7.50(c)). Furthermore, the pairwise comparison denotes that, in terms of the primary task score, the proposed task aware shared autonomy mode outperforms considerably both the manual mode ($p < 0.001$) and the reactive shared autonomy mode ($p < 0.05$), while the reactive shared autonomy mode performs significantly better than the manual mode ($p < 0.05$).

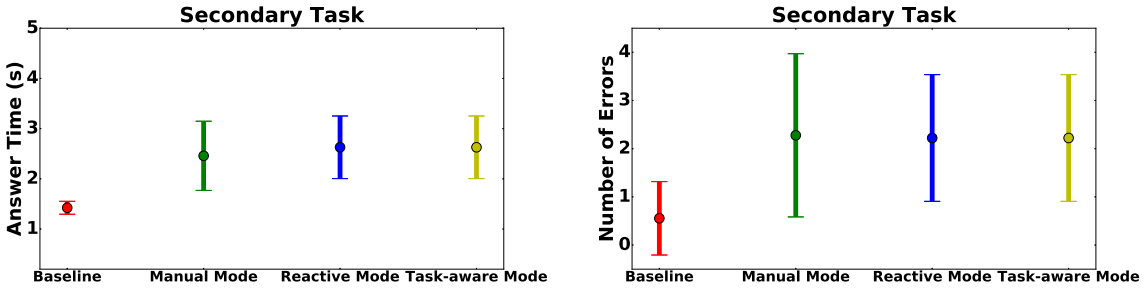
Regarding the secondary task performance, for the answer time (the average time a human participant took to answer one arithmetic question per trial), ANOVA suggests that there is a significant difference between the mean answer time with and without performing the primary task in the baseline trial ($p < 0.001$, Fig. 7.51(a)). According to the pairwise



(a) (b) (c)

Fig. 7.50 Primary task results of the three control modes. The means and standard deviations of: (a) the primary task completion time, in seconds; (b) the number of collisions during trials; (c) the task score combining the completion time and number of collisions, please notice that a *lower* score is better.

comparison, the human participants performed considerably faster in the baseline trial than with the navigation task in parallel ($p < 0.0001$), while there is no significant difference between this metric within the three control modes. With respect to the number of errors made in the secondary task, a significant difference is confirmed by ANOVA between the mean errors with and without performing the primary task in parallel ($p < 0.001$, Fig. 7.51(b)). The pairwise comparison shows that, the human participants achieved considerably fewer errors in the baseline trial than with the three control modes ($p < 0.001$), while the three control modes do not appear to have statistical differences between each other on this metric.



(a) (b)

Fig. 7.51 Secondary task results of the three control modes and the baseline trial. The means and standard deviations of: (a) the time to answer each arithmetic question, in seconds; (b) the number of errors.

For the overall NASA-TLX scores, ANOVA confirms that the three control modes have a significant effect on such metric ($p < 0.001$, Fig. 7.52). The pairwise comparison denotes that, the test subjects rated considerably lower workload with the proposed task aware shared autonomy mode than with the manual mode ($p < 0.0001$) and the reactive shared autonomy mode ($p < 0.0001$), while the other two control modes do not show statistical differences on this metric.

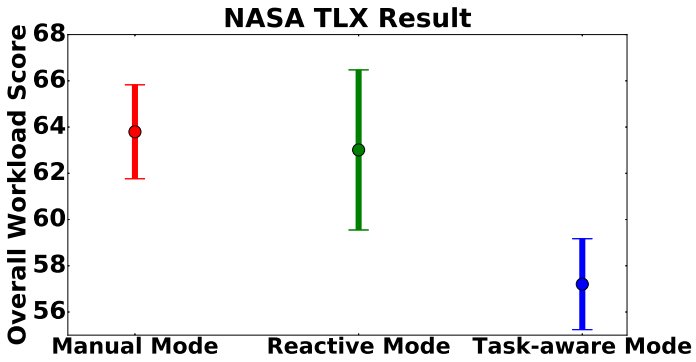


Fig. 7.52 The means and standard deviations of the overall NASA-TLX scores for the three control modes. These results describe the workload of the whole test (the primary and secondary tasks combined) being perceived and rated by the test subjects. Please notice that a *lower* score is better.

7.6.3 Discussion

Regarding the primary task, the manual mode achieves the shortest completion time, but with the cost of considerably more collision times than the other two semi-autonomous control modes. For safety reasons, the two semi-autonomous control modes are conservative on robot speed control, leading to an inherent cost of employing them in terms of the task completion time. This phenomenon was also observed in [144, 24, 26], where it took the test subjects longer time to complete the driving task with the equivalent semi-autonomous mode, compared with the manual mode. When designing a human-robot shared autonomy system, it often involves with a trade-off between speed and safety from a system perspective.

In terms of the overall primary task score, the proposed task aware shared autonomy mode significantly outperforms the other two control modes. This result confirms the effectiveness of the proposed task aware shared autonomy mode in enhancing the overall performance of the human-robot team. Both the proposed task aware shared autonomy mode and the reactive shared autonomy mode offer the collision avoidance as the basic safety guarantee, resulting into equally few collisions during trials, hence they both perform considerably better than the manual mode in terms of safety. However, the proposed task aware shared autonomy mode is able to appropriately guide the human operator to carry out intentional action primitives with the contextual information, while the reactive shared autonomy mode can only provide instantaneous low-level motion assistance pertaining to either obstacle avoidance or collision stopping. In essence, it is such difference in design strategy that results into the significant difference in their primary task performances. Apart from the statistical analysis after test, this judgement is also reinforced by observations during trials and informal conversations

with test subjects after trials. After tests with the proposed task aware shared autonomy mode, many test subjects mentioned that they clearly felt to be guided with the intentional actions, especially when crossing doorways and following wall segments, while most participants reported that they received no assistance other than actively slowing down the robot for collision avoidance, when they worked with the reactive shared autonomy mode, and it took them much effort to manually adjust the robot motions towards the target during trials.

For the secondary task, the test participants achieved significantly better performance (in terms of both the task completion time and the number of errors) during the baseline trials (*i.e.* doing the secondary task without driving). Because the test configuration required the human subjects to concentrate on the secondary task with priority over the primary task whenever it was presented, this result implies that the secondary task performance of the test participants was negatively influenced by introducing the primary navigation task during actual trials. Meanwhile, the fact that, there are no statistical differences regarding the secondary task completion time and the number of errors among control modes, suggests two conclusions: 1) different control modes do not have explicit effect on secondary task performance; 2) all the test participants performed the secondary task with the similar level of engagement during trials, which is expected from the test instruction.

The statistical analysis of the NASA-TLX denotes that there is a considerable reduction in workload with the proposed task aware shared autonomy mode compared with the other two control modes, even there were distractions brought by the secondary task for all control modes during all trials. This is because that the proposed task aware shared autonomy mode provided the human operator with motion guidance for estimated contextual action primitives during their executions, and the motion assistance was gradually and smoothly blended with the user input according to the estimation confidence. If the estimation corresponded to the intention of the human operator, the vague user input was compensated with more precise and safe robot motion command to perform the intentional contextual task (action primitive), hence the workload of the human operator was reduced. Even if sometimes the estimation was not correct and the motion assistance was observed to be inappropriate, the human operators were able to quickly adjust the robot towards the intentional track because of the smooth blending and their driving experience obtained from the training course. Moreover, the NASA-TLX results suggest that, during overall trials, the estimation corresponded to the intention of the test participant and the motion assistance was positive. In contrast, when working with the reactive shared autonomy mode and the manual mode, the human operators had to concentrate on giving precise user input to command the robot to perform the intentional contextual task all by themselves most of the time, which significantly increased

their workload and the risk to the safety of the robot due to lack of SA during operation (especially for the manual mode, as there was no assistance from the robot within this control mode at all). For this reason, many human subjects reported their strong preference for the proposed task aware shared autonomy mode over the rest two control modes after tests.

In summary, the test results from both objective and subjective metrics verify the effectiveness of the proposed contextual-task aware adaptive shared autonomy system in safely and efficiently assisting the human operator with the mobile robot teleoperation in a cluttered indoor scenario, which further confirms the merit of the most important contribution of this thesis.

7.7 Summary

This chapter extensively evaluated the proposed contextual-task aware adaptive shared autonomy system, with the focus on testing the multiple contextual task learning and recognition approaches proposed in the previous chapters respectively, since they serve as the cornerstone of this thesis. Specifically, it was to evaluate the task recognition performance of these approaches in comparison with the baseline approaches in various indoor testing scenarios, to verify that the proposed approaches are able to effectively learn human motion patterns for various task executions from demonstrations, and correctly estimate the on-going tasks the human operator performs with the contextual information and the learned models during operation.

After introducing the general settings of the conducted evaluations, this chapter began with the assessments of the proposed RBF framework combined with the batch GMR models learned from human demonstrations in both simulations and real experiments. The results with the statistical analysis proved that the proposed approach performed considerably better than the baseline approach. Then the proposed fast online GMR algorithm combined with the RBF framework for contextual task recognition was evaluated. The results demonstrated that the proposed approach is able to make the proposed shared autonomy system efficiently learn from the incremental demonstrations and recognize the intentional tasks of the human operator in an online manner. Meanwhile, it was also showed that the proposed fast online approach achieved superior task recognition performance over the batch GMR approach, and possessed real-time performance during operation.

In the following section, the effectiveness of the proposed SOGP classifier on classifying multiple contextual tasks was verified using real robot in a set of tests, and two statements were proved. First, even being sparsified from the full GP, the proposed SOGP classifier outperformed the baseline classifiers in classification accuracy on the test data, and possessed real-time performance in data processing thanks to the adopted sparsity approximation. Second, when being trained with few classes, the proposed SOGP classifier provided considerably better uncertainty estimation about the resulting class labels than the baseline classifiers, indicating that the proposed SOGP classifier is more suited to detect under-representation classes without compromising classification power, and accordingly advanced a significant step towards a life-long active learning assistive robot.

