BEHAVIOUR-DRIVEN MOTION SYNTHESIS

Èric Pairet Artau

A thesis submitted for the degree of

Doctor of Philosophy

February, 2022

Abstract

Heightened demand for alternatives to human exposure to strenuous and repetitive labour, as well as to hazardous environments, has led to an increased interest in real-world deployment of robotic agents. Targeted applications require robots to be adept at synthesising complex motions rapidly across a wide range of tasks and environments. To this end, this thesis proposes leveraging abstractions of the problem at hand to ease and speed up the solving. We formalise abstractions to hint relevant robotic behaviour to a family of planning problems, and integrate them tightly into the motion synthesis process to make real-world deployment in complex environments practical. We investigate three principal challenges of this proposition.

Firstly, we argue that behavioural samples in form of trajectories are of particular interest to guide robotic motion synthesis. We formalise a framework with behavioural semantic annotation that enables the storage and bootstrap of sets of problem-relevant trajectories.

Secondly, in the core of this thesis, we study strategies to exploit behavioural samples in task instantiations that differ significantly from those stored in the framework. We present two novel strategies to efficiently leverage offline-computed problem behavioural samples: (i) online modulation based on geometry-tuned potential fields, and (ii) experience-guided exploration based on trajectory segmentation and malleability.

Thirdly, we demonstrate that behavioural hints can be extracted on-the-fly to tackle highly-constrained, ever-changing complex problems, from which there is no prior knowledge. We propose a multi-layer planner that first solves a simplified version of the problem at hand, to then inform the search for a solution in the constrained space.

Our contributions on efficient motion synthesis via behaviour guidance augment the robots' capabilities to deal with more complex planning problems, and do so more effectively than related approaches in the literature by computing better quality paths in lower response time. We demonstrate our contributions, in both laboratory experiments and field trials, on a spectrum of planning problems and robotic platforms ranging from high-dimensional humanoids and robotic arms with a focus on autonomous manipulation in resembling environments, to high-dimensional kinematic motion planning with a focus on autonomous safe

navigation in unknown environments. While this thesis was motivated by challenges on motion synthesis, we have explored the applicability of our findings on disparate robotic fields, such as grasp and task planning. We have made some of our contributions open-source hoping they will be of use to the robotics community at large.

List of Publications

Publications in the Compendium

The presented thesis is a compendium of the following research articles:

- **Èric Pairet**, Juan David Hernández, Marc Carreras, Yvan Petillot, and Morteza Lahijanian. "Online mapping and motion planning under uncertainty for safe navigation in unknown environments". In: *IEEE Transactions on Automation Science and Engineering* (2021), pp. 1–23

- **Èric Pairet**, Constantinos Chamzas, Yvan Petillot, and Lydia Kavraki. "Path Planning for Manipulation using Experience-driven Random Trees". In: *IEEE Robotics and Automation Letters* 6.2 (2021), pp. 3295–3302

- Paola Ardón, **Èric Pairet**, Yvan Petillot, Subramanian Ramamoorthy, Ronald PA Petrick, and Katrin S Lohan. "Self-Assessment of Grasp Affordance Transfer". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*. First two authors contributed equally to this work. IEEE. 2020, pp. 9385–9392

- **Èric Pairet**, Paola Ardón, Michael Mistry, and Yvan Petillot. "Learning Generalizable Coupling Terms for Obstacle Avoidance via Low-Dimensional Geometric Descriptors". In: *IEEE Robotics and Automation Letters* 4.4 (2019), pp. 3979–3986

- **Èric Pairet**, Paola Ardón, Michael Mistry, and Yvan Petillot. "Learning and Composing Primitive Skills for Dual-arm Manipulation". In: *Annual Conference Towards Autonomous Robotic Systems*. Advanced Robotics at Queen Mary (ARQ) best paper award. Springer. 2019, pp. 65–77

Other Journal Articles and Peer-reviewed Conference Papers

The work developed in this thesis also led to the following publications:

- Jonatan Scharff Willners, Ignacio Carlucho, Sean Katagiri, Chandler Lemoine, Joshua Roe, Dylan Stephens, Tomasz Łuczyński, Shida Xu, Yaniel Carreno, **Èric Pairet**, Sen Wang,

Corina Barbalata, and Yvan Petillot. "From market-ready ROVs to low-cost AUVs". In: *OCEANS 2021*. IEEE. 2021

- Paola Ardón, Maria Eugenia Cabrera, **Èric Pairet**, Ronald PA Petrick, Subramanian Ramamoorthy, Katrin S Lohan, and Maya Cakmak. *Affordance-Aware Handovers with Human Arm Mobility Constraints*. Best presentation award. 2021. Workshop on Learning for Caregiving Robots, IEEE International Conference on Robotics and Automation

- Jonatan Scharff Willners, Yaniel Carreno, Shida Xu, Tomasz Łuczyński, Sean Katagiri, Joshua Roe, **Èric Pairet**, Yvan Petillot, and Sen Wang. "Robust Underwater SLAM using Autonomous Relocalisation". In: *13th IFAC Conference on Control Applications in Marine Systems, Robotics, and Vehicles*. 2021

- Yaniel Carreno, Pierre Le Bras, **Èric Pairet**, Paola Ardón, Mike Chantler, and Ronald PA Petrick. "An Integrated Framework for Remote Planning". In: *Proceedings of the International Conference on Automated Planning and Scheduling - Integrated Planning, Acting, and Execution Workshop*. 2021

- Paola Ardón, **Èric Pairet**, Katrin S Lohan, Subramanian Ramamoorthy, and Ronald Petrick. "Building Affordance Relations for Robotic Agents - A Survey". In: *International Joint Conference on Artificial Intelligence* (2021)

- Paola Ardón, Maria Eugenia Cabrera, **Èric Pairet**, Ronald PA Petrick, Subramanian Ramamoorthy, Katrin S Lohan, and Maya Cakmak. "Affordance-Aware Handovers with Human Arm Mobility Constraints". In: *IEEE Robotics and Automation Letters* 6.2 (2021), pp. 3136–3143

- Jonatan Scharff Willners, Daniel Gonzalez-Adell, Juan David Hernández, **Èric Pairet**, and Yvan Petillot. "Online 3-Dimensional Path Planning with Kinematic Constraints in Unknown Environments Using Hybrid A* with Tree Pruning". In: *Sensors* 21.4 (2021)

- Yaniel Carreno, **Èric Pairet**, Yvan Petillot, and Ronald PA Petrick. "A Decentralised Strategy for Heterogeneous AUV Missions via Goal Distribution and Temporal Planning". In: *Proceedings of the International Conference on Automated Planning and Scheduling*. Vol. 30. 2020, pp. 431–439

- Yaniel Carreno, **Èric Pairet**, Yvan Petillot, and Ronald PA Petrick. "Decentralised Task Allocation and Planning for Heterogeneous AUVs". In: *Proceedings of the International Conference on Automated Planning and Scheduling System Demonstration*. 2020

- Yaniel Carreno, **Èric Pairet**, Paola Ardón, Yvan Petillot, and Ronald PA Petrick. "Task Allocation and Planning for Offshore Mission Automation". In: *Proceedings of the International Conference on Automated Planning and Scheduling System Demonstration*. 2020

- Katrin Lohan, Muneeb Imtiaz Ahmad, Christian Dondrup, Paola Ardón, **Èric Pairet**, and Alessandro Vinciarelli. "Adapting Movements and Behaviour to Favour Communication in Human-Robot Interaction". In: *Modelling Human Motion.* Springer, 2020, pp. 271–297

- Yaniel Carreno, **Èric Pairet**, Yvan Petillot, and Ronald PA Petrick. "Task Allocation Strategy for Heterogeneous Robot Teams in Offshore Missions". In: *Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems.* 2020, pp. 222–230

- David Robb, Muneeb Ahmad, Carlo Tiseo, Simona Aracri, Alistair C McConnell, Vincent Page, Christian Dondrup, Francisco Garcia, Hai Nguyen, **Èric Pairet**, Paola Ardón, Tushar Semwal, Hazel Taylor, Lindsay Wilson, David Lane, Helen Hastie, and Katrin S Lohan. "Robots in the Danger Zone: Exploring Public Perception through Engagement". In: *15th ACM/IEEE International Conference on Human-Robot Interaction.* IEEE. 2020, pp. 93–102

- Paola Ardón, **Èric Pairet**, Ronald PA Petrick, Subramanian Ramamoorthy, and Katrin S Lohan. "Learning Grasp Affordance Reasoning through Semantic Relations". In: *IEEE Robotics and Automation Letters* 4.4 (2019), pp. 4571–4578

- Paola Ardón, **Èric Pairet**, Ron Petrick, Subramanian Ramamoorthy, and Katrin Solveig Lohan. "Reasoning on Grasp-Action Affordances". In: *Annual Conference Towards Autonomous Robotic Systems.* Best paper award finalist. Springer. 2019, pp. 3–15

- **Èric Pairet**, Paola Ardón, Xingkun Liu, José Lopes, Helen Hastie, and Katrin S Lohan. "A Digital Twin for Human-Robot Interaction". In: *14th ACM/IEEE International Conference on Human-Robot Interaction.* First two authors contributed equally to this work. IEEE. 2019, pp. 372–372

- **Èric Pairet**, Paola Ardón, Frank Broz, Michael Mistry, and Yvan Petillot. "Learning and Generalisation of Primitives Skills Towards Robust Dual-arm Manipulation". In: *Proceedings of the AAAI Fall Symposium on Reasoning and Learning in Real-World Systems for Long-Term Autonomy.* AAAI Press. 2018, pp. 62–69

- Paola Ardón, **Èric Pairet**, Subramanian Ramamoorthy, and Katrin Solveig Lohan. "Towards Robust Grasps: Using the Environment Semantics for Robotic Object Affordances". In: *Proceedings of the AAAI Fall Symposium on Reasoning and Learning in Real-World Systems for Long-Term Autonomy.* AAAI Press. 2018, pp. 5–12

- **Èric Pairet**, Juan David Hernández, Morteza Lahijanian, and Marc Carreras. "Uncertainty-based Online Mapping and Motion Planning for Marine Robotics

Guidance". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems.* IEEE. 2018, pp. 2367–2374

Acknowledgements

The work presented throughout this manuscript becomes a reality with the unconditional support of many incredible people. Their continuous encouragement and discussions have been essential to complete this work. I would like to give my gratitude to all of them.

Foremost, I would like to express my deepest gratitude to my family in this way: *moltes gràcies, a tots, per inculcar-me els valors que m'han convertit amb qui sóc avui, i per poder comptar amb vosaltres sempre que ho he necessitat. Especialment, a tu, mare, que mai t'has rendit per treure el millor de mi, i per ensenyar-me a ser exigent i a perseverar envers la dificultat. Els meus èxits són també vostres.* Also, to my friends back home: *merci per ser tant collonuts i, que malgrat la distància, sempre hem trobat la oportunitat de compartir una birra on fos que parés.* On this thread, a special mention and thanks to Paola Ardón for all kindness, patience and support through all bits and pieces, ups and downs of the PhD, and keeping up with the endless talks about my research. I could not have asked for a better peer reviewer, friend, nor life partner; we have gone through this together, as we will for whatever comes next.

I would not have the privilege to write these words without the support of my supervisors, Prof. Yvan Petillot and Prof. Michael Mistry. Their advice has been there whenever it was needed, and they have always ensured that this work was coming to fruition. A special thanks to Prof. Yvan Petillot, to whom I am deeply indebted for giving me the opportunity to join the Edinburgh Centre for Robotics, for his constant guidance, commitment and faith in me. Also, I wish to extend my gratitude to Dr. Ron Petrick for all his insightful feedback, and to Dr. Katrin Lohan and Prof. Helen Hastie for enabling me to expand my research scope within the ORCA Hub project, as well as to Dr. Frank Broz for his advice in the early stages of this PhD. I would like to thank the administrative team, Derek Davis, Lorna Brown and Lindsay Wilson, for all assistance on bookings and unexpected journeys.

A significant part of my PhD has been devoted to collaborations. I am thankful for every single collaboration opportunity I have had; each has been a source of knowledge and inspiration to me. Among many others, I especially thank Dr. Juan David Hernández: *muchas gracias por tu mentoría, amistad y consejo siempre que lo he necesitado; has sido un excelente mentor para*

Research Thesis Submission

Please note this form should be bound into the submitted thesis.

Name*:	ÈRIC PAIRET ARTAU		
School:	ENGINEERING AND PHYSICAL SCIENCES (EPS)		
Version: *(i.e. First, Resubmission, Final)*	Final	Degree Sought:	PhD

Declaration

In accordance with the appropriate regulations I hereby submit my thesis and I declare that:

1. The thesis embodies the results of my own work and has been composed by myself
2. Where appropriate, I have made acknowledgement of the work of others
3. The thesis is the correct version for submission and is the same version as any electronic versions submitted*.
4. My thesis for the award referred to, deposited in the Heriot-Watt University Library, should be made available for loan or photocopying and be available via the Institutional Repository, subject to such conditions as the Librarian may require
5. I understand that as a student of the University I am required to abide by the Regulations of the University and to conform to its discipline.
6. I confirm that the thesis has been verified against plagiarism via an approved plagiarism detection application e.g. Turnitin.

ONLY for submissions including published works

Please note you are only required to complete the Inclusion of Published Works Form (page 2) if your thesis contains published works)

7. Where the thesis contains published outputs under Regulation 6 (9.1.2) or Regulation 43 (9) these are accompanied by a critical review which accurately describes my contribution to the research and, for multi-author outputs, a signed declaration indicating the contribution of each author (complete)
8. Inclusion of published outputs under Regulation 6 (9.1.2) or Regulation 43 (9) shall not constitute plagiarism.

*   *Please note that it is the responsibility of the candidate to ensure that the correct version of the thesis is submitted.*

Signature of Candidate*:	*(signature)*	Date:	02-02-2022

Submission

Submitted By *(name in capitals)*:	ÈRIC PAIRET ARTAU
Signature of Individual Submitting:	*(signature)*
Date Submitted:	02-02-2022

For Completion in the Student Service Centre (SSC)

Limited Access	Requested	Yes		No		Approved	Yes		No	
*E-thesis Submitted (**mandatory for final theses**)*										
Received in the SSC by *(name in capitals)*:						Date:				

Inclusion of Published Works

Declaration

This thesis contains one or more multi-author published works. In accordance with Regulation 6 (9.1.2) I hereby declare that the contributions of each author to these publications is as follows:

Citation details	È. Pairet, P. Ardón, M. Mistry, and Y. Petillot. "Learning and Composing Primitive Skills for Dual-arm Manipulation", in Conf. Towards Autonomous Robotic Systems. July 2019. Advanced Robotics at Queen Mary (ARQ) best paper award
È. Pairet	Designed, implemented and tested the proposed methodology
P. Ardón	Technical discussion and feedback
M. Mistry	Supervised work
Y. Petillot	Supervised work
Signature:	
Date:	02-02-2022

Citation details	P. Ardón, È. Pairet, R. Petrick, S. Ramamoorthy, and K. S. Lohan. "Learning Grasp Affordance Reasoning through Semantic Relations," in IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems and IEEE Robotics and Automation Letters. November 2019. First two authors contributed equally to this work.
P. Ardón	Designed, implemented and tested the methodology on grasp-affordances
È. Pairet	Designed and implemented the library of actions for grasp-affordances
Y. Petillot	Supervised work
R. Petrick	Supervised work
S. Ramamoorthy	Supervised work
K. Lohan	Supervised work
Signature:	
Date:	02-02-2022

Citation details	È. Pairet, P. Ardón, M. Mistry, and Y. Petillot. "Learning Generalisable Coupling Terms for Obstacle Avoidance via Low-dimensional Geometric Descriptors," in IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems and IEEE Robotics and Automation Letters. November 2019.
È. Pairet	Designed, implemented and tested the proposed methodology
P. Ardón	Technical discussion and feedback
M. Mistry	Supervised work
Y. Petillot	Supervised work
Signature:	
Date:	02-02-2022

Citation details	È. Pairet, C. Chamzas, Y. Petillot, and L. Kavraki. "Path Planning for Manipulation using Experience-driven Random Trees", in IEEE Robotics and Automation Letters. February 2021.
È. Pairet	Designed, implemented and tested the proposed methodology
C. Chamzas	Technical discussion and feedback, and provided the testing setup
Y. Petillot	Supervised work
L. Kavraki	Supervised work
Signature:	
Date:	02-02-2022

Citation details	È. Pairet, J. D. Hernández, M. Carreras, Y. Petillot, and M. Lahijanian. "Online Mapping and Motion Planning under Uncertainty for Safe Navigation in Unknown Environments", in IEEE Transactions on Automation Science and Engineering. 2021. In Press.
È. Pairet	Designed, implemented and tested the proposed methodology
J. D. Hernández	Technical discussion and feedback (while in Girona and Edinburgh)
M. Carreras	Supervised work (while in Girona)
Y. Petillot	Supervised work (while in Edinburgh)
M. Lahijanian	Supervised work, and technical discussion and feedback (while in Girona and Edinburgh)
Signature:	
Date:	02-02-2022

Contents

List of Figures

List of Tables

Acronyms

2D two-dimensional

3D three-dimensional

AD* anytime dynamic A*

AICO approximate inference control

AIT* adaptively informed trees

ARA* anytime repairing A*

AUV autonomous underwater vehicle

BIT* batch informed trees

CHOMP covariant Hamiltonian optimisation for motion planning

D* dynamic A*

DMP dynamic movement primitive

DoF degree of freedom

DRRT deformable RRT

ERT experience-driven random trees

ERTConnect bi-directional ERT

EST expansive-spaces tree

GMM Gaussian mixture model

GMR Gaussian mixture regression

GPR Gaussian process regression

GVD generalised Voronoi diagrams

HMM hidden Markov model

ILC iterative learning control

ITOMP incremental trajectory optimization for motion planning

KOMO k-order Markov optimisation

KPIECE kinodynamic planning by interior-exterior cell exploration

LbD learning by demonstration

LMS least mean squares

LWPR locally weighted projection regression

LWR locally weighted regression

NN neural network

OMPL open motion planning library

ORCA offshore robotics for certification of assets

PI^2 path integral policy improvement

PRM probabilistic roadmap

PRM* asymptotic optimal PRM

RABIT* regionally accelerated BIT

RC regressor chain

RFWR receptive field weighted regression

RieMO Riemannian motion optimisation

RL reinforcement learning

RPP randomized path planner

RRT rapidly-exploring random tree

RRT* asymptotic optimal RRT

RRTConnect bi-directional RRT

SLERP spherical linear interpolation

SST stable sparse RRT

STOMP stochastic trajectory optimisation motion planning

STRIDE search tree with resolution independent density estimation

SYCLOP synergistic combination of layers of planning

T-RRT transition-based RRT

T-RRT* asymptotic optimal T-RRT

TL transfer learning

TrajOpt trajectory optimisation

UAV unmanned aerial vehicle

Part I

PRELIMINARIES

Introduction

"Thinking drives behaviors. Behaviors drive action. Action drives results..."