Afterwards, the performance of the proposed DPGMM-SOGP approach on unsupervised learning and estimating multiple contextual tasks was evaluated. The experimental results from the real data confirmed that, the proposed approach serves as a generic framework for representing and exploiting the knowledge of the human motion patterns performing various contextual task types without manual annotations, which is not only able to recognize the task types seen during training, but also generalizable to appropriately interpret the motion patters of task types not used for training. More importantly, the proposed approach is capable of detecting unknown motion patterns distinctive from those used in the training set thanks to the superior introspective capability of the SOGP classifier, which is highly appealing for an active learning scenario. The same section also investigated the performance of the proposed local intentional trajectory prediction algorithm with the classified motion cluster and the recognized contextual information. The qualitative test results demonstrated that, the proposed approach is able to predict with high accuracy the local trajectory the human operator intends to follow in the short term to perform various task types by utilizing the knowledge of the classified motion cluster from human demonstrations. The quantitative experiment results verified that, the proposed approach outperformed the baseline approaches in local trajectory prediction, while possessing the realtime property in learning and making estimations during operation. These results are critical for the application of this thesis, since the algorithm is supposed to generate online reference models employed by the state-of-art mobile robot motion controller, in order to obtain safe robot motion commands to assist various contextual tasks during operation. The overall evaluation results in this section verified the scalability and effectiveness of the proposed approaches in learning, recognizing and utilizing the motion pattern knowledge of the human operator in an unsupervised data-driven manner from human demonstrations, and the proposed contextual-task aware adaptive shared autonomy system is realized with them.

In the following section, the proposed reactive shared autonomy approach was evaluated with a small user study, which safely assisted the human operator with controlling a quadrotor to perform a remote navigation task in simulation. With this approach, it led to the considerable enhance of the operation safety compared with the purely manual control.

Finally, a carefully controlled and repeatable user study with 18 human test participants was conducted, aiming to evaluate the performance of the proposed contextual-task aware adaptive shared autonomy system over the baseline approaches in assisting mobile robot teleoperation in a cluttered indoor scenario. In addition to the primary navigation task, the test participants had to perform secondary tasks for some of the time simultaneously, which increased their cognitive load and hence the difficulty of the whole test. With rigorous statistical analysis, the results from both objective and subjective metrics demonstrated the advantages of the proposed contextual-task aware shared autonomy approach over either purely manual control mode or the proposed reactive shared autonomy approach for the tasks involved in the test, and verified the necessity and effectiveness of providing *proactive* motion assistance by recognizing the motion intention of the human operator and the associated semantic components in a probabilistic way, which is the core idea and contribution of this thesis.

In summary, the above experimental results confirmed the contributions of this thesis to implement and evaluate a contextual-task aware adaptive shared autonomy system to assist mobile robot teleoperation.

Chapter 8

Conclusion and Outlook

8.1 Conclusion

While being teleoperated for task execution in unstructured, unknown or even hazardous environments, a mobile robot is supposed to assist the human operator with the intelligent perception and control models in a task-appropriate way, to guarantee a safe and efficient task execution in remote. This implies that the human and the robot have to share autonomy with each other during operation. The major challenge is how to best coordinate the two sources of intelligence from the human and the robot, to achieve an adaptive and effective autonomy sharing in the context of mobile robot teleoperation. To address this challenge, the fundamental strategy of this thesis is to recognize the on-going task the human operator intends to perform to complete an action primitive based on the contextual information, *i.e.* user inputs and environment perceptions, and provide appropriate motion assistance to the human operator with the execution of the estimated task.

According to this strategy, and accounting for the uncertainties from acquiring and processing contextual information, a probabilistic contextual-task aware adaptive shared autonomy system was proposed. It infers the contextual task the human operator executes with uncertainty measurements, and appropriately assist the human operator with the recognized task based on the estimation probability. In this way, the level of autonomy is seamlessly switched between the manual control (when the probability of the task recognition is low) and the autonomous control (when the probability of the task recognition is high) in a task-appropriate way during operation. Chapter 3 mathematically formalized such

framework, and indicated that the contextual task recognition is its cornerstone. In addition to this, a reactive shared autonomy system based on BN was also reported in the same chapter. Although it is not among the major contributions, its technical limitations in assisting mobile robot teleoperation reveal possible improvements. This motivates the author of this thesis to employ a set of data-driven approaches within the proposed probabilistic shared autonomy framework, to derive the models of various contextual task executions for task recognition from human demonstrations, instead of manually building them, since the way the human operator executes a task via robot is implicit, and the data-driven approaches render the framework able to adapt to the needs of the human operator over long time in an intuitive way. The employments of the data-driven approaches for contextual task recognition constitute the major scientific contributions achieved in this thesis, and are summarized as below.

In Chapter 4, GMR models were employed to learn the motion patterns of various task types from human demonstrations in a batch way. Each task type to be estimated was described with a set of simple and salient task features. To incorporate the uncertainty of estimating multiple contextual tasks during operation, a RBF was adopted as the base framework, and combined with the learned GMR models to stably and smoothly infer the most probable operational intention of the human operator among multiple candidates over time. Furthermore, a fast online and incremental contextual task learning approach was proposed, to enable the system to learn the motion patterns online and incrementally from the demonstrations. With the incrementally incorporated demonstration dataset of each task type, a state-of-art FANN search algorithm was employed to retrieve a small training dataset closest to the current robot state, and a GMR model with a very few number of the mixture components was built online and combined with the RBF to estimate the likelihoods of the corresponding candidates.

In Chapter 5, to avoid the calibration of the time scale factor in the process model of the RBF and envision a life-long active learning scenario, contextual task recognition was formulated as classification problem, and the SOGP classifier was presented to learn from demonstrations and classify multiple contextual tasks, due to its superior introspective capability over other state-of-art classifiers, such as SVM, and scalability to large datasets. The SOGP classifier was implemented with the Gaussian regression model, resulting into closed-form solutions of the posterior predictive distribution and the model evidence respectively. The closed-form model evidence simplifies the optimization of the hyper-parameters of the model.

Chapter 6 reported an unsupervised contextual task learning and recognition approach, to relieve the workload of manually labeling the demonstration data into task types for robot to learn. The proposed approach consisted of two phases. In the first step, DPGMM was applied to cluster the human motion patterns performing different contextual task types from unannotated demonstrations, where the number of possible motion modes is inferred from the data itself instead of being manually specified *a priori* or determined through model selection. Post clustering, the proposed SOGP classifier was employed to classify the learned motion patterns during operation, due to its outstanding introspective capability and scalability to large datasets. This chapter also introduced the algorithm to predict local trajectory the human operator intends to follow in the short term to perform corresponding contextual task in a data-driven manner. It applies the proposed fast online GMR approach to classified motion cluster with recognized contextual information. To threshold the Mahalanobis distance computed with each estimated way point, it is able to achieve the trajectory prediction within a predefined tolerance bound regarding the regression outliers. This algorithm links the task recognition part and the motion assistance part of the proposed shared autonomy framework: the predicted trajectory can be employed as the reference model by the state-of-art mobile robot motion controller to generate safe motion commands, which are blended with the user inputs according to the classification confidence, to assist the human operator to carry out the intentional tasks actively and appropriately.

Finally, in Chapter 7, the proposed framework was implemented on both a simulated quadrotor and a real holonomic ground mobile platform, and a variety of experiments both in simulation and with real hardware were conducted in various indoor scenarios, to extensively evaluate two major aspects of the proposed system: 1) its performance in recognizing multiple contextual tasks with the models learned from human demonstrations, and 2) its ability to appropriately assist the human operator based on the task estimation through sharing autonomy. The experimental results demonstrated the scalability and the merit of the proposed framework.

To summarize, a novel shared autonomy system has been reported to assist mobile robot teleoperation in this thesis. It employs data-driven approaches to learn from demonstrations the motion patterns of the human operator performing various task types with contextual information, and uses the learned models to recognize the intentional task of the human operator. With the recognized tasks, the proposed shared autonomy system provides the corresponding motion assistance to the human operator in a probabilistic manner during operation. Extensive evaluations conducted in both simulations and real experiments demonstrated that the proposed framework is capable of effectively adapting behaviours through

intuitive interactions with users in both supervised and unsupervised data-driven manners, and it is able to efficiently assist the mobile robot teleoperation in a task-appropriate way. With the proposed approaches, the teleoperator can be proactively and appropriately assisted by increasing the cognition capability and the autonomy flexibility of the robot.

8.2 Outlook

Although experiments conducted in this thesis showed solid results, there are still rooms for possibilities and extensions to be developed and improved.

The proposed context-aware adaptive shared autonomy system considers only single task (action primitive) to estimate and assist. To better understand and support the user intention in a longer range, one possible extension is naturally to take a sequence of such primitive tasks into consideration, *i.e.*, to recognize the *task plan* on a higher level. To achieve this, a hierarchical model needs to be proposed to capture the spatial-temporal property when the human operator executes a sequence of primitive tasks on the low level to fulfil an abstract expectation on the high level over a certain scope. An exemplary work towards this goal can be found in the work of [142], which presents a hierarchical HMM to infer both the long and short range intentions of the user in the context of elderly walking assistance.

Additionally, the proposed system assumes that the human operator does not explicitly consider assistance when issuing input commands - and in particular, that the users do not adapt their strategy to the assistance. To alleviate this assumption in both prediction and assistance, an interesting direction is to extend the proposed model as a stochastic game, as presented in [114].

Although it is assumed in this thesis that the robot implicitly assists the human operator with the user input from a mechanical joystick, some comments from the test-participants in the user study suggest the need of devising a kind of visual or force feedback to intuitively illustrate the motion assistance provided by the robot, *i.e.* to present that how the robot predicts and actions. This echoes the message in [25]. Such kind of feedback system can help the human operator establish a better mental model of the motion assistance provided by the robot, and may increase the user acceptance of the robot and accelerate the deployment of such shared autonomy system in a human-robot team. Meanwhile, such additional feedback information is supposed to not further burden the cognitive load of the human operator. Therefore, it indicates another interesting research direction in the future.

Moreover, the model for contextual-task recognition is learned with handcrafted features from human demonstrations in this thesis. To improve this to prevent the requirement of manually designing task features, the *Deep-Learning* [109] technique provides a potential solution, where it is able to extract multi-level features from raw input information (*e.g.* raw sensor data) directly without any engineering design beforehand, thanks to the multiple convolutional layers for abstraction of learning representations from the given data. To apply the Deep-Learning technique in the context of shared autonomy between human and robot, *multimodal learning* has to be considered, *i.e.* to learn task patterns for assistance from a variety of data sources, *e.g.* LRF, camera and user input device like joystick. Meanwhile, shared autonomy involves with decision making based on recognition when assisting human operators. By employing the Deep-Learning framework in a shared autonomy system, several challenges regarding robot control remain to be addressed, such as real-time requirement for data processing, accomodation of uncertainties from perception and action, and generalization of the Deep-Learning control, as indicated in [180].

Apart from assisting mobile robot teleoperation, the presented idea and methods of autonomy sharing by contextual task recognition are generally applicable to other scenarios where a human drives a mobile robot. For example, the proposed contextual task recognition approaches can be applied to recognize and predict the driving intentions of an intelligent wheelchair user in indoor scenarios with the corresponding contextual information. When applying the proposed approach in a social environment, it is also supposed to consider the intentions of other people in the surroundings to better assist the user of the mobility system to navigate for task execution in such environment, such as to accompany a person or to avoid collision against a person in a socially acceptable way, whereas the proposed approach in this thesis focuses on inferring the task intention of only the user of the system. This arouses interesting future research directions regarding the social intention recognition and supporting, such as reported in [127]. Similarly, the proposed contextual-task aware adaptive shared autonomy system has the potential applications in the scenario of intelligent vehicle for driver assistance, where the vehicle is supposed to detect the driving intention of the driver as early as possible from a set of contextual information, such as the user input and the road situation, and provide the in-time correct assistance to promote the safety and efficiency of the driving condition, such as presented in [93].

Algorithms to Extract Semantic Elements from Map Image

For computing the task features of candidate tasks, which are employed by data-driven approaches for task recognition, the environment is processed to obtain the information regarding its semantic elements beforehand in this thesis, *e.g.* the positions of doorways, and the surface points of candidate objects and wall segments, due to the lack of open-source solutions for simultaneous mapping and semantic element localization. In this appendix, this environment process algorithm will be presented in the following as a supplementary document to this thesis.

The proposed algorithm acts on 2D occupancy grid maps which can be easily constructed from the state-of-art SLAM approaches with a 2D LRF mounted on a mobile platform. It reads in the map data as a gray-scale image (called a *map image* afterwards). The pixel values of the map image for an occupied grid cell, a free one, and an unknown one are 0, 254 and 127 respectively. Algorithms 4 and 5 present the primary computational steps with the map image used for segmenting objects and detecting doorways respectively.

Object Segmentation: the implementations from OpenCV [18] are used to pre-process the map image: firstly two standard image filtering operations: the *Morphological Gradient* and the *Binary Threshold*, are applied sequentially to the map image to retain its possible outlines; then the contours of the potential objects are retrieved in the map image with the *findContour* function. The contours whose distances to each other are under a certain threshold τ_d are merged.[1] After this step, the candidate objects (line 16 of Algorithm 4) and their surface points C are obtained, which can be used to calculate the related task features.

[1] The distance between two contours refers to the minimal Euclidean distance between the surface points of the two contours.

Algorithm 4 *2D Object Segmentation*

Given gray-scale map image I
initialize $O \leftarrow \{\}, C \leftarrow \{\}$
$I \leftarrow \text{MorphologyGradient}(I)$
$I \leftarrow \text{BinaryThreshold}(I)$
$C \leftarrow \text{FindContour}(I)$
for $i = 0; c_i \in C; i$++ **do**
  for $j = i+1; c_j \in C; j$++ **do**
    $d_j = \left\| c_i, c_j \right\|$
    if $d_j < \tau_d$ **then**
      $c_i \leftarrow c_i \cup c_j$
      $C \leftarrow C \setminus \{c_j\}$
    end if
  end for
end for
for each $c \in C$ **do**
  $o_c = \frac{1}{K} \sum_{i=1}^{K} s_i, s_i \in c$
  $O \leftarrow O \cup \{o_c\}$
end for
return O, C
Key: $\|\cdot\|$ = distance (Euclidean)
τ_d = distance threshold for merging contours

Doorway/Gap Detection: A doorway is interpreted as a gap between two objects in this thesis. As gaps might exist inside a big object, to effectively detect gaps, firstly each segmented object is splitted into n parts based on the azimuth of each surface point of the object with respect to the center of the object. Then the distances of these split objects to each other are calculated. The potential gaps are preliminarily detected with the width criterion: those gaps whose widths are within a range $(d_{\min}, d_{\max})$ will be retained. To filter out false gaps, the ratio of free grid cells along the line formed by the two end points of each candidate gap is calculated. Those gaps whose free cell ratio is over a threshold τ_{free} are considered as the final gaps (there should be totally free space within a gap, but a threshold is used to accommodate map noise).[2] The detected gaps will then be used for the related task feature calculation.[3]

[2] In practice, after this step, the obtained gaps whose distances to each other are under a threshold $\tau_{\text{gap\_dist}}$ (in the implementation of this thesis, it was manually set to 0.75m) are merged. The distance between two gaps are calculated using their center points.

[3] In the implementation of this thesis, the following parameter values were arrived at experimentally: $\tau_d = 1.0\text{m}, n = 24, d_{\min} = 1.1\text{m}, d_{\max} = 1.4\text{m}, \tau_{\text{free}} = 0.95$.

For detecting wall segments, the open-sourced method proposed in [17] is employed to firstly segment the whole environment into individual rooms whose contours represent the walls around them. Then the proposed object segmentation algorithm is applied to extract the biggest contour of each room to obtain its wall segments. Benefiting from this procedure, the doorways linking the segmented rooms can be obtained, and objects and gaps in each room (or in the room where the test is carried out) can be extracted, instead of doing the extractions over the whole map. This improves the computational efficiency.

The following figures illustrate the extraction results by applying the proposed algorithm to various occupancy grid maps with the size: 1024×1024 grid cells and the resolution $0.05m$ per cell.

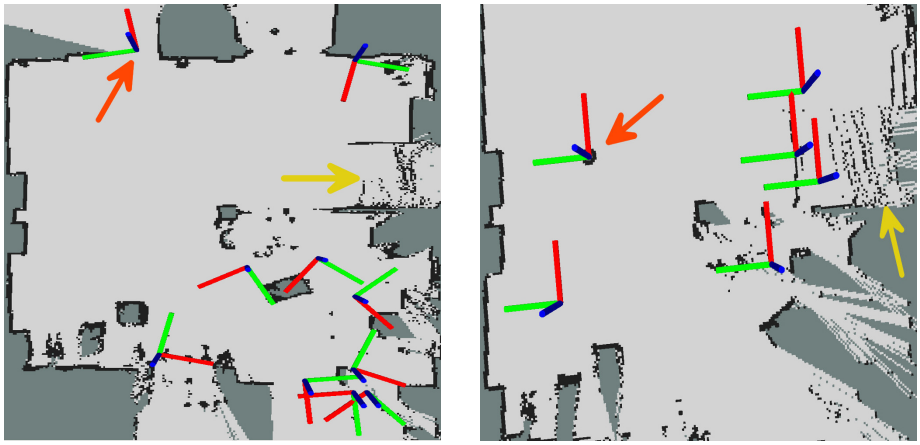
Fig. 1(a) shows part of the detection results of the scenario used for demonstrating the doorway crossing task. Apart from the required target (indicated by the orange arrow), the algorithm outputs many false candidate doorways, most of which are in fact gaps, since the doorway is interpreted as gap in this algorithm. A doorway on the right side of the required target was ignored because of the width criterion for doorway detection in the algorithm.[4]

Fig. 1(b) displays part of the segmentation results of the scenario employed for demonstrating the object inspection task. The required target is also noted by the orange arrow. Some false candidate objects were produced due to the noise of the map, the others were actually the walls of the scenario. Two major sources of the map noise are a small slope and a stairway beside in this scenario, and they are indicated in the figure by a yellow arrow. (The same kind of map noise can also be observed in Fig. 1(a).)

Fig. 2 shows the detection results of the algorithm applied to another scenario used for evaluation in this thesis. Fig. 2(a) displays two doorways extracted after segmenting the map into rooms by applying the approach proposed in [17], thus there are no gaps detected[5]. Fig. 2(b) illustrates three object segments and the wall segments in the form of point clouds together with the detected doorways within the room in which the test was performed. The white point cloud represents the wall segments, while the others denote the three object segments respectively.

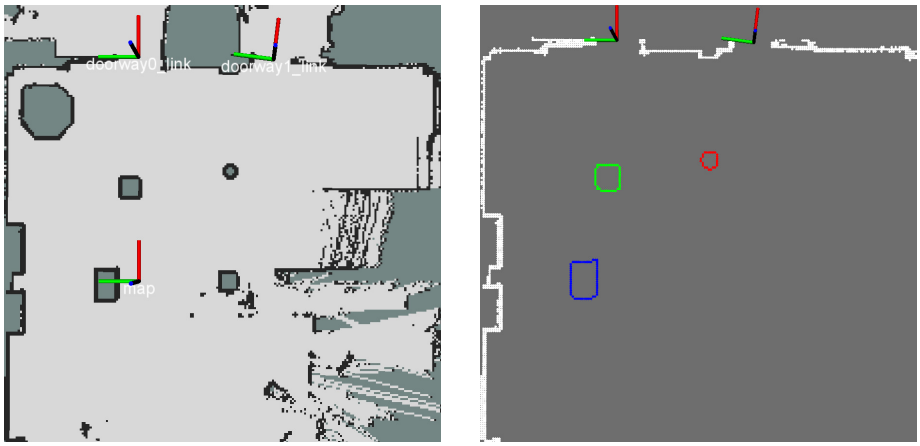
[4] Although the "gap" next to this one was also filtered out due to the width criterion, but in fact it is a glass wall with a steel frame which was detected by the LRF of the robot.

[5] Please notice, the coordinate system in the center of this map image indicates the original point of the map instead of a segmented doorway.



(a) (b)

Fig. 1 The detection results of the environment process algorithm applied to two different scenarios. The orange arrows indicate the required targets in each of the scenarios, and the yellow ones shows the map noise resulting from a small slope and a stairway beside of the scenarios: (a) The doorway detection results. A candidate doorway is represented by a colored coordinate system at the center point of it. The x, y, z axes of the coordinate system are represented by red, green, blue colors respectively; (b) The object segmentation results. A candidate object is displayed in the same way as the doorway candidates.