— JAMIE FLINCHBAUGH

Autonomous robotic systems have been long envisioned as the alternative to human exposure to strenuous and repetitive labour, as well as to hazardous environments. The last decades have witnessed notable progress towards such prospect, aiding humans in complex and routinary tasks as, for example, autonomous exploration of remote and inaccessible underwater [88] and space [26] environments, high-precision surgical assistance [76], search and rescue in disaster response [38] and, among many other breakthroughs, management of a variety of household chores [154], and more comfortable, efficient and safer transportation of people and goods [12].

A fundamental requirement to push the potential of robotics even further is the development of more autonomous, resilient, and effective planning algorithms [1]. Planning in robotics tackles the challenge of determining sequences of valid goals and actions that lead a robot, or a group of, to succeed in an assigned mission [52]. This is a complex problem that comprises multiple levels of analysis, e.g., mission, task and motion planning, and it is tightly linked to other ongoing robotic challenges, chiefly world perception and understanding, and system state estimation and control. This thesis focuses on the motion planning problem (also interchangeably referred to as motion synthesis throughout this manuscript), without losing sight of its connection and codependency to other essential robotic functionalities in the autonomy pipeline.

Planning in robotics

1.1 Context

In its most generic definition, motion planning seeks a collision-free, feasible and continuous trajectory from a robot (start) state to another (goal state) [95]. Such a challenge is typically addressed with motion synthesis policies, commonly referred to as motion planners, which

Motion planning

3

Planning
space
process observations of the world to calculate feasible robotic actions in the form of trajectories subject to a set of given problem specifications (see Figure 1.1). All elements that constitute the planning problem are represented in a particular, or combination of, planning spaces (e.g., work, configuration, state, control, belief) such that the planners can search for constraint complying trajectories. We elaborate on the discussion about planning spaces in Section 2.1.
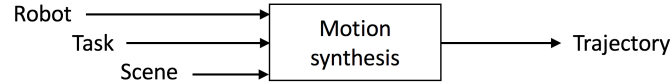
Figure 1.1: Motion synthesis aims to compute a trajectory that allows a robot to conduct a task in a particular scene. The elements (robot, task, scene) define the motion planning problem.

Algorithmic
properties
Ideally, a motion planner should always find a trajectory when a solution exists (algorithmic completeness), with the best possible cost (algorithmic optimality) subject to a desired objective (e.g., length, time, smoothness, human-likeness), in no time (algorithmic efficiency), and regardless of the planning problem (algorithmic scalability). See Section 2.1 for an extended discussion. In practice, however, solving the underlying motion planning problem is generally complex, and it gets harder with the dimensionality of the robotic system, the constraints imposed by the task, the complexity of the environment, and the necessary algorithmic properties. Many robotic applications are exposed to all these challenges, thus highlighting the need for efficient strategies for robotic motion synthesis.

The robotics literature is extremely rich in frameworks, algorithms, and techniques for motion synthesis. Each approach offers different algorithmic properties, thus each being particularly suitable to cope with a specific type of motion synthesis problem; a universal approach to motion synthesis that performs flawlessly regardless of the planning problem, i.e., robot, task and scene, remains an idealised conception. Generally, we can distinguish two main algorithmic trends to motion synthesis: exploratory and exploitative. We overview some of the methods under these two strategies below, but delve into their review and discussion in Chapter 2.

Exploratory
search

Plan from
scratch
Motion synthesis methods of exploratory nature undertake search routines to discover the connectivity of the space, aiming to find a continuous sequence of collision-free motions that lead to a trajectory from start to goal. Traditionally, such exploratory search is conducted from, scratch, i.e., without guidance over the entire planning space. In settings of reduced complexity and dimensionality, an option is to tackle the problem analytically, such as with potential fields [83, 84] or its randomized path planner (RPP) [14, 15] variant. Another family of algorithms search for solutions on a discretised version of the space, e.g., Dijkstra's [41], A* [56], D* [155, 156], or R* [102]. Their discrete support is usually built following inherently

rigid structures, such as grids or lattices, which do not always scale well to more complex settings. Alternatively, sampling-based algorithms sample the space randomly to build a connected graph of configurations. Seminal algorithms under this category include the probabilistic roadmap (PRM) [81], rapidly-exploring random tree (RRT) [94] and its bi-directional variant RRTConnect [91], expansive-spaces tree (EST) [65, 62, 63], and their corresponding optimal versions. In general, exploratory methods are algorithmically probabilistic complete, but their underlying exploratory search can be inefficient and lead to long computation times.

Oppositely, exploitative approaches to motion synthesis attempt to leverage some information related to the underlying problem to efficiently find collision-free trajectories from start to goal. Useful information can be that of any nature hinting relevant robotic behaviour to a particular or a family of planning problems, such as experienced robot invariant constraints in form of states and motions (e.g., [11, 157, 73, 110]) in standalone, or those correlated with geometric features of the scene (e.g., [178, 67, 98, 34, 114, 33]) and task information (e.g., [150, 131, 165, 117]). In the literature, such an exploitative approach is particularly attractive to well-defined planning problems that are repetitive and as such, relevant features can be computed offline and exploited on deployment. On this line, an option lies in the learning realm, where a-priori self- or externally-experienced relevant features in resembling problems are encoded and exploited with learning by demonstration (LbD) [19, 9, 147], reinforcement learning (RL) [74, 89] or transfer learning (TL) [163] techniques. In a different vein, repairing, guiding and optimising schemes [176] exploit problem-relevant features, a.k.a., initial seeds or warm starts, by exploring their vicinity for a solution either with heuristics or analytic gradients. In cases where the planning problem is not sufficiently well-structured to pre-compute relevant features, these can be computed on-the-fly, to then pursue similar exploitative strategies as those named above. Examples of works adopting this online strategy include, but are not limited to, multi-resolution frameworks that solve the underlying problem in sequentially informed stages, e.g., kinodynamic planning by interior-exterior cell exploration (KPIECE) [160] or synergistic combination of layers of planning (SYCLOP) [137, 136], and hybrid approaches that interleave planning from scratch and exploitative schemes, e.g., incremental trajectory optimization for motion planning (ITOMP) [129]. Overall, exploitative methods are extremely efficient to synthesise motion plans when provided with features that lie nearby a region where a solution to the underlying problem exists.

Exploitative search

Relevant features

Adopting either (exploratory or exploitative) motion synthesis approach has several implications in terms of design and performance. Exploratory approaches are more versatile as they exhaustively search for a solution over the entire space. Despite this implying minimal

design of the planning scheme, on deployment, significant time might be spent exploring irrelevant areas of the planning space that are unlikely to contribute to a solution. This issue is notably relevant in complex problems, to the extreme that planning from scratch may become intractable. Instead, well-framing an exploitative strategy in a particular application can come with serious performance benefits. Nonetheless, identifying relevant features to a large group of problem instantiations is not trivial, and formulating an exploitative scheme that allows for the encoding of problems, such that their representation is widely transferable and generalisable to other planning queries, remains an open area of ongoing research.

1.2   Problem Statement

Existing approaches to motion synthesis rely on computational techniques that are entirely exploratory or inherently data-driven. As such, these approaches are brittle and incapable of efficiently accounting for the variety of complex behaviours necessary for autonomous planning [119]. Comparing motion synthesis capabilities in robotics with that of animals and

Biomimetic motion synthesis

humans reveals significant scope for improvement. From a biomimetic point of view, animals and humans are capable of efficiently and safely trace plans in a wide range of environments and conditions, even when those are entirely novel. In cognitive sciences, one of the theories behind the efficiency of biomimetic motion synthesis supports that such process is driven by a vast repertoire of behavioural models which is continually expanded and adjusted over time and experiences [60, 115, 10, 117]. Hence, adopting such a strategy for robotic motion synthesis seems to be a promising venture to enhance its current capabilities.

We envision an analogous strategy to that described above to support more efficient and adaptable robotic motion synthesis. In particular, we foresee behavioural models to pose planning-specific problems in an abstract space, such that their similarities are accentuated and thus, problem-relevant features can be transferred and exploited across similar planning contexts. Therefore, as schematised in Figure 1.2, the concept of behaviour-driven exploitative motion synthesis comprises the following steps:

ABSTRACTION   process that poses the underlying planning problem differently, hereinafter referred to as *abstract problem*, to aid on extracting *relevant features*. The abstraction should capture the key aspects of the motion planning class the underlying problem belongs to. An excessively discriminating abstraction might prevent extracting useful features to successful exploitation.

EXTRACTION   process that analyses the *abstract problem* to compute a set of *relevant features* for the exploitation stage. An ideal extraction would allow

retrieving *relevant features* that are transferable to multiple instances of a motion planning class, and do so generically enough to allow pre-computing and storing the corresponding features in models or look-up structures, or, at least, allow computing the features rapidly on-the-fly.

EXPLOITATION process that leverages the extracted *relevant features* to guide the search of a solution for the underlying problem. An ideal exploitation should cope with the fact that the provided features might not always be closely related to the undergoing problem. As thus, a planning scheme that weighs exploration with exploitation of relevant areas of the planning space is important.
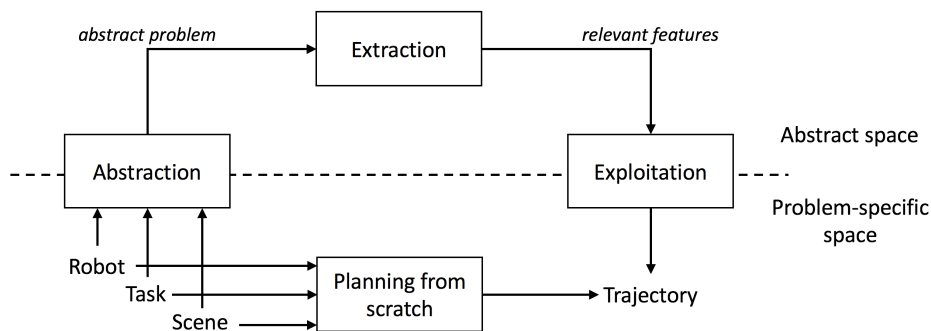
Figure 1.2: Conceptual idea of motion synthesis via abstractions. Alternatively to planning trajectories from scratch, an abstraction of the underlying problem and its corresponding solution can be leveraged to compute motion plans. The hypothesis on such an approach is for it to enable solving more complex problems, and do so more efficiently, than planning from scratch methods.

With the idea described above, we aim at driving the motion synthesis with behavioural inductive bias, i.e., with hints on the expected robotic behaviour on a given planning problem, to avoid exhaustive exploration of the potentially infinite behavioural possibilities. Instead, computational efforts are focused on promising areas of the planning space. Overall, the feasibility of undertaking such a behaviour-driven motion synthesis approach at large depends on the ability to obtain, identify and exploit relevant behavioural features for a given planning problem. In this regard, we raise the following research questions:

RQ1 *How might relevant features be identified, stored, and retrieved for recurrent tasks?*

RQ2 *How should relevant features be leveraged in favour of efficient motion synthesis?*

RQ3 *How might planning via abstractions help solving ever-changing planning problems?*

1.3   Thesis Objectives, Contributions and Outline

In this thesis, we contribute strategies suitable for rapid and scalable motion planning in a variety of fields. We argue that grounding robotic motion synthesis on adaptive behaviours may offer an exciting opportunity to trade-off algorithmic efficiency and versatility to address a wide range of motion planning problems. Our particular approach, as schematised in Figure 1.3, is twofold: first, hypothesise the sort of behaviour that addresses a designated planning problem, to then drive the motion synthesis via abstraction to infer behaviour-relevant motions.
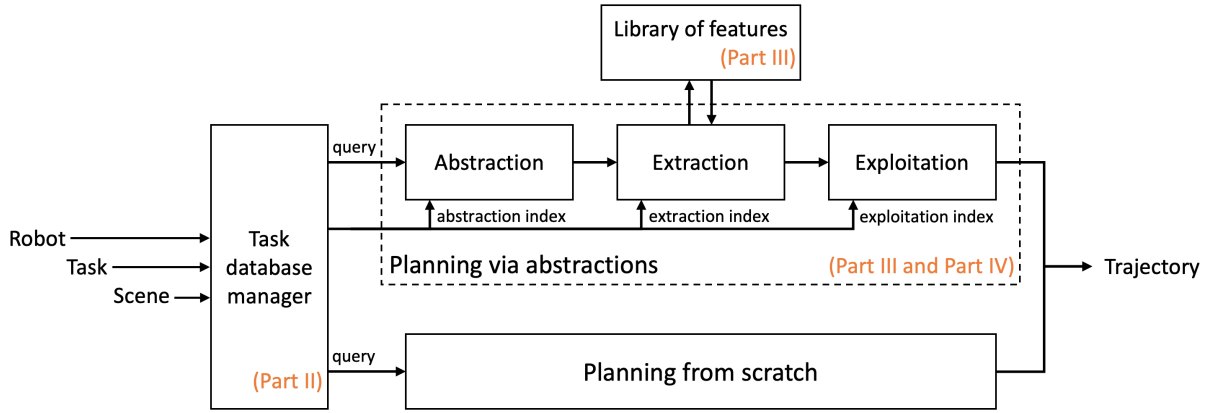
Figure 1.3: Planning scheme to guide the motion synthesis process through categorically related parts of the system's behavioural space. The `task manager` employs a relational database to decide whether any of the known `planning via abstraction` pipelines relates to the designated task, or it is better off `planning from scratch`. Different behaviour-driven planners are proposed according to the possibility to extract relevant features prior to deployment.

We employ the planning scheme in Figure 1.3 to address the research questions posed above along with several contributions. While the contributing approaches to each research question share the research line of behaviour-driven motion synthesis, their technical development does not necessarily build onto each other. In order to contextualise and better understand the real impact of our developments, we first provide a comprehensive review and discussion on existing strategies for motion synthesis in Chapter 2; special emphasis is placed on strategies related to each research question (see Section 2.3.1, Section 2.3.2 and Section 2.3.3 for RQ1, RQ2, and RQ3, respectively). Then, based on common limitations identified in the existing motion planning algorithms, the main contributions of the work presented in this thesis are gathered and distributed throughout this document as follows:

Part II   (addressing RQ1) We introduce a general framework with behavioural semantic annotation that enables the storage and bootstrap of robot behaviour (see

Chapter 3). We first discuss the conceptual idea of the framework and its capabilities (see Chapter 4), and then extend it to include affordances information (see Chapter 5). We demonstrate the framework using off-the-shelf dynamic movement primitives (DMPs) to encode and retrieve behavioural abstractions of high-dimensional kinematic systems in grasping and manipulation tasks, and discuss our findings in context with the main objective of this thesis (see Chapter 6).

Part III (addressing RQ2) Building on the capability to bootstrap relevant behavioural abstractions, we present two independent strategies to efficiently leverage offline-computed problem behavioural abstractions (see Chapter 7). The first contribution on this line is an online modulation of DMP-encoded behaviours based on a geometry-tuned potential fields formulation (see Chapter 8). The second contribution are two the novel experience-driven random trees (ERT) and ERTConnect planners that leverage from prior abstract solutions (see Chapter 9). We demonstrate and benchmark our approaches on kinematic manipulation tasks that differ from the prior problem abstractions at different levels, and discuss our findings in context with the main objective of this thesis (see Chapter 10).

Part IV (addressing RQ3) We study the applicability scope of behavioural abstractions in a complex, ever-changing planning problem (see Chapter 11). Our contribution is a multi-layered planning strategy capable of abstracting and exploiting online robotic behaviour in a high-dimensional, highly-constrained, and uncertain space (see Chapter 12). We demonstrate our approach in a navigation task through unknown environments, in laboratory experiments and in field trials, and discuss our findings in context with the main objective of this thesis (see Chapter 13).

The contributions on efficient motion synthesis via behaviour guidance along the three core parts of this thesis are summarised in Table 1.1, altogether with an overview of the robot platforms used in this thesis and their characterisation for motion synthesis in Table 1.2. Overall, our contributed behaviour-driven planning strategies augment the robotic capabilities to deal with more complex planning problems, and do so more effectively than related approaches in the literature by computing better quality paths in lower response time. A more in-detail summary of the work in this thesis is provided in Part V, with a review of the technical contributions in Chapter 14 and an outline of exciting avenues for future work in Chapter 15.

The work on motion synthesis developed in this thesis has contributed to several research lines in the scope of the ORCA Hub - project on offshore robotics for certification of assets. Also, several of our contributed motion synthesis algorithms have been made publicly available to the robotics community (links provided in the corresponding chapters).

	Abstraction	Abstract problem	Extraction	Relevant features	Exploitation	Deployment and [Platform (as in Table 1.2)]
Chapter 4	Semantic task-labelling	Task-related problem	Kinaesthetic teaching	Library of DMPs	User-guided DMP roll-out	Manipulation, novel scenarios [iCub]
Chapter 5	Semantic task-labelling	Task-related problem	Kinaesthetic teaching	Library of affordances	Affordance-guided DMP roll-out	Grasping and manipulation, novel scenarios [PR2]
Chapter 8	Projection onto parameter space	Parametric descriptor	Feed-forward grid-search	RC-NN weights	DMP + RC-NN-based coupling term roll-out	Manipulation, novel scenarios [Franka]
Chapter 9	Semantic task-labelling	Task-related problem	RRTConnect	Library of paths	ERT and ERTConnect	Manipulation, novel scenarios [Fetch]
Chapter 12	Projection onto 3D workspace	Problem at workspace	RRT*	Geometric lead path	Lead-guided SST	Navigation, unknown environments [SPARUS II and quadrotor]

Table 1.1: Overview of the approaches for motion synthesis via abstractions (as detailed in Figure 1.2) contributed in this thesis. Note that *library* in RQ1's scope (Chapter 4 and Chapter 5) refers to a collection of behavioural samples where each exemplifies a particular task, whereas as part of RQ2 (Chapter 8 and Chapter 9) it is a set of behavioural samples exemplifying the same task. ☐ Computed offline. ☐ Computed on-the-fly.