(a) (b)

Fig. 2 The extraction results of the environment process algorithm applied to another scenario used for evaluation in this thesis: (a) Two doorways were detected after segmenting the environment into rooms. Each doorway is indicated by a coordinate system fixed on its center point with the same color coding as in Fig. 1, except the one in the center representing the original point of the map; (b) The point clouds of the wall segments and three objects within the room where the test was performed. The white point cloud represents the wall segments, while the others denote the three object segments respectively. The detected doorways are also shown together.

Algorithm 5 *2D Doorway/Gap Detection*

Given grey-scale map image I and segmented object contours C
initialize $D \leftarrow \{\}$, $C_{\text{split}} \leftarrow \{\}$
for each $c \in C$ **do**
  for $i = 0; i < n; i$++ **do**
    for each $s \in c$ **do**
      if azimuth$(s) \in [\frac{2\pi}{n} i, \frac{2\pi}{n}(i+1)]$ **then**
        $c_{\text{split\_i}} \leftarrow c_{\text{split\_i}} \cup \{s\}$
      end if
    end for
    $C_{\text{split}} \leftarrow C_{\text{split}} \cup \{c_{\text{split\_i}}\}$
  end for
end for
for $i = 0; c_i \in C_{\text{split}}; i$++ **do**
  for $j = i+1; c_j \in C_{\text{split}}; j$++ **do**
    $d_j = \left\| c_i, c_j \right\|$
    if $d_j \in [d_{\min}, d_{\max}]$ **then**
      $\text{gap}_{\text{preliminary}} \leftarrow \{p_{\text{start}}, p_{\text{end}}\}$, where:
      $p_{\text{start}} \in c_i, p_{\text{end}} \in c_j$, and $d_j == \|p_{\text{start}}, p_{\text{end}}\|$
    end if
  end for
end for
for each gap $\in \text{gap}_{\text{preliminary}}$ **do**
  $\{p_{\text{start}}, p_{\text{end}}\} \leftarrow$ gap
  Extract points along the straight line $(p_{\text{start}}, p_{\text{end}})$ $(\rightarrow P_{\text{line}})$.
  initialize $\text{Num}_{\text{free\_grid}} = 0$
  for each $p \in P_{\text{line}}$ **do**
    if $I_p == 254$ **then**
      $\text{Num}_{\text{free\_grid}}$++
    end if
  end for
  if $\frac{\text{Num}_{\text{free\_grid}}}{|P_{\text{line}}|} > \tau_{\text{free}}$ **then**
    $\text{gap}_{\text{final}} \leftarrow \{\text{position} = \frac{p_{\text{start}} + p_{\text{end}}}{2}, \text{endpoints} = \{p_{\text{start}}, p_{\text{end}}\}\}$
    $D \leftarrow D \cup \text{gap}_{\text{final}}$
  end if
end for
return D

Key: $|\cdot|$ = size (number of elements)
$\|\cdot\|$ = distance (Euclidean)
n = number of parts for an object to be divided into
$d_{\min}, d_{\max}$ = min/max thresholds on gap width
τ_{free} = threshold of free space percent for filtering gaps

List of Publications

Peer-reviewed international journal:

Gao, M. and Zöllner, J. M. Sparse Contextual Task Learning and Classification to Assist Mobile Robot Teleoperation with Introspective Estimation. Journal of Intelligent & Robotic Systems, 2017. Springer

Peer-reviewed international conference:

Gao, M. and Zöllner, J. M. Local Contextual Trajectory Estimation with Demonstration for Assisting Mobile Robot Teleoperation. In European Conference on Mobile Robots (ECMR), 2017. IEEE

Gao, M., Kohlhaas, R., and Zöllner, J. M. Contextual Learning and Sharing Autonomy to Assist Mobile Robot by Trajectory Prediction (late breaking abstract). In Safety, Security and Rescue Robotics (SSRR), 2016 lEEE International Symposium on. IEEE

Gao, M., Kohlhaas, R., and Zöllner, J. M. Unsupervised Contextual Task Learning and Recognition for Sharing Autonomy to Assist Mobile Robot Teleoperation. In Informatics in Control, Automation and Robotics (INCO), 2016 13th International Conference on.

Gao, M., Schamm, T., and Zöllner, J. M. Contextual Task Recognition for Sharing Autonomy to Assist Mobile Robot Teleoperation with Introspective Estimation using Gaussian Process. In Autonomous Robot Systems and Competitions (ARSC), 2016 lEEE International Conference on. IEEE

Gao, M., Schamm, T., and Zöllner, J. M. Online and Incremental Contextual Task Learning and Recognition for Sharing Autonomy to Assist Mobile Robot Teleoperation. In Robotics

and Biomimetics (ROBIO), 2015 IEEE International Conference on. IEEE

Gao, M., Schamm, T., and Zöllner, J. M. Multiple Contextual Task Recognition for Sharing Autonomy to Assist Mobile Robot Teleoperation. In Intelligent Robotics and Application (IRA), 2015 International Conference on, volume 9245, pages 3–14. Springer

Gao, M., Oberländer, J., Schamm, T., and Zöllner, J. M. Shared autonomy for assisted mobile robot teleoperation by recognizing operator intention as contextual task. In Robotics and Biomimetics (ROBIO), 2014 IEEE International Conference on, pages 82–87. IEEE

Gao, M., Oberländer, J., Schamm, T., and Zöllner, J. M. Contextual Task-Aware Shared Autonomy for Assistive Mobile Robot Teleoperation. In Intelligent Robots and Systems (IROS), 2014 IEEE/RSJ International Conference on, pages 3311–3318. IEEE

Workshop:

Gao, M., Heppner, G., Schamm, T., and Zöllner, J. M. A Reactive Shared Autonomy Method for Assisted Mobile Robot Teleoperation based on Bayesian Approach. In Proceedings of the 3rd Telerobotics Workshop, IEEE/RSJ Int. Conf. on Intelligent Robots and Systems-IROS2014, pages 40–45.

List of Figures

List of Tables

Bibliography

[1] Aarno, D., Ekvall, S., and Kragic, D. (2005). Adaptive virtual fixtures for machine-assisted teleoperation tasks. *IEEE International Conference on Robotics and Automation*.

[2] Aarno, D. and Kragic, D. (2006). Layered HMM for motion intention recognition. In *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, pages 5130–5135. IEEE.

[3] Aarno, D. and Kragic, D. (2008). Motion intention recognition in robot assisted applications. *Robotics and Autonomous Systems*, 56(8):692–705.

[4] Abbink, D. A., Mulder, M., and Boer, E. R. (2012). Haptic shared control: smoothly shifting control authority? *Cognition, Technology & Work*, 14(1):19–28.

[5] Aigner, P. and McCarragher, B. (1997). Human integration into robot control utilising potential fields. In *Robotics and Automation, 1997. Proceedings., 1997 IEEE International Conference on*, volume 1, pages 291–296. IEEE.

[6] Alexander, H., Demeester, E., Vanacker, G., Vanhooydonck, D., Philips, J., Brussel, H. V., and Nuttin, M. (2007). Bayesian Plan Recognition and Shared Control under Uncertainty : Assisting Wheelchair Drivers by Tracking Fine Motion Paths. In *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, pages 3360–3366. IEEE.

[7] Allen, J. E. and Guinn, C. I. (1999). Mixed-initiative interaction. *Intelligent Systems and their Applications, IEEE*, 14(5):14–23.

[8] Anderson, S. J., Karumanchi, S. B., Iagnemma, K., and Walker, J. M. (2013). The intelligent copilot: A constraint-based approach to shared-adaptive control of ground vehicles. *Intelligent Transportation Systems Magazine, IEEE*, 5(2):45–54.

[9] Anderson, S. J., Walker, J. M., and Iagnemma, K. (2014). Experimental Performance Analysis of a Homotopy-Based Shared Autonomy Framework. *Human-Machine Systems, IEEE Transactions on*, 44(2):190–199.

[10] Argall, B. D., Chernova, S., Veloso, M., and Browning, B. (2009). A survey of robot learning from demonstration. *Robotics and autonomous systems*, 57(5):469–483.

[11] Arkin, R. (1998). *Behavior-Based Robotics*. MIT press.

[12] Bahrmann, F., Hellbach, S., and Böhme, H.-J. (2012). Please tell me where I am: A fundament for a semantic labeling approach. In *35th German Conference on Artificial Intelligence September 24-27, 2012 Saarbrucken, Germany*, page 120. Citeseer.

[13] Berczi, L.-P., Posner, I., and Barfoot, T. D. (2015). Learning to assess terrain from human demonstration using an introspective Gaussian-process classifier. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pages 3178–3185. IEEE.

[14] Billings, C. E. (1997). *Aviation automation: The search for a human-centered approach*.

[15] Birkenkampf, P., Leidner, D., and Borst, C. (2014). A knowledge-driven shared autonomy human-robot interface for tablet computers. In *Humanoid Robots (Humanoids), 2014 14th IEEE-RAS International Conference on*, pages 152–159. IEEE.

[16] Blei, D. M., Jordan, M. I., and Others (2006). Variational inference for Dirichlet process mixtures. *Bayesian analysis*, 1(1):121–143.

[17] Bormann, R., Jordan, F., Li, W., Hampp, J., and Haegele, M. (2016). Room Segmentation: Survey, Implementation, and Analysis. In *Robotics and Automation (ICRA), 2016 IEEE International Conference on*. IEEE.

[18] Bradski, G. and Kaehler, A. (2008). *Learning OpenCV: Computer Vision with the OpenCV Library*, volume 1. " O'Reilly Media, Inc.".

[19] Britton, N., Yoshida, K., Walker, J., Nagatani, K., Taylor, G., and Dauphin, L. (2015). Lunar micro rover design for exploration through virtual reality tele-operation. In *Field and Service Robotics*, pages 259–272. Springer.

[20] Brunner, B., Hirzinger, G., Landzettel, K., and Heindl, J. (1993). Multisensory shared autonomy and tele-sensor-programming - Key issues in the space robot technology experiment ROTEX. In *Proceedings of 1993 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '93)*, volume 3, pages 1–15. IEEE.

[21] Burridge, R. R. and Hambuchen, K. A. (2009). Using prediction to enhance remote robot supervision across time delay. In *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pages 5628–5634. IEEE.

[22] Buschka, P. and Saffiotti, A. (2002). A virtual sensor for room detection. In *Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference on*, volume 1, pages 637–642. IEEE.

[23] Calinon, S. and Billard, A. (2007). Incremental learning of gestures by imitation in a humanoid robot. In *Proceedings of the ACM/IEEE international conference on Human-robot interaction*, pages 255–262. ACM.

[24] Carlson, T. and Demiris, Y. (2008). Human-wheelchair collaboration through prediction of intention and adaptive assistance. *2008 IEEE International Conference on Robotics and Automation*, pages 3926–3931.

[25] Carlson, T. and Demiris, Y. (2010). Increasing robotic wheelchair safety with collaborative control: Evidence from secondary task experiments. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 5582–5587. IEEE.

[26] Carlson, T. and Demiris, Y. (2012). Collaborative control for a robotic wheelchair: evaluation of performance, attention, and workload. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 42(3):876–888.

[27] Carlson, T., Leeb, R., Chavarriaga, R., Millan, D. R., and Others (2012a). Online modulation of the level of assistance in shared control systems. In *Systems, Man, and Cybernetics (SMC), 2012 IEEE International Conference on*, pages 3339–3344. IEEE.

[28] Carlson, T., Leeb, R., Chavarriaga, R., and Millán, J. d. R. (2012b). The birth of the brain-controlled wheelchair. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 5444–5445. IEEE.

[29] Carlson, T. and Millán, J. d. R. (2013). Brain-Controlled Wheelchairs: A Robotic Architecture. *IEEE ROBOTICS & AUTOMATION MAGAZINE*, 20(1):65–73.

[30] Carlson, T., Monnard, G., Leeb, R., and Millán, J. d. R. (2011). Evaluation of proportional and discrete shared control paradigms for low resolution user inputs. In *Systems, Man, and Cybernetics (SMC), 2011 IEEE International Conference on*, pages 1044–1049. IEEE.

[31] Carlson, T., Tonin, L., Perdikis, S., Leeb, R., Millan, D. R., and Others (2013). A hybrid BCI for enhanced control of a telepresence robot. In *Engineering in Medicine and Biology Society (EMBC), 2013 35th Annual International Conference of the IEEE*, pages 3097–3100. IEEE.

[32] Cederborg, T., Li, M., Baranes, A., and Oudeyer, P.-Y. (2010). Incremental Local Inline Gaussian Mixture Regression for Imitation Learning of Multiple Tasks. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pages 267–274.

[33] Chang, C.-C. and Lin, C.-J. (2011). A Library for Support Vector Machines. *ACM Transactions on Interlligent Systems and Technology (TIST)*, 2(3):39.

[34] Chiou, M., Stolkin, R., Bieksaite, G., Hawes, N., Shapiro, K. L., and Harrison, T. S. (2016). Experimental analysis of a variable autonomy framework for controlling a remotely operating mobile robot. In *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*, pages 3581–3588. IEEE.

[35] Clanton, S. T., Rasmussen, R. G., Zohny, Z., and Velliste, M. (2013). Generalized virtual fixtures for shared-control grasping in brain-machine interfaces. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pages 323–328. IEEE.

[36] Crandall, J. W. and Goodrich, M. A. (2002). Characterizing efficiency of human robot interaction: A case study of shared-control teleoperation. In *Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference on*, volume 2, pages 1290–1295. IEEE.

[37] Csató, L. and Opper, M. (2002). Sparse on-line gaussian processes. *Neural computation*, 14(3):641–668.

[38] De Maesschalck, R., Jouan-Rimbaud, D., and Massart, D. L. (2000). The mahalanobis distance. *Chemometrics and intelligent laboratory systems*, 50(1):1–18.

[39] Debus, T., Stoll, J., Howe, R. D., and Dupont, P. (2001). Cooperative human and machine perception in teleoperated assembly. In *Experimental Robotics VII*, pages 51–60. Springer.

[40] Demeester, E., Hüntemann, A., Vanhooydonck, D., Vanacker, G., Degeest, A., Van Brussel, H., and Nuttin, M. (2006). Bayesian estimation of wheelchair driver intents: Modeling intents as geometric paths tracked by the driver. In *IEEE International Conference on Intelligent Robots and Systems*, pages 5775–5780. IEEE.

[41] Demeester, E., Hüntemann, A., Vanhooydonck, D., Vanacker, G., Van Brussel, H., and Nuttin, M. (2008). User-adapted plan recognition and user-adapted shared control: A Bayesian approach to semi-autonomous wheelchair driving. *Autonomous Robots*, 24(2):193–211.

[42] Demeester, E., Nuttin, M., Vanhooydonck, D., and Brussel, H. V. (2003). A model-based, probabilistic framework for plan recognition in shared wheelchair control: experiments and evaluation. In *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003) (Cat. No.03CH37453)*, volume 2, pages 1456–1461. IEEE.

[43] Demeester, E., Nuttin, M., Vanhooydonck, D., Vanacker, G., and Van Brussel, H. (2005). Global Dynamic Window Approach for Arbitrarily Shaped Holonomic and Non-holonomic Mobile Robots. In *IEEE/RSJ Int. Conference on Intelligent Robots and Systems*, pages 2357–2362.

[44] Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 1–38.

[45] Dixon, K. R. and Khosla, P. K. (2004). Learning by observation with mobile robots: A computational approach. In *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, volume 1, pages 102–107. IEEE.

[46] Dragan, A. and Srinivasa, S. (2012). Formalizing Assistive Teleoperation. *Robotics Science and Systems*, pages 1–6.

[47] Dragan, A. D. and Srinivasa, S. S. (2013). A policy-blending formalism for shared control. *The International Journal of Robotics Research*, 32(7):790–805.

[48] Escolano, C., Antelis, J. M., and Minguez, J. (2012). A telepresence mobile robot controlled with a noninvasive brain–computer interface. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 42(3):793–804.

[49] Fabrizi, E. and Saffiotti, A. (2000). Extracting topology-based maps from gridmaps. In *Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on*, volume 3, pages 2972–2978. IEEE.

[50] Fagg, A. H., Rosenstein, M., Platt, R., and Grupen, R. A. (2004). Extracting user intent in mixed initiative teleoperator control. In *Proc. American Institute of Aeronautics and Astronautics Intelligent Systems Technical Conference*.

[51] Ferber, J. (1999). *Multi-agent systems: an introduction to distributed artificial intelligence*, volume 1. Addison-Wesley Reading.

[52] Fernandez-Carmona, M., Fernandez-Espejo, B., Peula, J. M., Urdiales, C., and Sandoval, F. (2009). Efficiency based collaborative control modulated by biometrics for wheelchair assisted navigation. In *Rehabilitation Robotics, 2009. ICORR 2009. IEEE International Conference on*, pages 737–742. IEEE.

[53] Fischler, M. A. and Bolles, R. C. (1981). Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395.

[54] Fong, T. and Thorpe, C. (2001). Vehicle teleoperation interfaces. *Autonomous robots*, pages 9–18.

[55] Fong, T., Thorpe, C., and Baur, C. (2001). Advanced interfaces for vehicle teleoperation: Collaborative control, sensor fusion displays, and remote driving tools. *Autonomous Robots*, pages 77–85.

[56] Fong, T., Thorpe, C., and Baur, C. (2003). Robot, asker of questions. *Robotics and Autonomous Systems*, 42(3-4):235–243.

[57] Fox, D., Burgard, W., and Thrun, S. (1997). The dynamic window approach to collision avoidance. *Robotics & Automation Magazine*, pages 1–23.

[58] Franchi, A., Secchi, C., Ryll, M., Bulthoff, H. H., and Robuffo Giordano, P. (2012). Shared control: Balancing autonomy and human assistance with a group of quadrotor UAVs. *Robotics & Automation Magazine, IEEE*, 19(3):57–68.

[59] Fu, J. and Topcu, U. (2015). Pareto efficiency in synthesizing shared autonomy policies with temporal logic constraints. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pages 361–368. IEEE.

[60] Gao, M., Heppner, G., Schamm, T., and Zöllner, J. M. (2014a). A Reactive Shared Autonomy Method for Assisted Mobile Robot Teleoperation based on Bayesian Approach. In *Proceedings of the 3rd Telerobotics Workshop, IEEE/RSJ Int. Conf. on Intelligent Robots and Systems-IROS2014*, pages 40–45.

[61] Gao, M., Kohlhaas, R., and Zöllner, J. M. (2016a). Contextual learning and sharing autonomy to assist mobile robot by trajectory prediction. In *Safety, Security, and Rescue Robotics (SSRR), 2016 IEEE International Symposium on*, pages 274–275. IEEE.

[62] Gao, M., Kohlhaas, R., and Zöllner, J. M. (2016b). Unsupervised Contextual Task Learning and Recognition for Sharing Autonomy to Assist Mobile Robot Teleoperation. In *Informatics in Control, Automation and Robotics (INCO), 2016 13th International Conference on*.

[63] Gao, M., Oberländer, J., Schamm, T., and Zöllner, J. M. (2014b). Contextual Task-Aware Shared Autonomy for Assistive Mobile Robot Teleoperation. In *Intelligent Robots and Systems (IROS), 2014 IEEE/RSJ International Conference on*, pages 3311–3318. IEEE.

[64] Gao, M., Oberländer, J., Schamm, T., and Zöllner, J. M. (2014c). Shared autonomy for assisted mobile robot teleoperation by recognizing operator intention as contextual task. In *2014 IEEE International Conference on Robotics and Biomimetics, IEEE ROBIO 2014*, pages 82–87. IEEE.

[65] Gao, M., Schamm, T., and Zöllner, J. M. (2015a). Multiple Contextual Task Recognition for Sharing Autonomy to Assist Mobile Robot Teleoperation. In *Intelligent Robotics and Application (IRA), 2015 International Conference on*, volume 9245, pages 3–14. Springer.

[66] Gao, M., Schamm, T., and Zöllner, J. M. (2015b). Online and Incremental Contextual Task Learning and Recognition for Sharing Autonomy to Assist Mobile Robot Teleoperation. In *Robotics and Biomimetics (ROBIO), 2015 IEEE International Conference on*. IEEE.

[67] Gao, M., Schamm, T., and Zöllner, J. M. (2016c). Contextual Task Recognition for Sharing Autonomy to Assist Mobile Robot Teleoperation with Introspective Estimation using Gaussian Process. In *Autonomous Robot Systems and Competitions (ARSC), 2016 lEEE International Conference on*. IEEE.

[68] Gao, M. and Zöllner, J. M. (2017a). Local contextual trajectory estimation with demonstration for assisting mobile robot teleoperation. In *Mobile Robots (ECMR), 2017 European Conference on*. IEEE.