Platform	Middleware	Planning Group	DoF	Planning Space	Usage
iCub [112]	YARP [111]	Dual-arm with torso	17	End-effector space (SE(3))	Simulation and real robot
PR2	ROS [139]	Single-arm with torso	8	End-effector space (SE(3))	Simulation and real robot
Franka	OROCOS [25]	Single-arm	7	End-effector space ($\mathbb{R}^3$)	Simulation and real robot
Fetch [174]	ROS [139]	Single-arm with torso	8	Configuration space ($\mathbb{R}^8$)	Simulation and real robot
SPARUS II [32]	ROS [139]	Mobile base	2	Belief (SE(2) $\times \mathbb{R}^{3\times3}$) and control ($\mathbb{R}^2$) spaces	Simulation and real robot
Quadrotor [113]	ROS [139]	Mobile base	5	Belief (SE(3) $\times \mathbb{R}^{6\times6}$) and control ($\mathbb{R}^3$) spaces	Simulation

Table 1.2: Overview of the robot platforms used in this thesis, and their characterisation for motion synthesis.
See Section 2.1 for details on the technical nomenclature.

Background

"Standing on the shoulders of giants"

— Bernard of Chartres

Although motion synthesis is the central research area behind the work presented throughout this thesis, the validation of our contributions in real-world scenarios has involved other fields of study such as perception, mapping, state-estimation, and control. A comprehensive review of the state-of-the-art of all these areas would prove to be lengthy, diffuse, and unhelpful in contextualising this thesis' contributions. Therefore, this chapter mainly focuses on reviewing strategies for motion synthesis. We start in Section 2.1 with an introduction to the fundamentals of motion synthesis. Then, in Section 2.2, we review methods that undertake exploratory routines to solve motion planning problems; the study of these methods is not intended to be in-depth, but to provide a complete overview across the most common planning strategies in the literature. In Section 2.3, we visit existing strategies that attempt to inform the motion synthesis process to bound the exploratory load in relevant regions of the planning space. We conclude in Section 2.4 with a discussion on this thesis' contributions towards ongoing motion synthesis challenges.

2.1 Motion Synthesis: Fundamentals

All elements that constitute the planning problem are represented in a particular, or combination of, planning spaces. Motion synthesis, then, aims at finding a collision-free trajectory between two points on such space. Some common planning spaces are:

- **Task space:** represents the space in which a robot's task can be naturally expressed. Only knowledge on the task is required, not about the robot, to define the task space. For instance, if the task is to plan for the position of a point robot on a plane, then the task space is $\mathbb{R}^2$,

while if the robot was a rigid body, then the task space would be SE(2). Analogously, in a volumetric environment, the task space would be $\mathbb{R}^3$ and SE(3), respectively.

- **Workspace:** represents the set of two-dimensional (2D) or three-dimensional (3D) positions that can be reached by a robot. The workspace can also include orientation. The set of positions reachable with all possible orientations is referred to as the dexterous workspace. In the literature, the workspace of a manipulator is sometimes called the end-effector space.

- **Configuration space:** represents the set of all possible configurations that a robot can adopt. The configuration of a robot is determined by the state of all its DoFs. The configuration space of a point and solid body robot constrained in a 2D workspace would be $\mathbb{R}^2$ and SE(2), whereas if constrained in a 3D workspace, it would be $\mathbb{R}^3$ and SE(3), respectively. In the literature, the configuration space of a manipulator is also referred to as joint space, which commonly is $\mathbb{R}^n$, with n being the number of DoFs of the manipulator. When the dimensionality of the robot's configuration space is larger than that of the workspace, the robotic system is redundant, as certain poses in the workspace are reachable with multiple configurations.

- **State space:** represents the set of all possible states that a robot can adopt at each time differential either in the work or configuration space. A state completely describes the system by expanding the robot's positional knowledge with, for instance, velocity and acceleration. Planning in the state space is generally challenging as its span is potentially infinite.

- **Belief space:** represents a particular space (e.g., configuration or state space) in a stochastic manner. The underlying configurations and states are probabilistic distributions around their nominal estimate (beliefs), whose covariance accounts for the uncertainty of any element in the planning problem (e.g., motion model, robot localisation, wind and current perturbations).

- **Control space:** represents the set of actions (controls) that a robot can kinematically or dynamically execute from a particular state (e.g., dictates that a car cannot move sideways).

Each space can be divided into two subspaces: the free and occupied. The free space is the set of feasible configurations, states or beliefs that are not in collision with any obstacle. Contrary, the occupied space is the region of the space that involves a collision with the environment.

The choice of planning space depends on the considered constraints. Namely, we can distinguish between geometric and kinodynamic constraints. The former assumes that the robotic system can move instantaneously in any direction (i.e., kinematics and dynamics are negligible). In that case, planning in the configuration space suffices to validate configurations against the limits of each DoF and collisions. The latter type of constraints involves ensuring the feasibility of the computed motions, for which planning in the state space, altogether with feasible actions from

the control space, is essential. Additionally, one might consider probabilistic constraints, which require planning in the belief space and validating collisions against uncertain estimates.

Over a planning space, a planner seeks a collision-free trajectory that solves the motion synthesis problem. Some of the properties that are of interest in a planner are:

- **Efficiency:** refers to the ability to retrieve a valid motion plan with a reasonable amount of time. The tractable amount of time is dependant on the robotic application. Planners with nice planning and path properties, while offering low computation time, remain a challenge.

- **Completeness:** refers to the ability to find a solution when one exists. To that, lossless representations of the planning space and environment are required. However, completeness is traded off against efficiency. Two weaker notations exist: *resolution complete* and *probabilistic complete*. The former ensures finding a path, if one exists, when the discretisation of the space is fine enough to capture all relevant information. The latter refers to the increased chances of finding a solution, if any exists, as the time spent planning tends to infinite.

- **Optimality:** refers to the ability to retrieve the solution with the best cost (see below for some discussion on possible metrics). Analogously to the case of completeness, there are two weaker notations: *resolution optimal* and *asymptotic optimal*.

- **Scalability:** refers to the potential to employ a planner in problems of different complexity (i.e., 2D against 3D task space, or planar robot against a high-dimensional system), while maintaining the algorithmic properties and keeping computations tractable.

Independently from the properties of a planner, it might be of interest to compute motions of certain quality subject to one, or a set of, metrics. We introduce some common metrics next:

- **Length:** seeks short trajectories. In geometric spaces, optimal length corresponds to minimum execution time, but that is not necessarily true for kinodynamic spaces.

- **Smoothness:** seeks to avoid jerky motions to, for instance, preserve mechanically critical DoF or capture more steadily video stream with a camera.

- **Naturalness:** seeks human-like motions to maximise the confidence of operators working in proximity with robots, or the comfort in human-robot teams.

- **Clearance:** seeks to keep a minimum, or maximise, distance away from the obstacles. Although computationally expensive, high-clearance solutions reduce the risk of collisions due to real-world factors unrepresented in the planning space (e.g., uncertainties).

- **Safeness:** seeks to maximise the robot safety along a path. Arguably, similar aim to that of the clearance metric, but it can cope with more complex aspects such as uncertainties.

- **Information:** seeks a path that maximises information about the environment, thus promoting trajectories that pass by feature-rich regions of the environment.

There exist two main trends to compute trajectories subject to a desired quality metric. One is to employ an optimising planner that explores the planning space while improving the solution cost. While optimising planners are more computationally demanding, an alternative is to calculate a non-optimised trajectory and post-process it to improve the cost.

2.2   Exploratory Motion Synthesis

The most common strategy for robotic motion synthesis is by planning from scratch, i.e., without guidance over the entire planning space. Such an approach is characterised by relying on search routines to discover the connectivity of the space, aiming to find a sequence of motions that lead to a collision-free trajectory from start to goal. We review relevant exploratory methods below.

An early approach to motion synthesis, particularly to mobile base navigation in unknown 2D environments, was with a pre-defined set of perception-driven behavioural rules. Bug-based methods are sensor-based reactive motion planning approaches, in which a robot moves towards a global goal while avoiding obstacles with limited (local) knowledge of the environment. All bug-based algorithms build on two basic behaviours: go straight towards the goal and follow an obstacle's boundary. Bug-based variants differ in the sensory information used to detect the obstacles and its usage to switch between the basic behaviours. The Bug1 and Bug2 algorithms use tactile sensors to detect and contour obstacles until the goal is achieved [107]. The Tangent Bug leverages a non-zero range sensor to detect obstacles before contact and estimate the optimal direction to circumnavigate them [77]. These approaches lack completeness guarantees, and their applicability is limited to low-dimensional problems where optimality is not a must.

When the environment is revealed in advance, potential fields offer a reactive analytical approach for motion synthesis [83, 84]. Such an approach uses the problem's structural elements (e.g., goal configuration and obstacles) to compute potential functions, also referred to as energy functions. Particularly, the system is guided towards the goal with an attractive energy component, while it is pushed away from the obstacles with a repulsive energy component. The addition of all energy landscapes defines the total potential field, the negative vector gradient of which (gradient descent) is used to guide the robot throughout a collision-free path towards the goal. Potential fields may suffer from local minima and thus, generally lack algorithmic completeness. The RPP deals with this situation by using the gradient descent in conjunction with random walks, thus inserting some stochasticity in the planning process [14, 15]. However, the performance of the

RPP is highly dependent on parameter tuning and yet, computing the potential fields may be computationally expensive for dynamic environments or high-dimensional systems.

A widely adopted approach to make complex planning problems tractable is formulating them as a graph. Utilising graph theory to motion synthesis involves (i) approximating the connectivity of the space as a graph, and (ii) searching the graph for a collision-free path. Approaches in this category differ on the graph support type (e.g., roadmaps, grids, lattices, trees), and the strategy undertaken to build it (e.g., cell decomposition, search-based, sampling-based).

Cell decomposition methods represent the collision-free space by a set of non-overlapping, non-uniform cells. Decomposition techniques mainly differ in the usage of obstacle information to compute the cells, e.g., visibility graphs [106], generalised Voronoi diagrams (GVD) [50], freeway net [24] and, among many others, silhouette [93]. The computed decomposition serves as an adjacency graph, where cells correspond to nodes and their adjacency to edges. Such topological graph contains information about all possible routes across the environment, also referred to as a roadmap. Formally, a roadmap defines a subset of the planning space resulting from the union of one-dimensional curves, in which any start and goal configuration contained in the collision-free space can be connected by a path that meets the following requirements: (i) accessibility: there is a path from the start configuration to some (accessing) node in the roadmap (ii) departability: there is a path from some (departing) node in the roadmap to the goal configuration, and (iii) connectivity: there is a path connecting the roadmap's accessing and departing nodes. Therefore, in the context of cell decomposition techniques, a path is found by (i) determining the cells containing the start and goal configurations, and (ii) searching a path within the roadmap with graph search techniques, such as Dijkstra's [41] or A* [56] algorithms. Solving the graph is computationally inexpensive, but the main burden on employing cell decomposition for motion synthesis resides in the complexity of the decomposition itself.

Search-based methods employ adjacency information to approximate the planning space on a discrete support. In its simplest form, the planning space is represented as a uniformly discretised grid. The resolution of the grid implies a trade-off between computational expenses and space representation accuracy; while coarse discretisations permit faster searches but may fail to find paths across narrow passages, finer discretisations may allow solving more complex scenarios at the cost of longer computation times. Hence, search-based methods are resolution complete. Heuristics are employed to promote reaching the goal configuration before visiting or unfolding the entire grid. These heuristics typically come with strong theoretical guarantees on completeness and sub-optimality bounds [132], and stipulate the main differentiating aspect between search-based variants. Dijkstra's algorithm [41] and the A* [56] are suitable for low-dimensional problems dealing with static environments, the anytime repairing A* (ARA*) [101]

gives the best plan it can within the allowed time budget, the dynamic A* (D*) [155, 156] and D* Lite [90] deal with dynamic environments, and hybrid planners combine a range of features to cover a broader range of problems, such as the anytime dynamic A* (AD*) [100] or the R* [102]. More elaborated search-based strategies employ a lattice to define the adjacency of robot states and the allowed movement between them [135]. As discussed in Section 2.3.1, the allowed lattice adjacency induces some domain knowledge about the planning problem in the form of motion primitives. As thus, lattice graphs, in contrast to grid graphs, enable dealing with planning problems of higher dimensionality.

Sampling-based algorithms explore the connectivity of the planning space via sampling, thus not requiring its explicit representation as previous methods. Sampling-based techniques build graphs following a twofold procedure: (i) "sample" to randomly generate collision-free configurations, and (ii) "connect" to establish routes between the sampled configurations. This strategy is adopted in (a) multi-query planners, which build a roadmap that can respond to multiple motion synthesis problems, e.g., PRM [80] or its asymptotic optimal version PRM* [79], and in (b) single-query planners, which build a tree-like graph of configurations to solve a particular start-to-goal query, e.g., RRT [94, 96, 97], its asymptotic optimal RRT* [78] or bi-directional RRTConnect [91] versions. Sampling-based algorithms are probabilistic complete, i.e., when granted enough amount of time, they provide a solution if one exists [80, 13], and have an inherent Voronoi bias that promotes dispersion during the exploration [94]. There are many variants of these algorithms. Search tree with resolution independent density estimation (STRIDE) [53] analyses the most convenient tree expansion to prioritise less-explored regions (e.g., narrow passages). EST [65, 62, 63] adopts a branch-off strategy to expand a tree to avoid dependence on a notion of distance; distance metrics are not trivial in certain planning spaces. Some extensions tackle the lack of optimality of the basic sampling-based methods which often find solutions with unpredictable length and superfluous movements. The transition-based RRT (T-RRT) [71] and its optimal version T-RRT* [40] calculate low-cost paths that follow a costmap established over the planning space. Other variants, such as the Lazy PRM [20, 21], propose deferring the collision-check along edges until a path is needed, thus speeding up the exploration. Overall, the properties and modularity of sampling-based problems make them suitable to tackle a wide range of planning problems.

2.3 Exploitative Motion Synthesis

Exploitative approaches are particularly attractive to recurring motion synthesis problems. As such, problem information can be extracted offline, and leveraged to aid in solving related

queries. Such an approach involves some challenges, namely: acquiring, storing and retrieving problem-relevant features (see Section 2.3.1), and exploiting those in favour of rapid motion synthesis (see Section 2.3.2). Whereas for repetitive applications each challenge can be addressed independently, tackling them simultaneously opens up the possibility to deal with complex problems from which relevant information cannot be extracted in advance (see Section 2.3.3).

2.3.1 Feature Acquisition, Storage and Retrieval

In the autonomy pipeline, motion synthesis aids in enabling a robot to conduct a task in a particular scene. Any knowledge on the conforming elements of such a problem can be leveraged to inform the motion synthesis process. Examples of relevant problem information include, but are not limited to, experienced robot invariant constraints in form of states and motions in standalone (e.g., [11, 157, 73, 110]), or those correlated with geometric features of the scene (e.g., [178, 67, 98, 34, 114, 33]) and task information (e.g., [150, 131, 165, 117]).

There are different means to provide prior knowledge to a robotic system. The most conventional approach is via mathematical models describing the system's desired behaviour [54, 175]. Formalising a model to describe motion is generally complex, thus it is mostly restricted to defining low-level motion primitives. Nonetheless, these primitives within RL [74, 89] allow a robotic agent to self-learn more sophisticated planning policies via mental rehearsal or real-world interaction. The planners reviewed in Section 2.2 can also be used offline to collect samples and trajectories in similar motion planning problems. Instead of computing features from scratch, an option is to acquire them from an already experienced agent; TL [163] enables robot-to-robot information sharing, and LbD [19, 9, 147] for a human to demonstrate behaviour to a robot. The main challenge of these approaches is dealing with the possibly different anatomical constitution between agents [116].

Acquired features can be stored in a database in a raw format. Nonetheless, such an approach can rapidly lead to a prohibitive memory footprint when considering multiple features, and doing so for a variety of environments and tasks. There is a series of techniques that attempt to mitigate the storage burden, e.g., homotopy classification [18, 138], inference models [99] and sparse roadmaps [37, 118], among others. The strategy undertaken to encode and store the acquired data has a large impact on the required generalisation capabilities of the exploitative method when dealing with problem instantiations not reflected with features in the database.

Retrieving features from a database relevant to a given planning problem poses a challenge itself. An option is to exploit them exhaustively within a search-based or sampling-based scheme, thus restricting the exploratory nature of these algorithms onto a discrete support [54, 175].

This approach, however, may turn prohibitively expensive with databases constituted of an extensive set of behavioural samples. In this regard, a seminal work is the so-called "manoeuvre automata", a set of rules that generates complex behaviours by selection and concatenation of motion primitives [43]. More sophisticated problem-encoded information requires heuristics and metrics that aid in identifying the best candidate, or set of, to a particular problem. For instance, similarity metrics are widely used when dealing with database of trajectories (e.g., [11, 157, 73, 110]) or environmental information (e.g., [34, 33]). Another remarkable approach is the usage of relational databases to relate tasks with relevant features, e.g., [150, 131, 165, 117].

2.3.2   Feature Exploitation

Strategies to leverage relevant problem information for robotic motion synthesis have mainly built on learning, control and pure planning techniques. Generally, we distinguish two main trends: methods that exploit priors via modifying schemes (see Section 2.3.2.1) and guiding methods that exploit raw priors as reference (see Section 2.3.2.2).

2.3.2.1   Modifying Methods

A major area of research focuses on learning policies that capture the underlying objective of a given set of task demonstrations. Learning-based approaches for motion synthesis include iterative learning control (ILC) [22], RL [74, 89], TL [163] and LbD [19, 9, 147]. Among them all, LbD is closely relevant to this thesis. LbD aims at extracting relevant features from a demonstrated trajectory, or set of, such that the learnt policy can generalise to multiple task instances. Some mathematical supports to policy learning are Gaussian mixture model (GMM) [39], Gaussian mixture regression (GMR), Gaussian process regression (GPR) [144], DMP [70, 68], hidden Markov model (HMM) [141], locally weighted regression (LWR) [36], receptive field weighted regression (RFWR) [149] and locally weighted projection regression (LWPR) [169]. These methods are capable of computing plans quickly, but they struggle generalising to complex environments that involve significantly different task instances than those observed a priori [147]. Some authors have explored extending the generalisation capabilities of these methods with the concept of energy landscapes revised earlier in Section 2.2. To that, demonstrations are encoded as a LbD policy (primary model), whose underlying motion synthesis behaviour is modulated with an energy function (secondary model) based on proprioception and environmental feedback, e.g., online adjustment of force setpoints [48, 162], workspace and kinematic limits [48, 49], and obstacle avoidance [130, 61, 82, 142, 143, 66]. The generalisation scope of these methods is constrained by the capabilities of each model and their mutual interaction. Overall, learning methods confer rapid prior exploitation capabilities, but at the cost of low algorithmic scalability and completeness.

On the idea of following the gradient of an analytical function, we find optimisation-based algorithms. These formulate the motion planning problem as an iterative optimisation process subject to one or several cost functions. A cost function expresses specific requirements concerning the presence of obstacles, workspace and configuration limits, path smoothness, dynamical stability (for humanoid robots), and among many others, object manipulability (for manipulators). Regardless of the cost function definition, optimisation-based algorithms mainly differ in their use of derivative and environment encoding. Covariant Hamiltonian optimisation for motion planning (CHOMP) [146] and its extension to handle constraints [42], trajectory optimisation (TrajOpt) [151], Riemannian motion optimisation (RieMO) [145] and k-order Markov optimisation (KOMO) [167] optimise a derivable problem encoding via gradient-descent optimisation. Approximate inference control (AICO) formulates a probabilistic trajectory model and uses iterative approximate inference to solve the non-linear stochastic optimal control problem [166]. Similarly to RL techniques, stochastic trajectory optimisation motion planning (STOMP) [75] and path integral policy improvement (PI^2) [164] formalise the motion planning problem as a stochastic, derivative-free Monte Carlo problem. In a similar vein, elastic bands model an initial seed (trajectory) as a chain of virtual springs acting in contraction to a set of via points and repulsion to the presence of obstacles [140]. To find the equilibrium point between the contraction and repulsion forces applied on the elastic band, this method iterates using a downhill gradient search. This approach was later extended to elastic strips, which uses the same concept of springs but with control points [23, 92], and to establish the edges between the nodes of a manipulation roadmap [177].

2.3.2.2 Guiding Methods

There are mainly two orthogonal strategies to guide the motion synthesis process with relevant information about the planning problem: repairing a previously experienced space connectivity, and biasing the sampling into task-relevant areas.

At its core, repairing techniques aim at improving the connectivity of the space, either to optimise the connectivity cost or to reestablish connections between states that are no longer valid due to changes in the space. Basic techniques seek to improve a given trajectory considering new connections between existing vertices (path pruning) or states along the edges (path shortcutting) [16, 51, 58, 64]. These methods are generally fast, but their limited search makes them better purposed as a post-processing step to path optimisation. More advanced techniques invoke a planner locally to reestablish or optimise the connectivity between two states. Lightning [17] initially retrieves a unique trajectory from a library based on the start-goal proximity of a trajectory and the current query, as well as votes on the number of disconnected segments. Then, the repair step employs the RRTConnect to attempt to

re-connect the end-points of segments originated by variant constraints, e.g., obstacles. Alternatively, experiences can be jointly stored as a graph [134, 37]. Experience Graphs [134] build a roadmap of experiences to then search it or explore other routes according to some heuristics. Thunder [37] creates a sparse roadmap from all experiences. The roadmap is iteratively queried via the graph-based planner A^* until a valid path is found. If the graph does not contain a valid path, candidate paths, if any, are considered for repairing. The repairing invokes RRTConnect to reconnect the disconnected states along the candidate paths.

Biasing the sampling involves guiding the exploration towards task-relevant regions of the planning space. A common approach is to leverage from geometric features of the workspace to guide the sampling in the planning space, e.g., [178, 67, 98, 34, 114, 33]. In [178], reinforcement learning is used to adapt the sampling in the configuration space by extracting features in the workspace. The authors of [67] employ a conditional variational autoencoder that generates samples lying in relevant areas. On the same line, a database that maps workspace decomposition to local samplers is proposed in [34, 33]. Strategies that bias the sampling can significantly speed up queries, but they rely on identifying familiar workspace features to infer relevant samples in the configuration space. Therefore, their applicability is mainly limited to task instances that resemble those observed a priori, leading to a lack of generalisation to new environments.

2.3.3 Interleaved Feature Acquisition and Exploitation

There are planning problems that are not sufficiently well-structured to pre-compute relevant features in advance, or that lack metrics to choose a proper candidate from a database of problem-relevant features. As such, the exploitative approaches reviewed until this point are commonly used for systems that are mostly destined to live in a known static environment or to solve a well-defined problem repeatedly [170]. An alternative is to simultaneously compute and exploit relevant features on-the-fly. Notably, such an approach reduces the dependence on good quality data, as well as it removes the need of storage and retrieval of pre-computed information. Such strategy in the literature is commonly adopted to deal with motion synthesis problems that are intractable with traditional planning from scratch strategies, and do so more efficiently.

An interesting approach is the ITOMP algorithm, which interleaves planning and optimisation; the planner is given a fixed time budget to find a solution, which is then used as a warm-start for the optimiser [129]. Similarly, multi-layered planning frameworks seek a lead to guide (warm-start) the search of following stages [152]. In [137, 136], the authors introduced a synergistic two-layered planner: the high-level planner uses discrete search to initially determine those candidate regions (from a decomposed representation of the environment), which might contain

part of the final solution; a low-level planner employs a sampling-based motion planner to find a solution; the interface between layers updates the candidate regions according to the considered constraints. However, the proposed combination of planners does not guarantee asymptotic optimality, and the discrete planner becomes slow for high dimensional problems. Palmieri, Koenig, and Arras presented the Theta*-RRT scheme, which first uses the Theta* path planner to compute a lead path, which is then employed to bias the search of the RRT planner [128]. This approach, however, lacks asymptotic optimality guarantees given that the second planner is an RRT. More recently, a multi-layered approach based on the RRT* as a lead planner and the SST as the final planner has been proposed in [168]. The final planner's search space is strictly constrained around the lead path, raising concerns about the completeness guarantees of the overall architecture. Another family of motion synthesis planners seek to optimise on-the-fly the sampling and local connection of configurations. To that, the search space is bounded with environment-related information (e.g., deformable RRT (DRRT) [57], Dancing PRM* [86], and Volumetric Tree* [85]), and problem-related heuristics (e.g., Informed RRT* [45] and adaptively informed trees (AIT*) [158]). Some of these approaches have been extended to reuse batches of samples across iterations that fall within the search limits, e.g., batch informed trees (BIT*) [46, 44] and regionally accelerated BIT (RABIT*) [35].

2.4 Discussion

In this review, we have categorised the rich repertoire of existing frameworks, algorithms, and techniques for motion synthesis into two main trends: exploratory and exploitative. We have discussed the algorithmic properties that different methods in the literature confer, and their suitability to cope with certain types of planning problems. Yet, despite the numerous approaches to motion synthesis that exist, current methods struggle to satisfy the motion planning demands of emerging robotic applications. Examples of ongoing challenges include, but are not limited to: high-dimensional spaces, kinodynamic constraints, unstructured, dynamic and unknown environments, human-robot and robot-robot teams, and real-time applications. These challenges require more efficient strategies for robotic motion synthesis that enable dealing with more complex problems in tractable time.

In this thesis, we contribute strategies suitable for rapid and scalable motion planning in a variety of fields. Our particular approach is to ground robotic motion synthesis on adaptive behavioural abstractions in the form of trajectories. Research on that proposition leads to several technical contributions on: feature acquisition, storage and retrieval (see Part II), feature exploitation (see Part III), and interleaved feature acquisition and exploitation (see Part IV).

These developments are build on the basis of their associated limitations (see Section 2.3.1, Section 2.3.2, and Section 2.3.3, respectively).

Our contributions on efficient motion synthesis via behaviour guidance augment the robots' capabilities to deal with more complex planning problems, and do so more effectively than related approaches in the literature by computing better quality paths in lower response time. We demonstrate our contributions, in both laboratory experiments and field trials, on a spectrum of planning problems and robotic platforms ranging from high-dimensional humanoids and robotic arms with a focus on autonomous manipulation in resembling environments, to high-dimensional kinematic motion planning with a focus on autonomous safe navigation in unknown environments.

Part II

BOOTSTRAPPING PROBLEM-RELEVANT ROBOTIC BEHAVIOUR

Critical Review - Opening

In this part of the thesis, we focus on the challenge of acquiring, storing and recalling abstractions with the purpose of bootstrapping robotic behaviour. Enabling a robot with such abilities is essential to leverage abstractions in favour of rapid computation of motion plans in varied task contexts. Thus, our aim is to formalise a strategy that enables a robotic system to acquire and store abstractions, and exploit them to autonomously bootstrap problem-related behaviours.

3.1 Objectives and Contribution

Expectations on fully autonomous robots are for them to be adept at tackling a wide range of tasks, such as doing the dishes, walking the dog, or, among others, tidying up. Instinctively, each of these tasks can be inherently associated with a particular behaviour or skill. In this thesis, we argue that such intuition can be leveraged to induce some bias in the motion synthesis process. In order to enable a robot bootstrapping problem-relevant behaviour, we need the ability to identify, encode, and recall abstractions subject to different tasks. To these objectives, we incrementally present the contributing framework to bootstrap robotic behaviour as:

Chapter 4   A strategy to bootstrap problem-relevant robotic behaviour in support of the motion synthesis process. The proposed framework employs a relational database to relate planning problems, particularly assigned tasks, with relevant robotic behaviour. Behavioural abstractions are formalised as trajectories or trajectory-derived policies. Moreover, we experiment with the concatenation of multiple abstractions to compose more complex behaviours that are not intrinsically represented in the framework.

Chapter 5   An extension of the proposed framework that promotes robot autonomy by interpreting high-level task commands. To that aim, we leverage parallel research to this thesis on robot affordances to formalise a more sophisticated relational database that considers properties of the environment, task and target object to decide the best behavioural abstraction. Moreover, we propose a fast-forward evaluation mechanism to rank the suitability of multiple behavioural samples.

3.2   Methodology

A relational database is as structure that encompasses a collection of data items with pre-defined relationships between them. In this part of the thesis, we exploit the structure offered by relational databases to relate planning problems with suitable robotic behaviours. An initial iteration in this direction was conceptual as a viability test of employing relational databases to bootstrap robotic behaviour (see Chapter 4). In particular, we explored look-up structures to associate the definition of a task in the planning problem with a relevant behavioural abstraction. We elaborated on such initial proof-of-concept work to formalise more sophisticated relational databases that consider multiple aspects of the given planning problem (see Chapter 5). In particular, we leverage from findings in parallel research to this thesis on robotic affordances [8, 6] to contextualise, in a relational model, the designated task objectives with the target objects' affordances, and infer the most suitable robot behaviour in accordance with the expected effect.

Robotic behaviour can be encoded onto different supports (see Chapter 2). Our proposed framework captures behaviour associated with particular tasks in the form of trajectories. This is, we formalise problem abstractions that extract and leverage motion plans in a task-related (abstract) problem to efficiently drive robotic behaviour in new instances. Trajectories are of particular interest to us because they can be easily obtained with, for instance, human demonstrations, existing planning approaches, large-scale motion capture datasets, or widely available online video footage (e.g., from YouTube); across this thesis, we employ the former two approaches (human demonstrations in Chapter 4, Chapter 5 and Chapter 8, existing planning approaches in Chapter 9 and Chapter 12). Moreover, as the outcome of a motion synthesis problem is trajectories, these can be incorporated into the robot's behavioural knowledge about a task as it experiences new instances.

There is a set of challenges associated with the usage of trajectories as behavioural support, namely, defining suitable metrics for look-up across behaviours and within each one, transfer across planning problems, and the trade-off between storage capacity and generalisation capabilities. The framework presented in this part of the thesis addresses the former challenge of identifying a trajectory, or a set of, for a particular task. Contributions on the challenge of generalisable behaviour to reduce the storage footprint are described in Part III and Part IV. Nevertheless, in the development of our framework, we employ off-the-shelf machine learning tools to mitigate the challenges of encoding behaviour and storage needs, and to be able to demonstrate the applicability of our work on real-world problems. In particular, we employed dynamic movement primitives (DMPs) [70, 68] to encode trajectories and retrieve behavioural abstractions as tractable kinematic models. An overview of DMPs is provided in Chapter 4, and an in-detail technical discussion in Chapter 8.

An advantage of using relational databases for problem-relevant behavioural look-up is the ability to roughly discern whether the framework contains any suitable initialisation for the given task. For example, in Chapter 4, where a particular behaviour is abstracted as one DMP (or trajectory), the relational database provides a one-to-one task-DMP relation. Thus, it is trivial to identify when no relevant abstraction is available. In Chapter 5, we considered behavioural abstractions resulting in multiple possible outcomes, either for the existence of multiple goals, e.g., many possible grasping points, or for the availability of multiple samples of the same behaviour, e.g., a library of DMPs, or a library of raw trajectories. We proposed a fast-forward self-assessment strategy to rank the suitability of the behavioural samples. In either work, when no relevant abstraction is available, an option is to plan from scratch (as schematised in Figure 1.3) to prevent the local minima induced by the exploitation of an unsuitable behavioural initialisation. Alternatively, if multi-threading is an option, the most n suitable behavioural samples could be exploited, altogether with a mixture of exploratory and exploitative approaches. We adopt and further discuss such an idea in Chapter 9.

Learning and Composing Primitive Skills for Dual-arm Manipulation

In this chapter, we introduce the concept of a library of motions to bootstrap robotic behaviour. We leverage human expertise in conducting daily tasks to supply the library with relevant experiences. In particular, we use kinaesthetic teaching to guide the robot through the tasks, and encode the observed motions as dynamic movement primitives (DMPs) (also referred to as primitive skills). Novel task instances are solved by generalising and fusing primitive skills observed a priori. The proposed concept of a library of experiences is evaluated on the iCub humanoid robot and several synthetic experiments, by conducting a dual-arm pick-and-place task of a parcel in the presence of point-mass obstacles. Results suggest the suitability of the method to bootstrap robotic behaviour from a library of motions.

All proposed work is described in detail in the following published conference paper:

Learning and Composing Primitive Skills
for Dual-Arm Manipulation

Èric Pairet[1,2(✉)] , Paola Ardón[1,2] , Michael Mistry[1] , and Yvan Petillot[2]

[1] Institute of Perception, Action and Behaviour, University of Edinburgh,
Edinburgh, UK
{eric.pairet,paola.ardon}@ed.ac.uk, mmistry@inf.ed.ac.uk
[2] Engineering and Physical Sciences, Heriot-Watt University, Edinburgh, UK
y.r.petillot@hw.ac.uk

Abstract. In an attempt to confer robots with complex manipulation capabilities, dual-arm anthropomorphic systems have become an important research topic in the robotics community. Most approaches in the literature rely upon a great understanding of the dynamics underlying the system's behaviour and yet offer limited autonomous generalisation capabilities. To address these limitations, this work proposes a modelisation for dual-arm manipulators based on dynamic movement primitives laying in two orthogonal spaces. The modularity and learning capabilities of this model are leveraged to formulate a novel end-to-end learning-based framework which (i) learns a library of primitive skills from human demonstrations, and (ii) composes such knowledge simultaneously and sequentially to confront novel scenarios. The feasibility of the proposal is evaluated by teaching the iCub humanoid the basic skills to succeed on simulated dual-arm pick-and-place tasks. The results suggest the learning and generalisation capabilities of the proposed framework extend to autonomously conduct undemonstrated dual-arm manipulation tasks.

Keywords: Learning from demonstration · Humanoid robots ·
Model learning for control · Dual arm manipulation ·
Autonomous agents

1 Introduction

Complex manipulation tasks can be achieved by endowing anthropomorphic robots with dual-arm manipulation capabilities. Bi-manual arrangements extend the systems competences to efficiently perform tasks involving large objects or assembling multi-component elements without external assistance. These systems not only deal with the challenges of single-arm manipulators, such as trajectory planning and environmental interaction, but also require an accurate synchronisation between arms to avoid breaking or exposing the handled object to stress.

Traditional approaches have addressed the aforementioned challenges by means of control and planning-based methods [18]. These methods depend upon

66 È. Pairet et al.
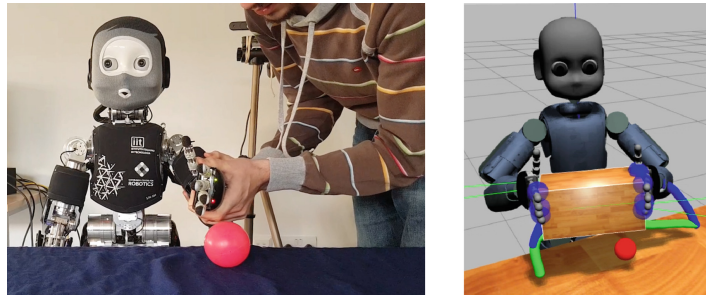
Fig. 1. iCub humanoid learning to contour an obstacle through kinaesthetic guiding (left), and composing multiple skills to conduct a dual-arm pick-and-place task (right).

an excellent understanding of the exact model underlying the system's and task's dynamics, which are commonly approximated to make the calculations computationally tractable [13]. On top of that, some of these methods lack scalability and generalisation capabilities, involving hand-defining all possible scenarios and actions [3,5]. All these issues have motivated the use of more natural techniques for robot programming, such as (LbD), in which a human movement is recorded to be later reproduced by a robot.

Despite the encouraging possibilities offered by adopting human knowledge for robot control, teaching complex systems, such as dual-arm manipulators, to respond and adapt to a broad case of scenarios remains an open challenge. Particularly, it is expected from a dual-arm system to generalise the provided demonstrations to confront novel scenarios in (a) the task space to deal with the changing requirements about trajectory planning and environmental interaction, and (b) the relative space to ensure the essential synchronisation between arms [12]. However, current learning-based architectures in the literature pursuing autonomy and robustness against the dynamic and unpredictable real-world environments are limited to single-arm arrangements [4,15,17]. Contrarily, learning-based frameworks for dual-arm robots do not generalise to undemonstrated states, thus being limited to highly controlled scenarios [7,19,21].