[69] Gao, M. and Zöllner, J. M. (2017b). Sparse contextual task learning and classification to assist mobile robot teleoperation with introspective estimation (in press). *Journal of Intelligent & Robotic Systems*.

[70] Gnatzig, S., Schuller, F., and Lienkamp, M. (2012). Human-machine interaction as key technology for driverless driving-A trajectory-based shared autonomy control approach. In *RO-MAN, 2012 IEEE*, pages 913–918. IEEE.

[71] Goertz, R. C. (1963). Manipulators used for handling radioactive materials. *Human factors in technology*, pages 425–443.

[72] Goil, A., Derry, M., and Argall, B. D. (2013). Using machine learning to blend human and robot controls for assisted wheelchair navigation. In *Rehabilitation Robotics (ICORR), 2013 IEEE International Conference on*, pages 1–6. IEEE.

[73] Göller, M., Steinhardt, F., Kerscher, T., Dillmann, R., Devy, M., Germa, T., and Lerasle, F. (2010). Sharing of control between an interactive shopping robot and it's user in collaborative tasks. In *RO-MAN, 2010 IEEE*, pages 626–631. IEEE.

[74] Goodrich, M. a. and Schultz, A. C. (2007). Human-Robot Interaction: A Survey. *Foundations and Trends® in Human-Computer Interaction*, 1(3):203–275.

[75] Griffiths, P. G. and Gillespie, R. B. (2005). Sharing Control Between Humans and Automation Using Haptic Interface: Primary and Secondary Task Performance Benefits. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 47(3):574–590.

[76] Grimmett, H., Paul, R., Triebel, R., and Posner, I. (2013). Knowing When We Don{'}t Know: Introspective Classification for Mission-Critical Decision Making. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 4531–4538. IEEE.

[77] Grollman, D. H. (2010). *Teaching old dogs new tricks: Incremental multimap regression for interactive robot learning from demonstration*. PhD thesis, Brown University.

[78] Grollman, D. H. and Jenkins, O. C. (2007). Dogged learning for robots. In *Robotics and Automation, 2007 IEEE International Conference on*, pages 2483–2488. IEEE.

[79] Grollman, D. H. and Jenkins, O. C. (2008). Sparse incremental learning for interactive robot control policy estimation. In *International Conference on Robotics and Automation*, pages 3315–3320. IEEE.

[80] Grollman, D. H. and Jenkins, O. C. (2010). Incremental learning of subtasks from unsegmented demonstration. In *IEEE/RSJ 2010 International Conference on Intelligent Robots and Systems, IROS 2010 - Conference Proceedings*, pages 261–266. IEEE.

[81] Hart, S. G. (2006). Nasa-task load index (nasa-tlx); 20 years later. In *Proceedings of the human factors and ergonomics society annual meeting*, volume 50, pages 904–908. Sage Publications.

[82] Hauser, K. (2013). Recognition, prediction, and planning for assisted teleoperation of freeform tasks. *Autonomous Robots*, 35(4):241–254.

[83] Hertkorn, K., Weber, B., Kremer, P., Roa, M. A., and Borst, C. (2013). Assistance for telepresence using online grasp planning. In *Humanoid Robots (Humanoids), 2013 13th IEEE-RAS International Conference on*, pages 507–513. IEEE.

[84] Hirzinger, G., Heindl, J., Landzettel, K., and Brunner, B. (1992). Multisensory shared autonomy-a key issue in the space robot technology experiment ROTEX. In *Intelligent Robots and Systems, 1992., Proceedings of the 1992 lEEE/RSJ International Conference on*, volume 1, pages 221–230. IEEE.

[85] Hokayem, P. F. and Spong, M. W. (2006). Bilateral teleoperation: An historical survey. *Automatica*, 42(12):2035–2057.

[86] Hughes, M. C. and Sudderth, E. (2013). Memoized Online Variational Inference for Dirichlet Process Mixture Models. In *Advances in Neural Information Processing Systems 26*, volume 2, pages 1133–1141.

[87] Hüntemann, A., Demeester, E., Nuttin, M., and Van Brussel, H. (2008). Online user modeling with Gaussian Processes for Bayesian plan recognition during power-wheelchair steering. In *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*, pages 285–292. IEEE.

[88] Huntemann, A., Demeester, E., Poorten, E. V., Van Brussel, H., and De Schutter, J. (2013). Probabilistic approach to recognize local navigation plans by fusing past driving information with a personalized user model. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 4376–4383. IEEE.

[89] Hwang, Y. K. and Ahuja, N. (1992). A potential field approach to path planning. *Robotics and Automation, IEEE Transactions on*, 8(1):23–32.

[90] Iba, S., Paredis, C. J. J., and Khosla, P. K. (2005). Interactive multimodal robot programming. *The international journal of robotics research*, 24(1):83–104.

[91] Iturrate, I., Antelis, J. M., Kübler, A., and Minguez, J. (2009). A noninvasive brain-actuated wheelchair based on a P300 neurophysiological protocol and automated navigation. *Robotics, IEEE Transactions on*, 25(3):614–627.

[92] Jacoff, A., Messina, E., Weiss, B. A., Tadokoro, S., and Nakagawa, Y. (2003). Test arenas and performance metrics for urban search and rescue robots. In *Intelligent Robots and Systems, 2003.(IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, volume 4, pages 3396–3403. IEEE.

[93] Jain, A., Koppula, H. S., Raghavan, B., Soh, S., and Saxena, A. (2015). Car that knows before you do: Anticipating maneuvers via learning temporal driving models. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3182–3190.

[94] Javdani, S., Srinivasa, S., and Bagnell, J. A. D. (2015). Shared Autonomy via Hindsight Optimization. In *Proceedings of Robotics: Science and Systems*, Rome, Italy.

[95] Kapoor, A., Grauman, K., Urtasun, R., and Darrell, T. (2010). Gaussian processes for object categorization. *International journal of computer vision*, 88(2):169–188.

[96] Kaupp, T., Douillard, B., Ramos, F., Makarenko, A., and Upcroft, B. (2007). Shared environment representation for a human-robot team performing information fusion. *Journal of Field Robotics*, 24(11-12):911–942.

[97] Kaupp, T. and Makarenko, A. (2008). Measuring human-robot team effectiveness to determine an appropriate autonomy level. In *Proceedings - IEEE International Conference on Robotics and Automation*, pages 2146–2151. IEEE.

[98] Kaupp, T., Makarenko, A., and Durrant-Whyte, H. (2010). Human–robot communication for collaborative decision making—A probabilistic approach. *Robotics and Autonomous Systems*, 58(5):444–456.

[99] Kelley, R., Tavakkoli, A., King, C., Nicolescu, M., Nicolescu, M., and Bebis, G. (2008). Understanding human intentions via hidden markov models in autonomous mobile robots. In *Proceedings of the 3rd ACM/IEEE international conference on Human robot interaction*, pages 367–374. ACM.

[100] Keskinpala, H. K., Adams, J. A., and Kawamura, K. (2003). PDA-based human-robotic interface. In *Systems, Man and Cybernetics, 2003. IEEE International Conference on*, volume 4, pages 3931–3936. IEEE.

[101] Kim, D.-J., Hazlett-Knudsen, R., Culver-Godfrey, H., Rucks, G., Cunningham, T., Portee, D., Bricout, J., Wang, Z., and Behal, A. (2012). How autonomy impacts performance and satisfaction: Results from a study with spinal cord injured subjects using an assistive robot. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 42(1):2–14.

[102] Klanke, S. and Vijayakumar, S. (2008). Lwpr supplementary documentation. Technical report, Tech. Rep.

[103] Klanke, S., Vijayakumar, S., and Schaal, S. (2008). A library for locally weighted projection regression. *Journal of Machine Learning Research*, 9(Apr):623–626.

[104] Kofman, J., Wu, X., Luu, T. J., and Verma, S. (2005). Teleoperation of a robot manipulator using a vision-based human-robot interface. *Industrial Electronics, IEEE Transactions on*, 52(5):1206–1219.

[105] Kohlbrecher, S., Meyer, J., von Stryk, O., and Klingauf, U. (2011). A Flexible and Scalable SLAM System with Full 3D Motion Estimation. In *Proc. IEEE International Symposium on Safety, Security and Rescue Robotics (SSRR)*. IEEE.

[106] Krishnan, S., Garg, A., Patil, S., Lea, C., Hager, G., Abbeel, P., and Goldberg, K. (2015). Transition State Clustering: Unsupervised Surgical Trajectory Segmentation For Robot Learning. In *International Symposium on Robotics Research (ISRR)*, pages 1–16.

[107] Lacey, G. and MacNamara, S. (2000). Context-aware shared control of a robot mobility aid for the elderly blind. *The International Journal of Robotics Research*, 19(11):1054–1065.

[108] Latif, H. O., Sherkat, N., and Lotfi, A. (2009). Teleoperation through eye gaze (TeleGaze): a multimodal approach. In *Robotics and Biomimetics (ROBIO), 2009 IEEE International Conference on*, pages 711–716. IEEE.

[109] LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *Nature*, 521(7553):436–444.

[110] Lee, D., Martinez-Palafox, O., and Spong, M. W. (2006). Bilateral teleoperation of a wheeled mobile robot over delayed communication network. In *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, pages 3298–3303. IEEE.

[111] Levine, S. P., Bell, D. A., Jaros, L. A., Simpson, R. C., Koren, Y., and Borenstein, J. (1999). The NavChair assistive wheelchair navigation system. *Rehabilitation Engineering, IEEE Transactions on*, 7(4):443–451.

[112] Li, M. and Okamura, A. M. (2003). Recognition of operator motions for real-time assistance using virtual fixtures. In *Haptic Interfaces for Virtual Environment and Teleoperator Systems, 2003. HAPTICS 2003. Proceedings. 11th Symposium on*, pages 125–131. IEEE.

[113] Li, Q., Chen, W., and Wang, J. (2011). Dynamic shared control for human-wheelchair cooperation. In *Proceedings - IEEE International Conference on Robotics and Automation*, pages 4278–4283. IEEE.

[114] Li, Y., Tee, K. P., Chan, W. L., Yan, R., Chua, Y., and Limbu, D. K. (2015). Role adaptation of human and robot in collaborative tasks. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pages 5602–5607. IEEE.