This paper presents a novel learning-based framework which endows a dual-arm system with a real-time and generalisable method for manipulation in undemonstrated environments (see Fig. 1). The framework models a dual-arm manipulator with a set of dynamic movement primitives laying in two orthogonal spaces to tackle the task's requirements separately from the synchronisation constraints. The modularity of the DMPs is leveraged to (i) create a library of primitive skills from human demonstrations, and (ii) exploit primitive skills simultaneously and sequentially to create complex behaviours. The potential of the proposal is demonstrated in simulation after recording skills with the iCub humanoid through kinaesthetic guiding. The results suggest the proposal's suitability to endow a dual-arm robot with the necessary learning and generalisation capabilities to autonomously address novel manipulation tasks.

2   Dual-Arm System Modelisation

This paper pursues an end-to-end learning-based framework which endows a dual-arm system with enhanced generalisation capabilities, meets the synchronisation constraints, and is easily programmable by non-robotics-experts. This work addresses all these requirements by means of learnable and composable primitive skills represented as dynamic movement primitives (DMPs) [9]. This section firstly overviews DMPs and its use in the literature. It then introduces the proposed typology of actions in a dual-arm system, which allows leveraging the strengths of a DMP-based modelisation in the dual-arm context.

2.1   Dynamic Movement Primitives

DMPs are a versatile tool for modelling and learning complex motions. They describe the dynamics of a primitive skill as a spring-damper system under the effect of a virtual external force called coupling term. This coupling term allows for learning and reproducing any dynamical behaviour, i.e. primitive skill. Importantly, (a) coupling terms can be learnt from human demonstrations, (b) they can be efficiently learned and generated, (c) a unique demonstration is already generalisable, (d) convergence to the goal is guaranteed, and (e) their representation is translation and time-invariant. Because of all these properties, DMPs are adapted to constitute the fundamental building blocks of this work. Next follows an introduction about DMPs and their usage to encode positional and orientational dynamics, and an overview of some coupling terms in the literature.

Positional Dynamics. Let the positional state of a one-degree of freedom (DoF) system be defined by its position, linear velocity and acceleration. Then, the system's state transition is defined with non-linear differential equations as:

$$\tau \dot{z} = \alpha_x(\beta_x(g_x - x) - z) + f_x(\cdot), \tag{1}$$

$$\tau \dot{x} = z, \tag{2}$$

where τ is a scaling factor for time, x is the system's position, z and $\dot{z}$ respectively are the scaled velocity and acceleration, α_x and β_x are constants defining the positional system's dynamics, g_x is the model's attractor, and $f_x(\cdot)$ is the coupling term. The coupling term applying at multiple DoFs at once is defined as $\mathbf{f}_x(\cdot)$. The system will converge to g_x with critically damped dynamics and null velocity when $\tau > 0$, $\alpha_x > 0$, $\beta_x > 0$ and $\beta_x = \alpha_x/4$ [9].

Orientational Dynamics. A possible representation of orientations is the unit quaternion $\mathbf{q} \in \mathbb{R}^4 = \mathbb{S}^3$ [20]. They encode orientations of a system as a whole, thus ensuring the stability of the orientational dynamics integration. Let the current orientational state of a system be defined by its orientation, angular

velocity and acceleration. Then, the orientational state transition is described by the following non-linear differential equations:

$$\tau\dot{\boldsymbol{\eta}} = \alpha_q(\beta_q\, 2\log(\mathbf{g}_q * \bar{\mathbf{q}}) - \boldsymbol{\eta}) + \mathbf{f}_q(\cdot), \tag{3}$$

$$\tau\dot{\mathbf{q}} = \frac{1}{2}\boldsymbol{\eta} * \mathbf{q}, \tag{4}$$

where $\mathbf{q}$ is the system's orientation, $\boldsymbol{\eta}$ and $\dot{\boldsymbol{\eta}}$ respectively are the scaled angular velocity and acceleration, α_q and β_q are constants defining the system's dynamics, $\mathbf{g}_q \in \mathbb{S}^3$ is the model's attractor, and $\mathbf{f}_q(\cdot) \in \mathbb{R}^4$ is the coupling term. The operators $\log(\cdot)$, $*$, and $\bar{\mathbf{q}}$ denote the logarithm, multiplication and conjugate operations for quaternions, respectively.

Coupling Terms. Coupling terms describe the system's behaviour, thus being useful to learn and retrieve any primitive skill. They are commonly used to encode the positional [9] and orientational [20] dynamics of a motion. Coupling terms are modelled in each dimension as a weighted linear combination of non-linear radial basis functions (RBFs) distributed along the trajectory. Thus, learning a certain movement relies on finding the weights of the RBFs which closely reproduce a demonstrated skill.

More complex behaviours may be achieved by exploiting an additional coupling term simultaneously with the motion-encoding one. This approach has been used to avoid joint limits and constraining the robot's workspace via repulsive forces pushing the system away from these limits [6]. Coupling terms have also been leveraged for obstacle avoidance with an analytic biologically-inspired approach describing how humans steer around obstacles [8,17]. Another use is for environmental and self-interaction purposes by means of a controller tracking a desired force profile [7]. To the best of the authors' knowledge, the practice of using coupling terms simultaneously has been limited to two primitive skills acting on the same frame or space [15]. Contrarily, this work further exploits the DMP modularity to describe a dual-arm system in two orthogonal spaces with the purpose of facing complex scenarios by composing multiple coupling terms.

2.2   Dual-Arm Primitive Skills Taxonomy

Skills for single-arm manipulation have been well analysed in the robotics community. While some of this knowledge can be extrapolated for a dual-arm manipulator as a whole, their complexity resides in the arms interaction. In the context of manipulation via a dual-arm system, a possible classification of any primitive skill falls into two groups: (a) absolute skills, which imply a change of configuration of the manipulated object in the Cartesian or absolute space $\mathcal{S}_a$, e.g. move or turn an object in a particular manner, and (b) relative skills, which exert an action on the manipulated object in the object or relative space $\mathcal{S}_r$, e.g. opening a bottle's screw cap, or hold a parcel employing force contact.

Each type of primitive skill uniquely produces movement in its space since they lay in orthogonal spaces such that $\mathcal{S}_a \perp \mathcal{S}_r$. It is natural to expect from

a dual-arm system to simultaneously carry out, at least, one absolute and one relative skill to accomplish a task. Let us analyse the task of moving a bottle to a particular position while opening its screw cap. Both end-effectors synchronously move to reach the desired configuration (absolute skill). At the same time, the left end-effector is constrained to hold the bottle upright (relative skill), while the right end-effector unscrews the cap (relative skill).

2.3 Dual-Arm DMP-Based Modelisation

Given the variety of primitive skills that a dual-arm system can execute, this work seeks to model the robotic platform in a generalisable yet modular fashion, which accounts for both absolute and relative skills. To this aim, let us consider the closed kinematic chain depicted in Fig. 2 operating in a three-dimensional (3D) workspace $\mathcal{W} = \mathbb{R}^3 \times \mathrm{SO}(3)$. Each arm i, where $i = \{L, R\}$, interacts with the same object $\mathcal{O}$. In this context, the absolute skill explains the movement of the object $\mathcal{O}$ in the workspace $\mathcal{W} = \mathcal{S}_a$, while the relative skill describes the actions of each end-effector i in $\mathcal{S}_r$, i.e. with respect to the object's reference frame $\{\mathcal{O}\}$. Note that $\{\mathcal{O}\}$ is the centre of the closed-chain dual-arm system.

The state of the closed-chain dual-arm system in the workspace can be described by the position/orientation, linear/angular velocities and accelerations of $\{\mathcal{O}\}$ in $\mathcal{S}_a$. As introduced previously, the system's state transition is subjected to its modelled dynamics. Figure 2 illustrates the proposed modelisation of the system's dynamics in $\mathcal{S}_a$ as a set of DMPs acting between the objects's frame $\{\mathcal{O}\}$ and its goal configuration $\mathbf{g}_o$, which accounts for a desired goal position $\mathbf{g}_{o_x} \in \mathbb{R}^3$ and orientation $\mathbf{g}_{o_q} \in \mathbb{R}^4$. Therefore, three positional DMPs as in (1)–(2) and one orientational DMP as in (3)–(4) are required to encode the system's dynamics in the absolute space $\mathcal{S}_a = \mathbb{R}^3 \times \mathrm{SO}(3)$.

In the relative space $\mathcal{S}_r$, the dynamics of each end-effector are modelled as DMPs referenced to the objects's frame $\{\mathcal{O}\}$. Since $\mathcal{S}_r = \mathbb{R}^3 \times \mathrm{SO}(3)$, each end-
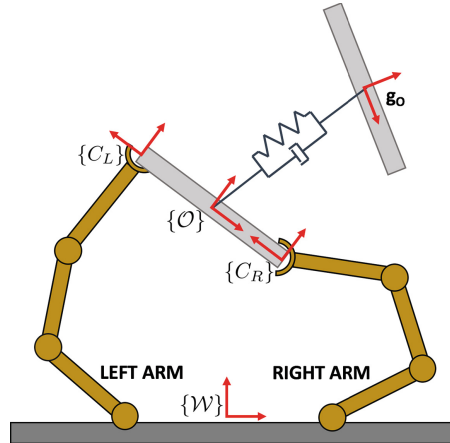
Fig. 2. DMP-based modelisation of a closed-chain dual-arm system in the absolute and relative spaces. This model is extended to deal with rotational dynamics.

70   È. Pairet et al.

effector dynamic's in the relative frame is described by three positional DMPs as in (1)–(2) and one orientational DMP as in (3)–(4).

Any action referenced to the object's frame can be projected to the end-effectors using the grasping geometry **G** of the manipulated object. This allows computing the required end-effector control commands to achieve a particular absolute task. A detailed explanation of this transformation can be found in [12].

3 Learning-Based Dual-Arm Manipulation

To endow a dual-arm manipulator with autonomy and robustness in novel scenarios while being easily programmable and customisable by non-robotics-experts, this work has decomposed and modelled the system's dynamics and synchronisation constraints as primitive skills lying in the system's absolute and relative space. Leveraging the formulated modelisation, this work proposes the framework schematised in Fig. 3 which creates and manages a library of primitive skills. The framework has two components: (i) a learning module that learns a set of primitive skills from human demonstrations, and (ii) a manager module that combines simultaneously and sequentially these primitives to address a wide range of complex tasks in unfamiliar environments.

3.1 Library Generation

A primitive skill is represented by its coupling term and frame of reference, i.e. either absolute or relative. Learning coupling terms only requires a human demonstrator teaching the characteristic skill. As previously introduced, different coupling terms might be better formulated with different mathematical representations, e.g. a weighted combination of non-linear RBFs to encode the dynamics of a task, an analytical obstacle avoidance expression, or among others, a force profile to control the environmental interaction.
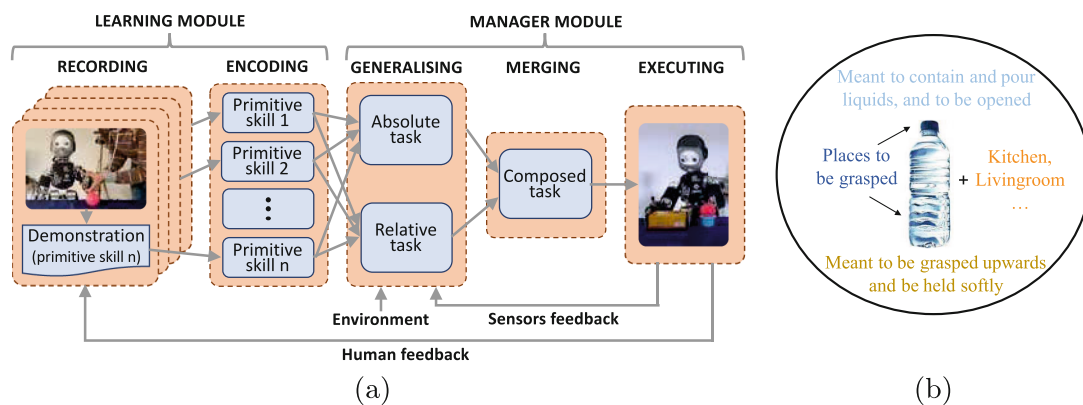
Fig. 3. Scheme of the proposed framework. (a) Learning: a library of primitive skills is learnt from human demonstrations. Manager: the primitives are combined simultaneously and sequentially to confront novel environments. (b) The required primitive skills are selected according to the affordance elements of the dual-arm task.

The modularity offered by the proposed DMP-based formulation and its use in two different spaces tackles the hindrance and ambiguity arising when demonstrating all features of a dual-arm task in an all-at-once fashion. This means that instead of learning a task as a whole, the framework harvests a collection of primitive skills. Creating a repertoire of skills referred to as a library, allows the demonstrator to teach in a one-at-a-time fashion, i.e. to focus on one feature of the demonstration at a time [4]. Moreover, this modular library can be employed for movement recognition purposes, where a demonstrated skill can be compared against the existing ones in the library. If the observed behaviour does not match any existing primitive, it is identified as a new skill and can be added to the framework's library [10].

3.2 Attaching Semantics

The framework needs additional information to successfully conduct a dual-arm manipulation task. Let us consider the robotic task of opening a bottle's screw cap, where the system needs to select a proper sequence of primitive skills in order to succeed (see Fig. 3(b)). This is first a grasping, where each end-effector holds a different component of the bottle, then a synchronous turning referenced in the system's relative space and finally, a placing and releasing primitives. Therefore, in order to ease this action selection, it is essential to attach a semantic description to each primitive skill.

Semantic labels bridge the gap between the low-level continuous representation of primitives and the high-level description of actions and their influence on objects. An approach to tackle the object affordances challenge consists in combining features from the object and their surroundings to infer on a suitable grasp-action based on their purpose of use [1,2]. The combination of such elements builds the relationship between context, actions and effects that provide a cognitive reasoning of an object affordance.

3.3 Library Management

Each coupling term stored in the framework's library represents a particular absolute or relative primitive skill. Reproducing a skill consists in using its coupling term as $\mathbf{f}_x(\cdot)$ or $\mathbf{f}_q(\cdot)$ in (1)–(4). This computation retrieves the skill's required accelerations, which can be integrated over time to obtain the skill's velocities $\dot{\mathbf{y}}_o$ for an absolute primitive or $\dot{\mathbf{y}}_{C_i}$ for the end-effector i relative primitive.

The individual retrieval of primitives already accounts for the inner DMP generalisation capabilities, such as different start and goal configurations, as well as obstacle locations. However, these primitive skills need to be combined to generate more complex movements, such as a pick-and-place task of a bottle accounting for the presence of unexpected obstacles (absolute space), while opening the bottle's screw cap considering the exerted force (relative space).

72   È. Pairet et al.

The presented framework addresses this prerequisite by simultaneously combining different absolute and relative skills as:

$$\begin{bmatrix} \dot{\mathbf{y}}_L \\ \dot{\mathbf{y}}_R \end{bmatrix} = \mathbf{G}^T \sum_{j=1}^{J} w_j \, \dot{\mathbf{y}}_{o_j} + \sum_{k=1}^{K} w_k \begin{bmatrix} \dot{\mathbf{y}}_{C_{L,k}} \\ \dot{\mathbf{y}}_{C_{R,k}} \end{bmatrix}, \tag{5}$$

where $\dot{\mathbf{y}}_i \in \mathbb{R}^6$ describes the linear and angular velocity commands of the $i = \{L,\ R\}$ end-effector satisfying the set of activated primitive skills, $\mathbf{G} \in \mathbb{R}^{6 \times 12}$ is the global grasp map of the two end-effectors grasp matrices as described in [13], and $\dot{\mathbf{y}}_{o_j} \in \mathbb{R}^6$ and $\dot{\mathbf{y}}_{C_{i,k}} \in \mathbb{R}^6$ are the velocities of the $j \in [1,\ J]$ absolute and $k \in [1,\ K]$ relative primitive skill stored in the library. Absolute and relative skill selection is conducted with the weights w_j and w_k, respectively.

The resulting framework does not only combines skills simultaneously, but also sequentially. This allows the execution of a complex task composed of a sequence of primitives. To do so, a primitive skill is executed by initialising it with the full state (pose, velocities and accelerations) of its predecessor primitive skill. Such an initialisation avoids abrupt jumps in the system's state.

4   Results and Evaluation

The proposed framework has been evaluated on the iCub humanoid robot. Particularly, the real platform has been used to load the framework's library with a set of primitive skills learnt from human demonstrations. These skills have been employed in simulation to conduct dual-arm pick-and-place tasks of a parcel in novel scenarios, demonstrating the proposal's potential for humanoid robots.

4.1   Experimental Platform

iCub is an open source humanoid robot with 53 DoFs [11] (see Fig. 4(b)). The most relevant ones in this work are the three-DoFs on the torso, the two seven-DoFs arms equipped with a torque sensor on the shoulder, and the two nine-DoFs anthropomorphic hands with tactile sensors in the fingertips and palm.

iCub operates under YARP. The deployment of the proposed framework on the iCub platform is schematised in Fig. 4. Mainly, four big functional modules can be distinguished: (i) the proposed framework described in this paper (blue blocks), (ii) the real/simulated platform with its visual perception, joint sensors and actuators (magenta blocks), (iii) the end-effectors control via the built-in YARP Cartesian controller [16] and an ad-hoc external torso controller (green blocks), and (iv) the HRI interface to parameterise the desired start and goal configurations for the task, and retrieve the robot's status (red blocks).

4.2   Learning Primitive Skills from Demonstration

For the system to succeed on the dual-arm pick-and-place of a parcel task in novel environments, the framework's library needs to be loaded with the absolute
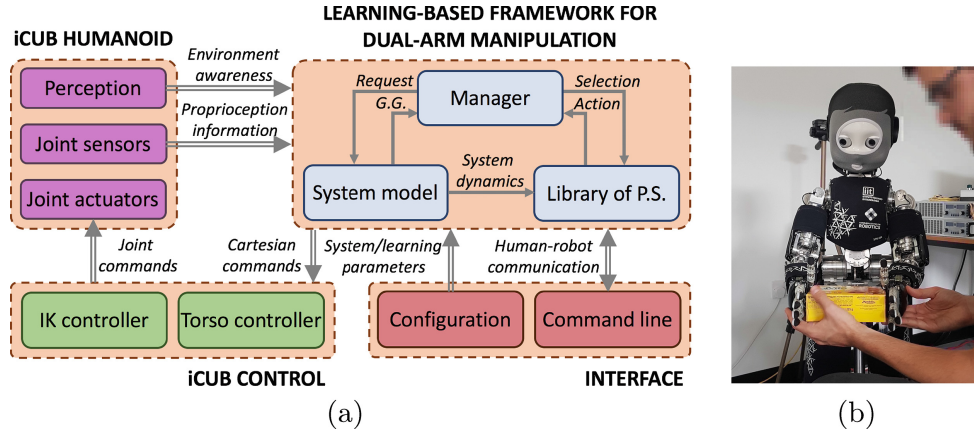
Fig. 4. (a) Layout of the framework deployment on the iCub robot. Note: grasping geometry (GG), primitive skill (PS). (b) iCub humanoid being taught grasp maintenance through kineasthetic guiding. (Color figure online)

primitive skills of (i) pick-and-place dynamics on a horizontal surface, (ii) rotational motion around the z-axis, and (iii) obstacle avoidance. Moreover, since the parcel has to be grasped by lateral contact of both end-effectors, the library also requires a relative skill to ensure grasp maintenance, i.e. prevention of contact separation. All these primitive skills have been demonstrated via kineasthetic guiding on the real iCub humanoid robot. To this aim, all joints have been set in gravity compensation, allowing the demonstrator to physically manoeuvre the robot through each primitive. Figure 4(b) depicts the kineasthetic teaching of obstacle avoidance and grasp maintenance primitives.