[115] Lin, C.-W., Khong, M.-H., and Liu, Y.-C. (2015). Experiments on human-in-the-loop coordination for multirobot system with task abstraction. *IEEE Transactions on Automation Science and Engineering*, 12(3):981–989.

[116] Lin, M.-H. and Song, K.-T. (2014). Design and experimental study of a shared-controlled omnidirectional mobile platform. In *Systems, Man and Cybernetics (SMC), 2014 IEEE International Conference on*, pages 3579–3584. IEEE.

[117] Liu, Z. and Von Wichert, G. (2014). Extracting semantic indoor maps from occupancy grids. *Robotics and Autonomous Systems*, 62(5):663–674.

[118] Lopes, A. C., Pires, G., Vaz, L., and Nunes, U. (2011). Wheelchair navigation assisted by Human-Machine shared-control and a P300-based Brain Computer Interface. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 2438–2444. IEEE.

[119] Lozano-Perez, T., Cox, I. J., and Wilfong, G. T. (2012). *Autonomous robot vehicles*. Springer Science & Business Media.

[120] Mandel, C., Lüth, T., Laue, T., Röfer, T., Gräser, A., and Krieg-Brückner, B. (2009). Navigating a smart wheelchair with a brain-computer interface interpreting steady-state visual evoked potentials. In *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pages 1118–1125. IEEE.

[121] Marayong, P., Li, M., Okamura, A. M., and Hager, G. D. (2003). Spatial motion constraints: Theory and demonstrations for robot guidance using virtual fixtures. In *Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on*, volume 2, pages 1954–1959. IEEE.

[122] Marion, P., Fallon, M., Deits, R., Valenzuela, A., Pérez D'Arpino, C., Izatt, G., Manuelli, L., Antone, M., Dai, H., Koolen, T., et al. (2017). Director: A user interface designed for robot operation with shared autonomy. *Journal of Field Robotics*, 34(2):262–280.

[123] Masone, C., Franchi, A., Bülthoff, H. H., and Giordano, P. R. (2012). Interactive planning of persistent trajectories for human-assisted navigation of mobile robots. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 2641–2648. IEEE.

[124] Masone, C., Giordano, P. R., Bulthoff, H. H., and Franchi, A. (2014). Semi-autonomous trajectory generation for mobile robots with integral haptic shared control. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 6468–6475. IEEE.

[125] Meyer, J., Sendobry, A., Kohlbrecher, S., Klingauf, U., and Von Stryk, O. (2012). Comprehensive simulation of quadrotor uavs using ros and gazebo. In *Simulation, Modeling, and Programming for Autonomous Robots*, pages 400–411. Springer.

[126] Michaud, F., Boissy, P., Labonte, D., Corriveau, H., Grant, A., Lauria, M., Cloutier, R., Roux, M.-A., Iannuzzi, D., and Royer, M.-P. (2007). Telepresence Robot for Home Care Assistance. In *AAAI Spring Symposium: Multidisciplinary Collaboration for Socially Assistive Robotics*, pages 50–55. California, USA.

[127] Morales Saiki, L. Y., Satake, S., Huq, R., Glas, D., Kanda, T., and Hagita, N. (2012). How do people walk side-by-side?: using a computational model of human behavior for a social robot. In *Proceedings of the seventh annual ACM/IEEE international conference on Human-Robot Interaction*, pages 301–308. ACM.

[128] Muja, M. and Lowe, D. G. (2009). Fast Approximate Nearest Neighbors with Automatic Algorithm Configuration. In *International Conference on Computer Vision Theory and Applications (VISAPP '09)*, pages 1–10. INSTICC Press.

[129] Murali, A., Garg, A., Krishnan, S., Pokorny, F. T., Abbeel, P., Darrell, T., and Goldberg, K. (2016). TSC-DL : Unsupervised Trajectory Segmentation of Multi-Modal Surgical Demonstrations with Deep Learning. In *Robotics and Automation (ICRA), 2016 IEEE International Conference on*. IEEE.

[130] Murphy, R. R., Dreger, K. L., Newsome, S., Rodocker, J., Steimle, E., Kimura, T., Makabe, K., Matsuno, F., Tadokoro, S., and Kon, K. (2011). Use of remotely operated marine vehicles at Minamisanriku and Rikuzentakata Japan for disaster recovery. In *Safety, Security, and Rescue Robotics (SSRR), 2011 IEEE International Symposium on*, pages 19–25. IEEE.

[131] Murphy, R. R., Kravitz, J., Stover, S. L., and Shoureshi, R. (2009). Mobile robots in mine rescue and recovery. *Robotics & Automation Magazine, IEEE*, 16(2):91–103.

[132] Narayanan, K. K., Posada, L. F., Hoffmann, F., and Bertram, T. (2011). Scenario and context specific visual robot behavior learning. In *Proceedings - IEEE International Conference on Robotics and Automation*, pages 1180–1185. IEEE.

[133] Nicolescu, M., Chadwicke Jenkins, O., Olenderski, A., and Fritzinger, E. (2008). Learning behavior fusion from demonstration. *Interaction Studies*, 9(2):319–352.

[134] Nicolescu, M., Jenkins, O. C., and Stanhope, A. (2007). Fusing robot behaviors for human-level tasks. In *2007 IEEE 6th International Conference on Development and Learning, ICDL*, pages 76–81. IEEE.

[135] Niekum, S. and Chitta, S. (2013). Incremental Semantically Grounded Learning from Demonstration. In *Robotics: Science and Systems IX*.

[136] Niekum, S., Osentoski, S., Konidaris, G., and Barto, A. G. (2012). Learning and generalization of complex tasks from unstructured demonstrations. In *IEEE International Conference on Intelligent Robots and Systems*, pages 5239–5246. IEEE.

[137] Nielsen, C. W., Goodrich, M. A., and Ricks, R. W. (2007). Ecological interfaces for improving mobile robot teleoperation. *Robotics, IEEE Transactions on*, 23(5):927–941.

[138] Niemeyer, G., Preusche, C., Stramigioli, S., and Lee, D. (2016). Telerobotics. In *Springer handbook of robotics*, pages 1085–1108. Springer.

[139] Okada, Y., Nagatani, K., Yoshida, K., Tadokoro, S., Yoshida, T., and Koyanagi, E. (2011). Shared autonomy system for tracked vehicles on rough terrain based on continuous three-dimensional terrain scanning. *Journal of Field Robotics*, 28(6):875–893.

[140] Okada, Y., Nagatani, K., Yoshida, K., Yoshida, T., and Koyanagi, E. (2010). Shared autonomy system for tracked vehicles to traverse rough terrain based on continuous three-dimensional terrain scanning. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 357–362. IEEE.

[141] O'Malley, M. K., Gupta, A., Gen, M., and Li, Y. (2006). Shared control in haptic systems for performance enhancement and training. *Journal of Dynamic Systems, Measurement, and Control*, 128(1):75–85.

[142] Papageorgiou, X. S., Chalvatzaki, G., Tzafestas, C. S., and Maragos, P. (2014). Hidden markov modeling of human normal gait using laser range finder for a mobility assistance robot. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 482–487. IEEE.

[143] Parasuraman, R., Sheridan, T. B., and Wickens, C. D. (2000). A model for types and levels of human interaction with automation. *IEEE transactions on systems, man, and cybernetics. Part A, Systems and humans : a publication of the IEEE Systems, Man, and Cybernetics Society*, 30(3):286–97.

[144] Parikh, S. P., Grassi, V., Kumar, V., and Okamoto, J. J. (2005). Usability study of a control framework for an intelligent wheelchair. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pages 4745–4750. IEEE.

[145] Parikh, S. P., Grassi Jr, V., Kumar, V., and Okamoto Jr, J. (2004). Incorporating user inputs in motion planning for a smart wheelchair. In *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, volume 2, pages 2043–2048. IEEE.

[146] Paul, R., Triebel, R., Rus, D., and Newman, P. (2012). Semantic categorization of outdoor scenes with uncertainty estimates using multi-class gaussian process classification. In *IEEE International Conference on Intelligent Robots and Systems*, pages 2404–2410. IEEE.

[147] Peinado, G., Urdiales, C., Peula, J. M., Fdez-Carmona, M., Annicchiarico, R., Sandoval, F., and Caltagirone, C. (2011). Navigation skills based profiling for collaborative wheelchair control. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 2229–2234. IEEE.

[148] Perrin, X., Chavarriaga, R., Colas, F., Siegwart, R., and Millán, J. d. R. (2010). Brain-coupled interaction for semi-autonomous navigation of an assistive robot. *Robotics and Autonomous Systems*, 58(12):1246–1255.

[149] Perzanowski, D., Schultz, A. C., Adams, W., Marsh, E., and Bugajska, M. (2001). Building a multimodal human-robot interface. *Intelligent Systems, IEEE*, 16(1):16–21.

[150] Piao, J. and McDonald, M. (2008). Advanced driver assistance systems from autonomous to cooperative approach. *Transport Reviews*, 28(5):659–684.

[151] Pitzer, B., Styer, M., Bersch, C., DuHadway, C., and Becker, J. (2011). Towards perceptual shared autonomy for robotic mobile manipulation. In *Proceedings - IEEE International Conference on Robotics and Automation*, pages 6245–6251. IEEE.

[152] Podnar, G., Dolan, J., Elfes, A., and Bergerman, M. (2008). Multi-level autonomy robot telesupervision.

[153] Poncela, A., Urdiales, C., Pérez, E. J., and Sandoval, F. (2009). A new efficiency-weighted strategy for continuous human/robot cooperation in navigation. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 39(3):486–500.

[154] Powers, D. (2011). Evaluation: From precision, recall and f-measure to roc., informedness, markedness & correlation. *Journal of Machine Learning Technologies*, 2(1):37 – 63.

[155] Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., Wheeler, R., and Ng, A. Y. (2009). ROS: an open-source Robot Operating System. In *ICRA workshop on open source software*, volume 3, page 5.

[156] Quinlan, S. and Khatib, O. (1993). Elastic bands: Connecting path planning and control. In *Robotics and Automation, 1993. Proceedings., 1993 IEEE International Conference on*, pages 802–807. IEEE.

[157] Rasmussen, C. and Williams, C. (2006). Gaussian processes for machine learning.

[158] Rebsamen, B., Guan, C., Zhang, H., Wang, C., Teo, C., Ang Jr, M. H., and Burdet, E. (2010). A brain controlled wheelchair to navigate in familiar environments. *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, 18(6):590–598.

[159] Rohmer, E., Pinheiro, P., Raizer, K., Olivi, L., and Cardozo, E. (2015). A novel platform supporting multiple control strategies for assistive robots. In *Robot and Human Interactive Communication (RO-MAN), 2015 24th IEEE International Symposium on*, pages 763–769. IEEE.

[160] Rösmann, C., Hoffmann, F., and Bertram, T. (2017). Integrated online trajectory planning and optimization in distinctive topologies. *Robotics and Autonomous Systems*, 88:142–153.

[161] Rusu, R. B. and Cousins, S. (2011). 3D is here: point cloud library. In *IEEE International Conference on Robotics and Automation*, pages 1 – 4. IEEE.

[162] Sa, I. and Corke, P. (2012). Verticle infrastructure inspection using a quadcopter and shared autonomy control. In *Proceedings of the 8th International Conference on Field and Service Robotics*. Springer.

[163] Sa, I., Hrabar, S., and Corke, P. (2015). Inspection of pole-like structures using a visual-inertial aided vtol platform with shared autonomy. *Sensors*, 15(9):22003–22048.

[164] Saleh, L., Chevrel, P., Claveau, F., Lafay, J.-F., and Mars, F. (2013). Shared steering control between a driver and an automation: Stability in the presence of driver behavior uncertainty. *Intelligent Transportation Systems, IEEE Transactions on*, 14(2):974–983.

[165] Sankaran, B., Pitzer, B., and Osentoski, S. (2012). Failure recovery with shared autonomy. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 349–355. IEEE.

[166] Schwarz, G. (1978). Estimating the dimension of a model. *The annals of statistics*, 6(2):461–464.

[167] Secchi, C., Franchi, A., Bulthoff, H. H., and Robuffo Giordano, P. (2013). Bilateral control of the degree of connectivity in multiple mobile-robot teleoperation. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 3645–3652. IEEE.

[168] Sehestedt, S., Kodagoda, S., and Dissanayake, G. (2010). Models of motion patterns for mobile robotic systems. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 4127–4132. IEEE.

[169] Shamaei, K., Che, Y., Murali, A., Sen, S., Patil, S., Goldberg, K., and Okamura, A. M. (2015). A paced shared-control teleoperated architecture for supervised automation of multilateral surgical tasks. In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pages 1434–1439. IEEE.

[170] Sheridan, T. B. (1992). *Telerobotics, automation and human supervisory control*. The MIT press.

[171] Shim, H.-M., Lee, E.-H., Shim, J.-H., Lee, S.-M., and Hong, S.-H. (2005). Implementation of an intelligent walking assistant robot for the elderly in outdoor environment. In *Rehabilitation Robotics, 2005. ICORR 2005. 9th International Conference on*, pages 452–455. IEEE.

[172] Simpson, R. C. and Levine, S. P. (2002). Voice control of a powered wheelchair. *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, 10(2):122–125.

[173] Soh, H. and Demiris, Y. (2013). When and how to help: An iterative probabilistic model for learning assistance by demonstration. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pages 3230–3236. IEEE.

[174] Soh, H. and Demiris, Y. (2015). Learning assistance by demonstration: Smart mobility with shared control and paired haptic controllers. *Journal of Human-Robot Interaction*, 4(3):76–100.

[175] Stefanov, N., Passenberg, C., Peer, A., and Buss, M. (2013). Design and evaluation of a haptic computer-assistant for telemanipulation tasks. *Human-Machine Systems, IEEE Transactions on*, 43(4):385–397.

[176] Storms, J., Vozar, S., and Tilbury, D. (2014). Predicting human performance during teleoperation. In *Proceedings of the 2014 ACM/IEEE international conference on Human-robot interaction*, pages 298–299. ACM.

[177] Storms, J. G. and Tilbury, D. M. (2014). Blending of human and obstacle avoidance control for a high speed mobile robot. In *American Control Conference (ACC), 2014*, pages 3488–3493. IEEE.

[178] Sung, H. (2004). *Gaussian Mixture Regression and Classification*. PhD thesis, RICE UNIVERSITY.

[179] Taha, T., Miró, J. V., and Dissanayake, G. (2008). POMDP-based long-term user intention prediction for wheelchair navigation. *IEEE International Conference on Robotics and Automation*, pages 3920–3925.

[180] Tai, L. and Liu, M. (2016). Deep-learning in mobile robotics - from perception to control systems: A survey on why and why not. *arXiv preprint arXiv:1612.07139*.

[181] Tambe, M., Scerri, P., and Pynadath, D. V. (2002). Adjustable autonomy for the real world. *Journal of Artificial Intelligence Research*, 17(1):171–228.

[182] Teleoperation, N. and Peters, A. (2005). Extraction of Salient Features for Mobile Robot. In *American Control Conference, 2005. Proceedings of the 2005*, pages 4903–4908. IEEE.

[183] Thrun, S., Burgard, W., and Fox, D. (2005). *Probabilistic robotics*. MIT press.

[184] Thrun, S. and Mitchell, T. M. (1995). *Lifelong robot learning*. Springer.

[185] Tonin, L., Leeb, R., Tavella, M., Perdikis, S., and Millán, J. d. R. (2010). The role of shared-control in BCI-based telepresence. In *Systems Man and Cybernetics (SMC), 2010 IEEE International Conference on*, pages 1462–1466. IEEE.

[186] Toohey, K. and Duckham, M. (2015). Trajectory similarity measures. *SIGSPATIAL Special*, 7(1):43–50.

[187] Triebel, R., Grimmett, H., Paul, R., and Posner, I. (2016). Driven learning for driving: How introspection improves semantic mapping. In *Robotics Research*, pages 449–465. Springer.

[188] Triebel, R., Stühmer, J., Souiai, M., and Cremers, D. (2014). Active online learning for interactive segmentation using sparse gaussian processes. In *German Conference on Pattern Recognition*, pages 641–652. Springer.

[189] Trieu, H. T., Nguyen, H. T., and Willey, K. (2008). Shared control strategies for obstacle avoidance tasks in an intelligent wheelchair. In *Engineering in Medicine and Biology Society, 2008. EMBS 2008. 30th Annual International Conference of the IEEE*, pages 4254–4257. IEEE.

[190] Trieu, H. T., Willey, K., and Nguyen, H. T. (2009). Adaptive shared control strategies based in the Bayesian recursive technique in an intelligent wheelchair. In *Engineering in Medicine and Biology Society, 2009. EMBC 2009. Annual International Conference of the IEEE*, pages 7118–7121. IEEE.

[191] Urdiales, C., Fernández-Carmona, M., Peula, J. M., Annicchiaricco, R., Sandoval, F., and Caltagirone, C. (2010). Efficiency based modulation for wheelchair driving collaborative control. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 199–204. IEEE.

[192] Urdiales, C., Peula, J. M., Fdez-Carmona, M., Barrué, C., Pérez, E. J., Sánchez-Tato, I., Del Toro, J. C., Galluppi, F., Cortés, U., Annichiaricco, R., and Others (2011). A new multi-criteria optimization strategy for shared control in wheelchair assisted navigation. *Autonomous Robots*, 30(2):179–197.

[193] Urdiales, C., Peula, J. M., Fernandez-Carmona, M., Annicchiaricco, R., Sandoval, F., and Caltagirone, C. (2009). Adaptive collaborative assistance for wheelchair driving via CBR learning. In *2009 IEEE International Conference on Rehabilitation Robotics, ICORR 2009*, pages 731–736. IEEE.

[194] Urdiales, C., Poncela, A., Sanchez-Tato, I., Galluppi, F., Olivetti, M., and Sandoval, F. (2007). Efficiency based reactive shared control for collaborative human/robot navigation. In *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, pages 3586–3591. IEEE.

[195] Vijayakumar, S., D'souza, A., and Schaal, S. (2005). Incremental online learning in high dimensions. *Neural computation*, 17(12):2602–2634.

[196] Vozar, S. E. (2013). *A Framework for Improving the Speed and Performance of Teleoperated Mobile Manipulators*. PhD thesis, The University of Michigan.

[197] Wang, H. and Liu, X. P. (2014). Adaptive shared control for a novel mobile assistive robot. *Mechatronics, IEEE/ASME Transactions on*, 19(6):1725–1736.

[198] Wang, X., Yang, C., Ma, H., and Cheng, L. (2015). Shared control for teleoperation enhanced by autonomous obstacle avoidance of robot manipulator. In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pages 4575–4580. IEEE.

[199] Wang, Z. (2013). Intention inference and decision making with hierarchical gaussian process dynamics models.

[200] Wasson, G., Sheth, P., Alwan, M., Granata, K., Ledoux, A., and Huang, C. (2003). User intent in a shared control framework for pedestrian mobility aids. In *Intelligent Robots and Systems, 2003.(IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, volume 3, pages 2962–2967. IEEE.

[201] Witzig, T., Zollner, J. M., Pangercic, D., Osentoski, S., Jakel, R., and Dillmann, R. (2013). Context aware shared autonomy for robotic manipulation tasks. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pages 5686–5693. IEEE.

[202] Xu, A. and Dudek, G. (2012). Trust-driven interactive visual navigation for autonomous robots. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 3922–3929. IEEE.

[203] Xu, A., Kalmbach, A., and Dudek, G. (2014). Adaptive Parameter EXploration (APEX): Adaptation of robot autonomy from human participation. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 3315–3322. IEEE.

[204] Yu, H., Spenko, M., and Dubowsky, S. (2003). An adaptive shared control system for an intelligent mobility aid for the elderly. *Autonomous Robots*, 15(1):53–66.

[205] Yu, N., Wang, K., Li, Y., Xu, C., and Liu, J. (2015). A haptic shared control algorithm for flexible human assistance to semi-autonomous robots. In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pages 5241–5246. IEEE.

[206] Yu, W. and Alqasemi, R. (2005). Telemanipulation assistance based on motion intention recognition. *IEEE International Conference on Robotics and Automation*, pages 1121–1126.

[207] Zhang, J., Wang, J., and Chen, W. (2014). A control system of driver assistance and human following for smart wheelchair. In *Robotics and Biomimetics (ROBIO), 2014 IEEE International Conference on*, pages 1927–1932. IEEE.