During the demonstrations, proprioception information is retrieved via YARP ports to learn the coupling terms $\mathbf{f}_x(\cdot)$ and $\mathbf{f}_q(\cdot)$ in (1)–(4) characterising the different skills. For the pick-and-place and rotational dynamics, the coupling terms are encoded as a weighted linear combination of non-linear RBFs distributed along the trajectory as in [9]. The obstacle avoidance is learnt by finding the best-fitting parameters of the biologically-inspired formulation as in [17]. Finally, the grasp maintenance skill is learnt by setting the parcel's grasping geometry as a pose tracking reference as in [7].

4.3 Experiments on Simulated iCub Humanoid

The evaluation of the framework on the pick-and-place setup has been conducted on a simulated iCub robotic platform. Particularly, the four primitive skills previously learnt and loaded in the framework's library are simultaneously and sequentially combined to conduct three consecutive dual-arm pick-and-place task in novel environments (see Fig. 5).

Given an initial random configuration laying on the table and within iCub's workspace (see Fig. 5(a)), the first action consists of grasping the parcel. This is achieved by retrieving the parcel's configuration, then use the learnt parcel's geometry to compute the grasping points, and finally approach them laterally

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

(e)   (f)   (g)   (h)
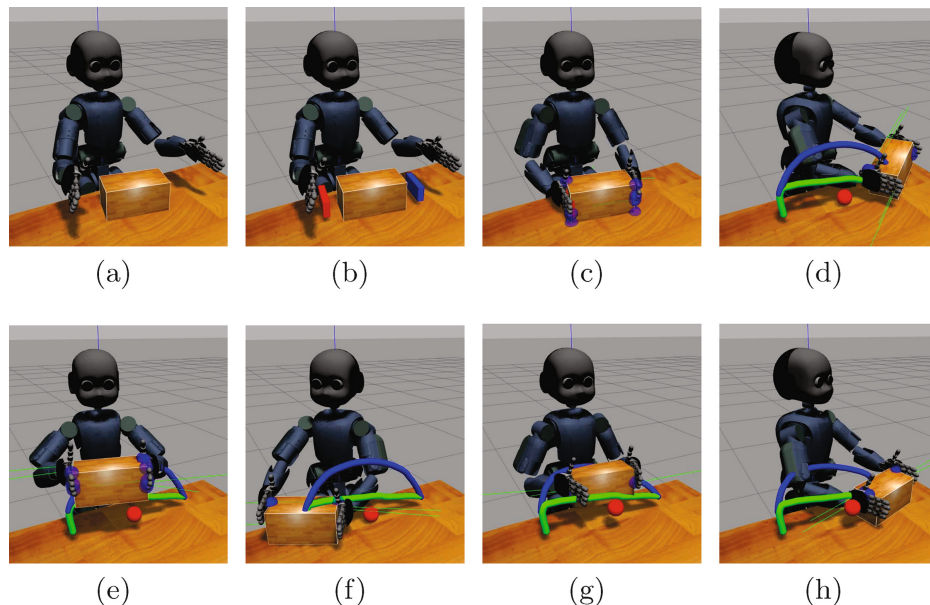
Fig. 5. iCub humanoid succeeding in novel dual-arm pick-and-place tasks by simultaneously and sequentially combining primitive skills. Demonstrated pick-and-place (green trajectory). Framework's response (blue trajectory). Obstacle (red sphere). (a) Parcel initial state. (b)–(c) Grasping parcel laterally. (d)–(f) and (f)–(h) Pick-and-place execution with different start, goal and obstacle configurations. (Color figure online)

via the middle-setpoints displayed as red and blue prisms for the right and left end-effector, respectively (see Fig. 5(a)–(c)). From this stage on, the grasp maintenance skill ensures that both end-effectors are in flat contact with the box to avoid undesired slippage.

The following three consecutive movements require picking-and-placing the parcel between different configurations laying on the central, right and left side of iCub's workspace. The former pick-and-place does not require avoiding any obstacle, thus the built-in DMPs generalisation capabilities are sufficient to address this task (see Fig. 5(c)–(d)). However, the two latter pick-and-place tasks involve adapting the learnt dynamics to address novel scenarios. When the obstacle (red sphere) is collinear with the start and goal positions, i.e. below the demonstrated task (green trajectory), the iCub humanoid circumnavigates the obstacle from the top (see Fig. 5(d)–(f)). Instead, for an obstacle located forward the demonstration, the framework guides the system through a collision-free trajectory near iCub's chest (see Fig. 5(g)–(h)).

The experimental evaluation conducted with the simulated iCub humanoid robot has demonstrated various of the aforementioned framework's features. Having a repertoire of primitive skills available in the framework's library allows exploiting them simultaneously and sequentially to confront complex tasks in novel scenarios. The reported case is one of the 16 successful experiments out of a total of 20 trials. In all cases, the robot had to accomplish the three consecutive dual-arm pick-and-place tasks with different start and goal locations,

while avoiding novel obstacles and ensuring grasp maintenance. Failure in any of these tasks made the trial unsuccessful. Interestingly, in the four failed trials one of iCub's forearms collided with the obstacle. This is because the biologically-inspired obstacle avoidance formulation only considers the carried object and should be extended to the object-arm space. The flexibility of the proposed framework could be leveraged to integrate in its library a potential field-inspired approach for obstacle avoidance which also checks for link collisions [14].

5 Final Remarks and Future Work

This work has presented a novel end-to-end learning-based framework which endows a dual-arm manipulator with real-time and generalisable manipulation capabilities. The framework is built upon the proposed extension of the DMP-based modelisation for dual-arm systems, which considers two different frames to reference the movement generation, force interaction and constraints requirements. Based on this arrangement, the proposed framework is twofold: (i) learns from human demonstrations to create a library of primitive skills, and (ii) combines such knowledge simultaneously and sequentially to confront novel scenarios.

The suitability of the proposed approach has been demonstrated in a dual-arm pick-and-place setting, where the iCub humanoid first learnt a repertoire of primitive skills from human demonstrations and then composed such knowledge to successfully generalise to novel scenarios. The framework is not restricted to the presented experimental evaluation nor platform. Any system capable of learning from demonstrations can benefit from this work. Moreover, the framework's modularity allows loading to its library any primitive skill that might be required for dual-arm manipulation purposes.

Future work will significantly extend the library of primitive skills such that more challenging dual-arm manipulation behaviours can be addressed within the framework. In this regard, imminent efforts will focus on learning force-dependant primitive skills or other actions requiring complex synchronisation between end-effectors, such as the opening of a bottle's screw cap or succeeding in the peg-in-a-hole tasks.

References

1. Ardón, P., Pairet, È., Petrick, R., Ramamoorthy, S., Lohan, K.S.: Reasoning on grasp-action affordances. In: Konstantinova, J., et al. (eds.) TAROS 2019, LNAI, vol. 11529, pp. 3–15. Springer, Heidelberg (2019)
2. Ardón, P., Pairet, È., Ramamoorthy, S., Lohan, K.S.: Towards robust grasps: using the environment semantics for robotic object affordances. In: Proceedings of the AAAI Fall Symposium on Reasoning and Learning in Real-World Systems for Long-Term Autonomy, pp. 5–12. AAAI Press (2018)

76 È. Pairet et al.

3. Argall, B.D., Chernova, S., Veloso, M., Browning, B.: A survey of robot learning from demonstration. Robot. Auton. Syst. **57**(5), 469–483 (2009)
4. Bajcsy, A., Losey, D.P., O'Malley, M.K., Dragan, A.D.: Learning from physical human corrections, one feature at a time. In: Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction, pp. 141–149. ACM (2018)
5. Billard, A., Calinon, S., Dillmann, R., Schaal, S.: Robot programming by demonstration. In: Siciliano, B., Khatib, O. (eds.) Springer Handbook of Robotics, pp. 1371–1394. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-30301-5_60
6. Gams, A., Ijspeert, A.J., Schaal, S., Lenarčič, J.: On-line learning and modulation of periodic movements with nonlinear dynamical systems. Auton. Robots **27**(1), 3–23 (2009)
7. Gams, A., Nemec, B., Ijspeert, A.J., Ude, A.: Coupling movement primitives: Interaction with the environment and bimanual tasks. IEEE Trans. Robot. **30**(4), 816–830 (2014)
8. Hoffmann, H., Pastor, P., Park, D.H., Schaal, S.: Biologically-inspired dynamical systems for movement generation: automatic real-time goal adaptation and obstacle avoidance. In: 2009 IEEE International Conference on Robotics and Automation, ICRA 2009, pp. 2587–2592. IEEE (2009)
9. Ijspeert, A.J., Nakanishi, J., Hoffmann, H., Pastor, P., Schaal, S.: Dynamical movement primitives: learning attractor models for motor behaviors. Neural Comput. **25**(2), 328–373 (2013)
10. Ijspeert, A.J., Nakanishi, J., Schaal, S.: Movement imitation with nonlinear dynamical systems in humanoid robots. In: 2002 Proceedings of the IEEE International Conference on Robotics and Automation, ICRA 2002, vol. 2, pp. 1398–1403. IEEE (2002)
11. Metta, G., Sandini, G., Vernon, D., Natale, L., Nori, F.: The iCub humanoid robot: an open platform for research in embodied cognition. In: Proceedings of the 8th Workshop on Performance Metrics for Intelligent Systems, pp. 50–56. ACM (2008)
12. Pairet, È., Ardón, P., Broz, F., Mistry, M., Petillot, Y.: Learning and generalisation of primitives skills towards robust dual-arm manipulation. In: Proceedings of the AAAI Fall Symposium on Reasoning and Learning in Real-World Systems for Long-Term Autonomy, pp. 62–69. AAAI Press (2018)
13. Pairet, È., Hernández, J.D., Lahijanian, M., Carreras, M.: Uncertainty-based online mapping and motion planning for marine robotics guidance. In: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 2367–2374. IEEE (2018)
14. Park, D.H., Hoffmann, H., Pastor, P., Schaal, S.: Movement reproduction and obstacle avoidance with dynamic movement primitives and potential fields. In: 2008 8th IEEE-RAS International Conference on Humanoid Robots, Humanoids 2008, pp. 91–98. IEEE (2008)
15. Pastor, P., Hoffmann, H., Asfour, T., Schaal, S.: Learning and generalization of motor skills by learning from demonstration. In: 2009 IEEE International Conference on Robotics and Automation, ICRA 2009, pp. 763–768. IEEE (2009)
16. Pattacini, U., Nori, F., Natale, L., Metta, G., Sandini, G.: An experimental evaluation of a novel minimum-jerk cartesian controller for humanoid robots. In: 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 1668–1674. IEEE (2010)
17. Rai, A., Meier, F., Ijspeert, A., Schaal, S.: Learning coupling terms for obstacle avoidance. In: 2014 14th IEEE-RAS International Conference on Humanoid Robots (Humanoids), pp. 512–518. IEEE (2014)

Learning and Composing Primitive Skills for Dual-Arm Manipulation   77

18. Smith, C., et al.: Dual arm manipulation: a survey. Robot. Auton. Syst. **60**(10), 1340–1353 (2012)
19. Topp, E.A.: Knowledge for synchronized dual-arm robot programming. In: 2017 AAAI Fall Symposium Series. AAAI Press (2017)
20. Ude, A., Nemec, B., Petrić, T., Morimoto, J.: Orientation in cartesian space dynamic movement primitives. In: 2014 IEEE International Conference on Robotics and Automation (ICRA), pp. 2997–3004. IEEE (2014)
21. Zöllner, R., Asfour, T., Dillmann, R.: Programming by demonstration: dual-arm manipulation tasks for humanoid robots. In: IROS, pp. 479–484 (2004)

Self-Assessment of Grasp Affordance Transfer

In this chapter, we tackle the challenge of end-to-end autonomous robotic manipulation. To that aim, we leverage the concept of a library of motions within our affordance framework presented in [8, 6, 7]; robotic affordances allow an autonomous agent to estimate the most suitable grasp and action, and expected effect subject to a given task. We cross-correlate physical features on objects, human-demonstrated motions and grasp preferences to harvest an understanding of affordance relations, and exploit it in novel task instances. As the affordance relations provide multiple hypothesis on suitable grasp-action pairs, we exploit the rapid roll-out of the DMP-encoded experiences to forward simulate and evaluate the outcome of executing the affordance task. The hypotheses are ranked by performance success with a heuristic confidence function and used to build a library of affordance task experiences. Experimental evaluation shows that our method exhibits a significant performance improvement against other approaches for end-to-end manipulation. Experiments on a PR2 robotic platform demonstrate our method's highly reliable deployability to conduct real-world manipulation tasks autonomously.

All proposed work is described in detail in the following published conference paper:

Self-Assessment of Grasp Affordance Transfer

Paola Ardón*, Èric Pairet*, Yvan Petillot, Ronald P. A. Petrick,
Subramanian Ramamoorthy, and Katrin S. Lohan

Abstract— **Reasoning about object grasp affordances allows an autonomous agent to estimate the most suitable grasp to execute a task. While current approaches for estimating grasp affordances are effective, their prediction is driven by hypotheses on visual features rather than an indicator of a proposal's suitability for an affordance task. Consequently, these works cannot guarantee any level of performance when executing a task and, in fact, not even ensure successful task completion. In this work, we present a pipeline for self-assessment of grasp affordance transfer (SAGAT) based on prior experiences. We visually detect a grasp affordance region to extract multiple grasp affordance configuration candidates. Using these candidates, we forward simulate the outcome of executing the affordance task to analyse the relation between task outcome and grasp candidates. The relations are ranked by performance success with a heuristic confidence function and used to build a library of affordance task experiences. The library is later queried to perform one-shot transfer estimation of the best grasp configuration on new objects. Experimental evaluation shows that our method exhibits a significant performance improvement up to 11.7% against current state-of-the-art methods on grasp affordance detection. Experiments on a PR2 robotic platform demonstrate our method's highly reliable deployability to deal with real-world task affordance problems.**

I. INTRODUCTION

Affordances have attained new relevance in robotics over the last decade [1], [2]. Affordance refers to the possibility of performing different tasks with an object [3]. As an example, grasping a pair of scissors from the tip affords the task handing over, but not a cutting task. Analogously, not all the regions on a mug's handle comfortably afford to pour liquid from it. Current grasp affordance solutions successfully detect the parts of an object that afford different tasks [4]–[9]. This allows agents to contextualise the grasp according to the objective task and also, to novel object instances. Nonetheless, these approaches lack an insight into the level of suitability that the grasp offers to accomplish the task. As a consequence, current literature on grasp affordance cannot guarantee any level of performance when executing the task and, in fact, not even a successful task completion.

On the grounds of the limitations mentioned above, a system should consider the expected task performance when deciding a grasp affordance. However, this is a challenging problem, given that the grasp and the task performance are codefining and conditional on each other [10]. Recent research in robot affordances proposes to learn this relation

*These authors contributed equally to this work.

The authors are with the Edinburgh Centre for Robotics at the University of Edinburgh and Heriot-Watt University, Edinburgh, UK. This research is supported by the Scottish Informatics and Computer Science Alliance (SICSA), EPSRC ORCA Hub (EP/R026173/1) and consortium partners. {paola.ardon,eric.pairet}@ed.ac.uk
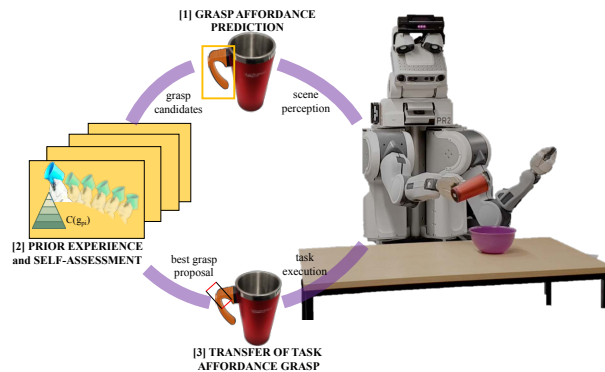
Fig. 1: PR2 self-assessing a pouring affordance task. The system first predicts the object's grasp affordances. Then, based on prior affordance task experiences and a heuristic confidence metric, it self-assesses the new object's grasp configuration that is most likely to succeed at pouring.

via trial and error of the task [11]–[13]. Nevertheless, given the extensive amount of required data, the method can solely learn a single task at a time and perform on known scenarios. In contrast, an autonomous agent is expected to be capable of dealing with multiple task affordance problems even when those involve unfamiliar objects and new scenarios.

In this paper, we present a novel experience-based pipeline for self-assessment of grasp affordance transfer (SAGAT) that seeks to overcome the lack of deployment reliability of current state-of-the-art methods of grasp affordance detection. The proposed approach, depicted in Fig. 1, starts by extracting multiple grasp configuration candidates from a given grasp affordance region. The outcome of executing a task from the different grasp candidates is estimated via forward simulation. These estimates are employed to evaluate and rank the relation of task performance and grasp configuration candidates via a heuristic confidence function. Such information is stored in a library of task affordances. The library serves as a basis for one-shot transfer to identify grasp affordance configurations similar to those previously experienced, with the insight that similar regions lead to similar deployments of the task. We evaluate the method's efficacy on addressing novel task affordance problems by training on one single object and testing on multiple new ones. We observe a significant performance improvement up to 11.7% in the considered tasks when using our proposal in comparison to state-of-the-art approaches on grasp affordance detection. Experimental evaluation on a PR2 robotic platform demonstrates highly reliable deployability of the proposed method in real-world task affordance problems.
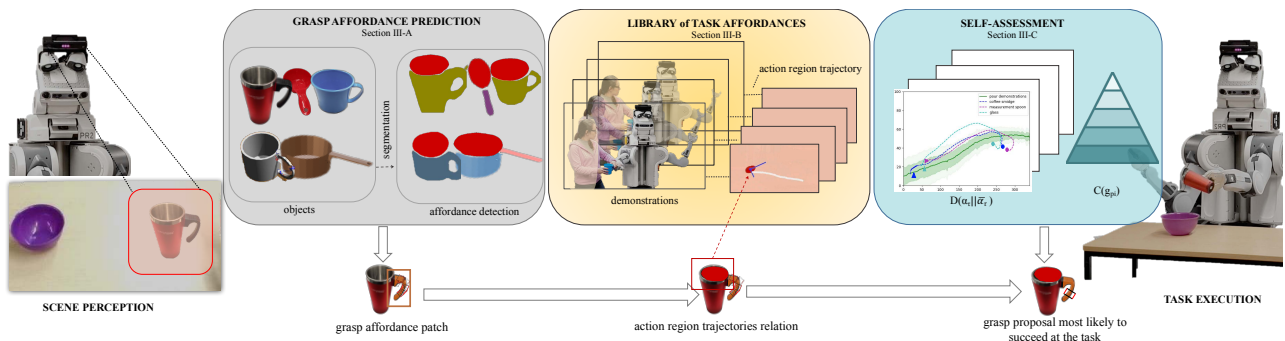
Fig. 2: Proposed framework for self-assessment of grasp affordance transfer. After predicting a grasp affordance region, the most suitable grasp is determined based on a library of prior task affordance experiences and a heuristic confidence metric.

II. RELATED WORK

Understanding grasp affordances for objects has been an active area of research for robotic manipulation tasks. Ideally, an autonomous agent should be able to identify all the tasks that an object can afford, and infer the grasp configuration that leads to a successful completion of each task. A common approach to tackle this challenge is via visual features, e.g. [4]–[7]. Methods based on visual grasp affordance detection identify candidate grasps either via deep learning architectures that detect grasp areas on an object [4]–[6], or via supervised learning techniques that obtain grasping configurations based on an object's shape [7]. While these techniques offer robust grasp candidates, they uniquely seek grasp stability. Consequently, these methods cannot guarantee any level of performance when executing a task, and in fact, not even a successful task completion. In order to, move towards reliable task deployment on autonomous agents, there is the need to bridge the gap between grasp affordance detection and task-oriented grasping.

Grasp affordances: Work on grasp affordances aims at robust interactions between objects and the autonomous agent. However, it is typically limited to a single grasp affordance detection per object, thus reducing its deployment in real-world scenarios. Some works, such as [14], focus on relating abstractions of sensory-motor processes with object structures (e.g., object-action complexes (OACs)) to extract the best grasp candidate given an object affordance. Others use purely visual input to learn affordances using deep learning [6], [8] or supervised learning techniques to relate objects and actions [9], [15]–[17]. Although these works are successful in detecting grasp affordance regions, they hypothesise suitable grasp configurations based on visual features, rather than indicators that hint such proposals suitability to accomplish an affordance task.

Task affordances: The end goal of grasping is to manipulate an object to fulfil a goal-directed task. When the grasping problem is contextualised into tasks, solely satisfying the grasp stability constraints is no longer sufficient. Nonetheless, codefining grasp configurations with task success is still an open problem. Along this line, some works focus entirely on learning tasks where the object category does not influence the outcome, such as pushing or pulling [15], [17]. Hence, reliable extraction of grasp configurations is neglected. Another approach is to learn grasp quality measures for task performance via trial and error [11]–[13]. Based on the experiences, these studies build semantic constraints to specify which object regions to hold or avoid. Nonetheless, their dependency on great amounts of prior experiences and the lack of generalisation between object instances remain to be the main hurdle of these methods.

Our work seeks to bridge the gap between grasp affordances and task performance existing in prior work. The proposed approach unifies grasp affordance reasoning and task deployment in a self-assessed system that, without the need for extensive prior experiences, is able to transfer grasp affordance configurations to novel object instances.

III. PROPOSED METHOD

An autonomous agent must be able to perform a task affordance in different scenarios. Given a particular object and task $\mathcal{T}$ to perform, the robot must select a suitable grasp affordance configuration g_p^* that allows executing the task's policy π_τ successfully. Only the correct choice of both g_p^* and π_τ leads to the robot being successful at addressing the task affordance problem. Despite the strong correlation between g_p^* and the π_τ execution performance, current approaches in the literature consider these elements to be independent. This results in grasping configurations that are not suitable for completing the task.

In this section, we introduce our approach to self-assess the selection of a suitable grasp affordance configuration according to an estimate of the task performance. Fig. 2 illustrates the proposed pipeline which (i) detects from visual information a set of grasping candidates lying in the object's grasp affordance space (Section III-A), (ii) exploits a learnt library of task affordance policies to forward simulate the outcome of executing the task from the grasping candidates (Section III-B), and then (iii) evaluates the grasp configuration candidates subject to a heuristic confidence metric (Section III-C) which allows for one-shot transfer of the grasp proposal (Section III-D). Finally, in Section III-E, we detail how theses components fit in the scheme of a robotic agent dealing with task affordance problems autonomously.

A. Prediction of Grasp Affordance Configurations

The overall goal of this work is, given an object's grasp affordance region G^*, to find a grasp configuration g_p^* that allows the robot to successfully employ an object for a particular task. In the grasp affordance literature, it is common to visually detect and segment the grasp affordance region G^* using mapping to labels [6], [8], [9]. While these methods all predict g_p^* via visual detection hypotheses, none estimate the configuration proposals based on a task performance insight. This relational gap endangers a successful task execution. Instead, an autonomous agent should be capable of discerning the most suitable grasp that benefits the execution of a task.

To bridge this gap, in our method we consider a grasp affordance region G^* in a generic form such as the bounding box provided by [9] (see Fig. 3a). We are interested in pruning this region by finding multiple grasp proposal candidates. With this aim, we use the pre-trained DeepGrasp model [5], a deep CNN that computes reliable grasp configurations on objects. The output grasp proposals g_{p_i} from DeepGrasp, which do not account for affordance relation, are shown in Fig. 3b. The pruned region (see Fig. 3c), denoted as $g_{p_i} \in G^*$, provides a set of grasp configuration candidates that accounts for both reliability and affordability.

B. Library of Task Affordances

The success of an affordance task $\mathcal{T}$ lies in executing the corresponding task policy π_τ from a suitable grasp configuration g_p^*. This is a difficult problem given that the π_τ and g_p^* are codefining [10]. Namely, the task's requirements constrain the possibly suitable grasp configurations g_p^*, at the same time that the choice of g_p^* conditions the outcome of executing the task's policy π_τ. Additionally, determining whether the execution of a task is successful requires a performance indicator. To cope with this challenge, we build on our previous work [18] to learn a library $\mathcal{L}$ of task affordances from human demonstrations. The library aims at simultaneously guiding the robot on the search of a suitable task policy π_τ while informing about its expected outcome α_τ when successful. All these elements serve as the basis of the method described in Section III-C to determine g_p^* via self-assessment of the candidates $g_{p_i} \in G^*$.

In this work, we build the library of task affordances as:

$$\mathcal{L} = \big\{ \mathcal{T}_1 \rightarrow \{\pi_{\tau_1}, A_{\tau_1}\}, \cdots, \mathcal{T}_n \rightarrow \{\pi_{\tau_n}, A_{\tau_n}\} \big\}, \quad (1)$$

where π_τ is a policy encoding the task in a generalisable form, and $\alpha_\tau \in A_\tau$ is a set of possible successful outcomes when executing π_τ. In our implementation, π_τ is based on dynamic movement primitives (DMPs) [19], [20]. DMPs are differential equations encoding behaviour towards a goal attractor. We initialise the policies via imitation learning, and use them to reproduce an observed motion while generalising to different start and goal locations, as well as task durations.

Regarding the set of possible successful outcomes $\alpha_\tau \in A_\tau$, we provide the robot with multiple experiences. We define the outcome α_τ as the state evolution of the object's action region S_O through the execution of the task. We employ mask RCNN (M-RCNN) [21] to train a

(a) G^* from [9] (b) g_{p_i} from [5] (c) Combined $g_{p_i} \in G^*$
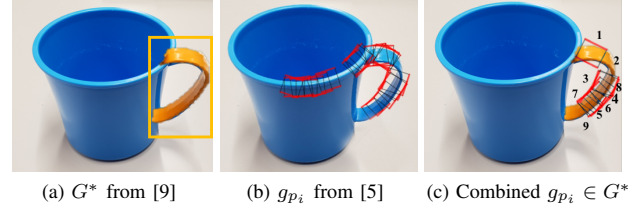
Fig. 3: Prediction of grasp affordance configurations for the pouring task. (a) Patch affording the pouring task, (b) reliable grasp configurations from DeepGrasp, (c) pruned space for reliable grasp candidates that afford the task pouring.

model that detects objects subparts as action regions S_O. As exemplified in Fig. 4, the action region state provides a meaningful indicator of the task. This information is used as the basis for our confidence metric, which evaluates the level of success of an affordance task for a grasping proposal.

C. Search-Based Self-Assessment of Task Affordances

The task policies π_τ learnt in Section III-B allow a previously experienced task from any candidate grasp $g_{p_i} \in G^*$ to be performed. Nonetheless, executing π_τ from any grasp configuration may not always lead to suitable performance. For example, Fig. 4 depicts the case where grasping the mug from g_{p_1} prevents the robot from performing a pouring task as adequately as when grasping it from g_{p_2}.

We propose to self-assess the outcome of executing the task's policy π_τ from $g_{p_i} \in G^*$ before deciding the most

(a) Unsuccessful pour (grasping at g_{p_1})

(b) Successful pour (grasping at g_{p_2})
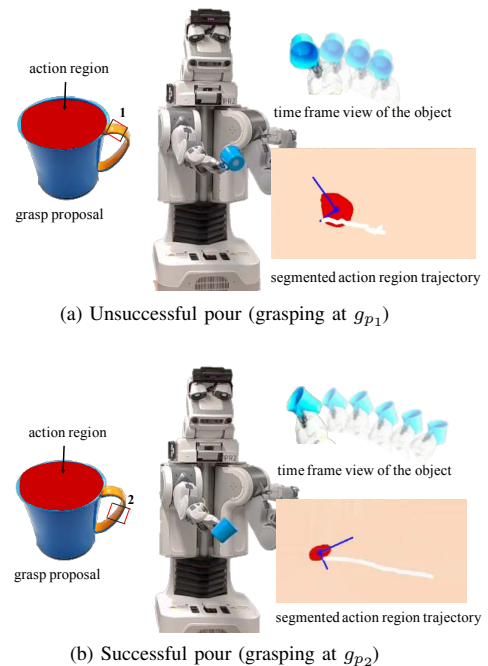
Fig. 4: Example of a pouring task from two different grasp configurations. Each situation illustrates the raw two-dimensional (2-D) camera input of the object and the segmented action region that affords the pouring task.

suitable grasp configuration g_p^* on a new object. This is efficiently done by forward simulation of the DMP-encoded π_τ. From each roll-out, we look at the object's state action region $\overline{\alpha}_\tau$ as a suitable task performance indicator. To this aim, we consider the entropy between the demonstrated successful task outcomes α_τ and the simulated outcome $\overline{\alpha}_\tau$ in the form of Kullback-Leibler divergence [22]:

$$D(\alpha_\tau || \overline{\alpha}_\tau) = \sum_{i \in I} \alpha_\tau(i) \log \left(\frac{\alpha_\tau(i)}{\overline{\alpha}_\tau(i)} \right), \quad (2)$$

which results in a low penalisation when the forward simulated outcome $\overline{\alpha}_\tau$ is similar to a previously experienced outcome in A_τ, and a high penalisation otherwise. Then, we propose to rank the grasping candidates $g_{p_i} \in G^*$ according to a confidence metric which estimates the suitability of a candidate g_{p_i} for a given $\mathcal{T}$ as:

$$C(g_{p_i}) = \max_{\alpha_\tau \in A_\tau} D^{-1}(\alpha_\tau || \overline{\alpha}_\tau). \quad (3)$$

Finally, we select the grasping configuration g_p^* among all grasping candidate $g_{p_i} \in G^*$ as:

$$g_p^* = \arg \max_{g_{p_i} \in G^*} C(g_{p_i}) \quad s.t. \quad C(g_{p_i}) > \delta, \quad (4)$$

which returns the grasp configuration with highest confidence of successfully completing the task. This assessment is subject to a minimum user-defined confidence level δ that rejects under-performing grasp configuration proposals. As explained in the experimental setup, such a threshold is adjusted from demonstration by a binary classifier.

D. One-Shot Self-Assessment of Task Affordances

The search-based strategy presented in Section III-C in the grasp affordance region can be time and resource consuming if performed for every single task affordance problem. Alternatively, we propose to augment the library in (1) with an approximate of the prior experienced outcomes α_τ per grasp configuration g_{p_i}, such that it allows for one-shot assessment. Namely, we extract the spatial transform of all experienced grasps with respect to the detected grasp affordance region G^*. The relevance of these transforms is ranked in a list R according to their confidence score computed following (3). Therefore, the augmented library is denoted as:

$$\mathcal{L} = \left\{ \mathcal{T}_1 \rightarrow \{\pi_{\tau_1}, A_{\tau_1}, R_{\tau_1}\}, \cdots, \mathcal{T}_n \rightarrow \{\pi_{\tau_n}, A_{\tau_n}, R_{\tau_n}\} \right\}. \quad (5)$$

At deployment time, we look at the spatial transform from the new grasping candidates that resembles the most well-ranked transform in R. This allows us to hierarchically self-assess the candidates by order of prospective success.

E. Deployment on Autonomous Agent

Algorithm 1 presents the outline of SAGAT's end-to-end deployment, which aims at improving the success of an autonomous agent when performing a task. Given visual perception of the environment, the desired affordance, the pre-trained model to extract the grasp affordance relation (see Section III-A), the model to detect the action region,

---

Algorithm 1: deployment of SAGAT

1 Input:
2   CVF: camera visual feed
3   *affordance*: affordance choice
4   graspAffordance: grasp affordance model
5   actionRegion: MRCNN learnt model
6   libTaskAffordances: task affordance library

7 Output:
8   g_p^*: most suitable grasp affordance configuration

9 begin
10    $G^* \leftarrow$ graspAffordance(CVF, *affordance*)
11    $S_O \leftarrow$ actionRegion(CVF, *affordance*)
12    $g_p \leftarrow$ libTaskAffordances(G^*, *affordance*)
13   **while not** isEmpty(g_p) **do**
14      $g_{p_i} \leftarrow$ popHighestCondifence(g_p)
15      $\overline{\alpha}_\tau \leftarrow$ forwardSimulateTask(g_{p_i}, S_O)
16   **if** prospectiveTaskSuccess($\overline{\alpha}_\tau$) **then**
17   **return** g_{p_i}

18   **return** none

---

and the learnt library of task affordances (see Section III-B to Section III-D) (lines 2 to 6), the end-to-end execution is as follows. First, the visual data is processed to extract the grasp affordance region (line 10) and the object's action region (line 11). The resulting grasp affordance region along with the desired affordance are used to estimate the grasp configuration proposals on the new object using the library of task affordances as prior experiences (line 12). The retrieved set of grasp configuration candidates is analysed in order of decreasing prospective success (line 13 to line 17) until either exhausting all candidates or finding a suitable grasp for the affordance task. Importantly, the hierarchy of the proposed self-assessment analysis allows for one-shot transfer of the grasp configuration proposals, i.e. to find, on the first trial, a suitable grasp affordance by analysing the top-ranked grasp candidate. Nonetheless, the method also considers the case that exhaustive exploration of all candidates might be required, thus ensuring algorithmic completeness.

Notably, the proposed method is not dependant on a particular grasp affordance or action region description. This modularity allows the usage of the proposed method in a wide range of setups. We demonstrate the generality of the proposed method by first, using multiple state-of-the-art approaches for grasp affordance detection, and then, determining the improvement on task performance and deployability when used altogether with our approach.

IV. EXPERIMENTAL EVALUATION AND DISCUSSION

The proposed methodology endows a robot with the ability to determine a suitable grasp configuration to succeed on an affordance task. Importantly, such a challenge is addressed without the need for extensive prior trials and errors. We demonstrate the potential of our method following the experimental setup described in Section IV-A and a thorough eval-
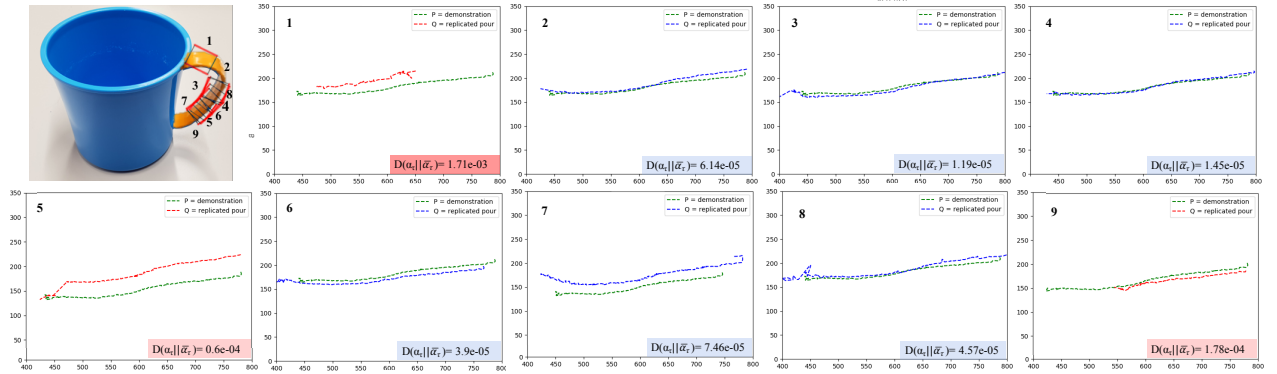
Fig. 5: Entropy measurements on the 2-D frame for the pouring task. We consider as reference a socially acceptable pouring demonstration (green) against successful (blue) and undesired (red) task repetitions from different grasp candidates. The candidates are numbered with the corresponding observed effect. Successful tasks present low entropy whereas undesired effects have higher entropy. Our proposal exploits this relation to discern among grasp candidates at deployment time.

uation based on the following tests: (i) the spatial similarity between learnt and computed configurations across objects (Section IV-B), (ii) the accuracy of the task affordance deployment when transferred to new objects (Section IV-C), and (iii) the performance of our proposal when compared to other methodologies (Section IV-D).

A. Experimental Setup

The end-to-end execution framework presented in Algorithm 1 is deployed on a PR2 robotic platform, in both simulated and real-world scenarios. We use a Kinect mounted on the PR2's head as our visual sensor and the position sensors on the right arm joints to encode the end-effector state pose for learning the task policies in the library.

We evaluate the proposed approach with an experimental setup that considers objects with variate affordable actions and suitable grasping configurations. Particularly, the library of task affordances is built uniquely using the blue mug depicted in Fig. 5, but evaluated with the objects depicted in Fig. 6. As can be observed, the training and testing sets present a challenging and significant variability on the grasp affordance relation. Our experimental setup also considers multiple affordances, namely: pouring, handover and shaking. The choice of these affordances is determined by those being both common among the considered objects and socially acceptable according to [9].

The task policy and its expected effect corresponding to each affordance are taught to the robot via kinaesthetic demonstration. The end-effector state evolution is used to learn the task policy in form of a set of DMPs, and the state evolution of the container's action region segmented on the 2-D camera frame to learn the expected effect. As depicted in Fig. 5 for the pouring task, the learnt policy is replicated 9 times from different grasping candidates, including suitable grasp affordances (blue) and undesired deployments (red).

The collected demonstrations are used to adjust the confidence threshold in (4) via a binary classifier, where the confidence level computed following (3) is the support, and

the label {"successful", "undesired"} is the target. Only successful deployments are included in the library.

B. Spatial Similarity of Grasp Configurations

Our method allows the system for one-shot transfer of grasp configurations to new objects. As explained in Section III-D, we rank the grasp candidates on new objects as those that closely resemble the experiences stored in the library of task affordances. This approximation is based on the expectation that similar spatial configurations should offer similar performance when dealing with the same task. In this set of experiments, we demonstrate the validity of such a hypothesis by evaluating the spatial similarity between the proposals estimated on new objects and the ones previously identified as suitable and stored in the library.

For an object, we calculate the Euclidean distance between the segmented action region S_O and the obtained grasp configuration g_p^*. Fig. 7 shows the obtained distances denoted as $d_h(S_O, g_p^*)$. The blue horizontal line represents the mean distance obtained during the demonstrations. Overall, we observe similar distances from action regions to grasp configurations across objects. For dissimilar cases such as 4 and 5 (ashtray and bowl respectively), the difference is

Fig. 6: Novel objects to test the self-assessed grasp transfer.

(a) Pour ($d_{hd} = 0.21$)

(b) Shake ($d_{hd} = 0.23$)

(c) Handover ($d_{hd} = 0.20$)
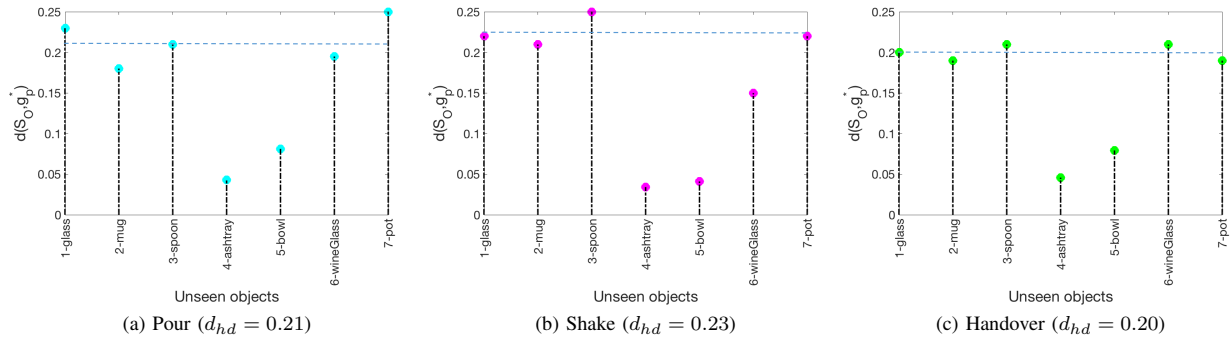
Fig. 7: Visualisation of the dissimilarity metric between an object's action region and the corresponding suitable grasp configuration, in comparison to the mean dissimilarity observed during the demonstrations (d_{hd}, blue horizontal line).

given by the fact that the obtained grasping region for most of the tasks lies on the edges of the object compartment. Even though these grasping configurations are relatively close to the action region, we will see on Table I that the average performance of the tasks is preserved.

To further evaluate similarity across obtained grasping configurations, we are also interested in how much the system prunes the grasping space based on the information stored in the library. As defined in (4), we use a confidence threshold for the pruning process of the grasping space. Thus, based on the prior of well-performing grasp configurations, highly dissimilar proposals are not considered on the self-assessed transfer process. Fig. 8 depicts the rejection rate of grasp configuration proposals per task affordance. From the plot, we see that the pouring task shows the highest rejection rate, especially for objects that have handles. This hints that for this task the grasping choice is more critical.

C. One-Shot Transfer of Task Affordances

The second experimental test analyses the performance of our method when addressing task affordances on new
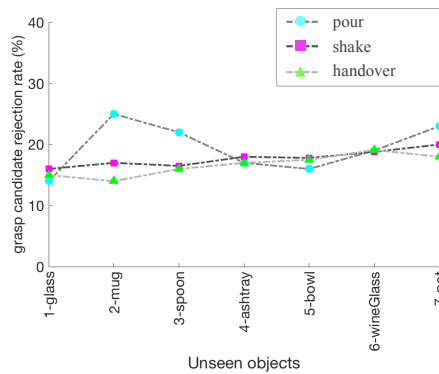
Fig. 8: Rejection rate of grasp candidates with prospective unsuccessful task deployment. Grasp configurations, as extracted with DeepGrasp [5], that do not relate to the prior on successful task deployment, as stored in the library, are rejected in the one-shot transfer scheme.

(a) Pour task affordance

(b) Shake task affordance

(c) Handover task affordance
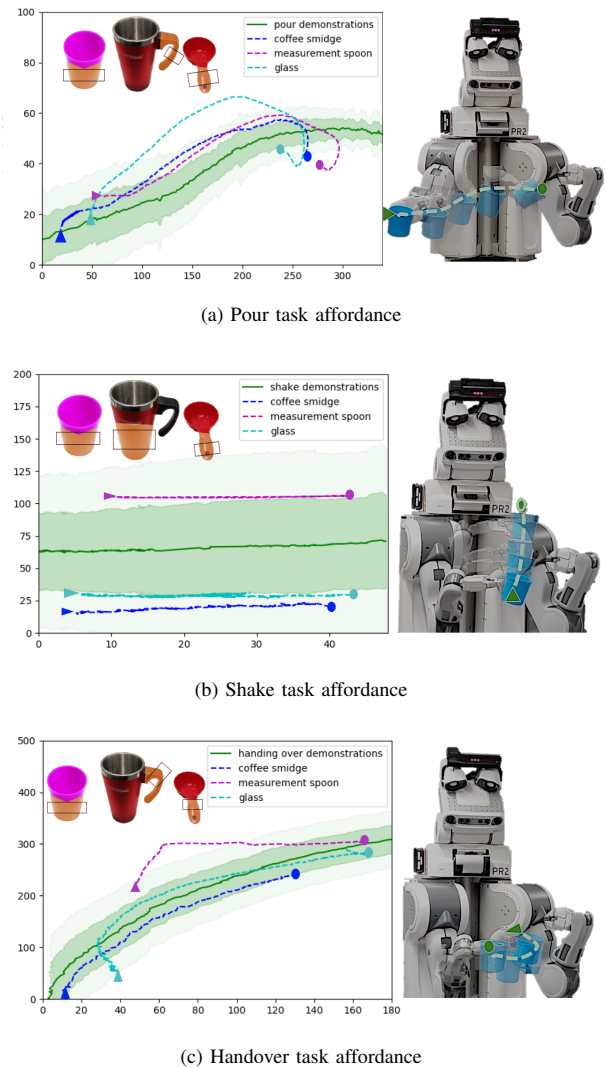
Fig. 9: Task affordance performance when deployed on novel objects (colour-coded lines) in comparison with the multiple successful demonstrations (green scale distribution).
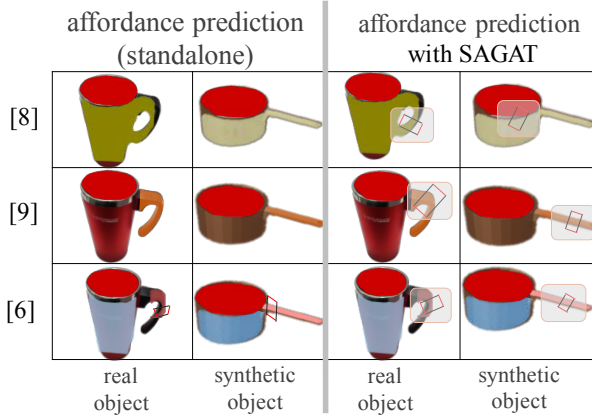
Fig. 10: Comparison of grasp affordance detection for the task of pouring with state-of-the-art methods and SAGAT. The resulting grasp configuration proposals obtained with SAGAT are highlighted for better visualisation.

	[8]	[8]+SAGAT	[9]	[9]+SAGAT	[6]	[6]+SAGAT
Pour	70%	**82%**	72%	**83%**	73%	**85%**
Shake	84%	**87%**	85%	**87%**	86%	**88%**
Handover	80%	**85%**	81%	**86%**	82%	**86%**

TABLE I: Comparison of success rates on task affordance deployment when using state-of-the-art grasp affordance extractors as stand-alone and with our method.

objects. The goal of this evaluation is to determine if the chosen grasp configuration enables objects to perform the task affordance as successfully as the prior stored in the library. Fig. 9 depicts the mean and variance (green scale) of the prior experiences in the library for the tasks pour, shake and handover. Each task was performed with three real objects with notably different features: a travel mug (dark blue), measurement spoon (magenta) and a glass (blue). The resulting effect when performing the tasks from the computed grasping configuration is colour-coded on top of the prior experiences distribution.

Subject to the task affordance, the three objects show different grasp affordance regions. After the one-shot self-assessment procedure, the computed grasp configurations are the most spatially similar to the most successful grasp configuration in the experience dataset. Importantly, as shown in Fig. 9, this strategy is invariant to the initial and final states of the task. This is reflected in the obtained task affordance effect, which falls inside the variance of the demonstrations.

D. Comparison of Task Deployment Reliability

The last experimental test is to demonstrate at which level the proposed method enhances the task deployment reliability when used in conjunction with methods for grasp affordance detection [6], [8], [9]. To conduct this evaluation, we use the open-source implementations of [6], [8], [9] on all objects illustrated in Fig. 6, in the real and simulated robotic platform. The obtained grasp regions are used to execute the task in two different ways: (i) in stand-alone fashion, i.e. as originally proposed, and (ii) as input of our SAGAT approach to determine the most suitable grasp candidate. Fig. 10 shows some examples of the grasp affordance detected with the previously mentioned methods and our approach.

We use the policies in the learnt library of task affordances to replicate the pour, shake and handover tasks on each object, for each grasp affordance, and for each method when used as stand-alone and combined with SAGAT. This results

in a total of 126 tasks deployments on the robotic platform[1]. Table I summarises the obtained results. As can be observed, deploying a task using state-of-the-art methods on grasp affordance detection provides an average success rate of 79.2% across tasks. With our approach, the deployability success is enhanced for all the tasks, with an average rate of 85.4%. Interestingly, the 5.2% improvement is not equally distributed across tasks; more challenging tasks experience a higher success rate. This is the case of the pouring tasks where deployability success is increased by 11.67%.

V. CONCLUSIONS AND FUTURE WORK

In this paper, we presented a novel experience-based pipeline for self-assessment of grasp affordance transfer (SAGAT). Our approach enhances the deployment reliability of current state-of-the-art methods on grasp affordance detection, by extracting multiple grasp configuration candidates from a given grasp affordance region. The outcome of executing a task from different grasp candidates is estimated via forward simulation. These estimates are evaluated and ranked via a heuristic confidence function in relation to task performance and grasp configuration candidates. Such information is stored in a library of task affordances, which serves as a basis for one-shot transfer estimation to identify grasp affordance configurations similar to those previously experienced, with the insight that similar regions lead to similar deployments of the task. We evaluate the method's efficacy on novel task affordance problems by training on a single object and testing on multiple new ones. We observe a significant performance improvement up to approximately 11.7% in our experiments when using our proposal in comparison to state-of-the-art approaches on grasp affordance detection. Experimental evaluation on a PR2 robotic platform demonstrates highly reliable deployability of the proposed method to deal with real-world task affordance problems.

This work encourages multiple interesting directions for future work. Our follow-up work will study a unified probabilistic framework to infer the most suitable grasp affordance candidate. We envision that this will allow sets of actions and grasps to be predicted when dealing with multiple correlated objects in the scene. Consequently, including a task planning layer that connects actions with grasp affordances. Another interesting extension is the assessment of the end-state comfort-effect for grasping in human-robot collaboration tasks, such that the robot's grasp affordance considers the human's grasp capabilities.

---

[1] A compilation of experiments can be found in: https://youtu.be/nCCc3_Rk8Ks

VI. Acknowledgements

References

[1] L. Jamone, E. Ugur, A. Cangelosi, L. Fadiga, A. Bernardino, J. Piater, and J. Santos-Victor, "Affordances in psychology, neuroscience, and robotics: A survey," *IEEE Transactions on Cognitive and Developmental Systems*, vol. 10, no. 1, pp. 4–25, 2018.

[2] H. Min, C. Yi, R. Luo, J. Zhu, and S. Bi, "Affordance research in developmental robotics: a survey," *IEEE Transactions on Cognitive and Developmental Systems*, vol. 8, no. 4, pp. 237–255, 2016.

[3] J. Gibson, "The theory of affordances," in *Perceiving, Acting, and Knowing: Toward and Ecological Psychology* (R. Shaw and J. Bransford, eds.), pp. 62–82, Hillsdale, NJ: Erlbaum, 1977.

[4] I. Lenz, H. Lee, and A. Saxena, "Deep learning for detecting robotic grasps," *International Journal of Robotics Research*, vol. 34, no. 4-5, pp. 705–724, 2015.

[5] F.-J. Chu, R. Xu, and P. A. Vela, "Real-world multiobject, multigrasp detection," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3355–3362, 2018.

[6] F.-J. Chu, R. Xu, and P. A. Vela, "Learning affordance segmentation for real-world robotic manipulation via synthetic images," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1140–1147, 2019.

[7] J. Bohg and D. Kragic, "Learning grasping points with shape context," *Robotics and Autonomous Systems*, vol. 58, no. 4, pp. 362–377, 2010.

[8] T.-T. Do, A. Nguyen, and I. Reid, "Affordancenet: An end-to-end deep learning approach for object affordance detection," in *International Conference on Robotics and Automation (ICRA)*, 2018.

[9] P. Ardón, È. Pairet, R. P. Petrick, S. Ramamoorthy, and K. S. Lohan, "Learning grasp affordance reasoning through semantic relations," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 4571–4578, 2019.

[10] L. Montesano, M. Lopes, A. Bernardino, and J. Santos-Victor, "Learning object affordances: From sensory–motor coordination to imitation," *IEEE Trans. Robotics*, vol. 24, pp. 15–26, 2008.

[11] K. Fang, Y. Zhu, A. Garg, A. Kurenkov, V. Mehta, L. Fei-Fei, and S. Savarese, "Learning task-oriented grasping for tool manipulation from simulated self-supervision," *The International Journal of Robotics Research*, p. 0278364919872545, 2019.

[12] A. Mandlekar, Y. Zhu, A. Garg, J. Booher, M. Spero, A. Tung, J. Gao, J. Emmons, A. Gupta, E. Orbay, *et al.*, "Roboturk: A crowdsourcing platform for robotic skill learning through imitation," in *Conference on Robot Learning*, pp. 879–893, 2018.

[13] O. Kroemer, E. Ugur, E. Oztop, and J. Peters, "A kernel-based approach to direct action perception," in *2012 IEEE International Conference on Robotics and Automation*, pp. 2605–2610, IEEE, 2012.

[14] N. Krüger, C. Geib, J. Piater, R. Petrick, M. Steedman, F. Wörgötter, A. Ude, T. Asfour, D. Kraft, D. Omrčen, *et al.*, "Object–action complexes: Grounded abstractions of sensory–motor processes," *Robotics and Autonomous Systems*, vol. 59, no. 10, pp. 740–757, 2011.

[15] D. Song, K. Huebner, V. Kyrki, and D. Kragic, "Learning task constraints for robot grasping using graphical models," in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pp. 1579–1585, IEEE, 2010.

[16] L. Montesano and M. Lopes, "Learning grasping affordances from local visual descriptors," in *Development and Learning, 2009. ICDL 2009. IEEE 8th International Conference on*, pp. 1–6, IEEE, 2009.

[17] B. Moldovan, P. Moreno, M. van Otterlo, J. Santos-Victor, and L. De Raedt, "Learning relational affordance models for robots in multi-object manipulation tasks," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pp. 4373–4378, IEEE, 2012.

[18] È. Pairet, P. Ardón, M. Mistry, and Y. Petillot, "Learning and composing primitive skills for dual-arm manipulation," in *Annual Conference Towards Autonomous Robotic Systems*, pp. 65–77, Springer, 2019.

[19] A. J. Ijspeert, J. Nakanishi, H. Hoffmann, P. Pastor, and S. Schaal, "Dynamical movement primitives: learning attractor models for motor behaviors," *Neural computation*, vol. 25, no. 2, pp. 328–373, 2013.

[20] È. Pairet, P. Ardón, M. Mistry, and Y. Petillot, "Learning generalizable coupling terms for obstacle avoidance via low-dimensional geometric descriptors," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3979–3986, 2019.

[21] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, pp. 2961–2969, 2017.

[22] F. Pérez-Cruz, "Kullback-leibler divergence estimation of continuous distributions," in *2008 IEEE international symposium on information theory*, pp. 1666–1670, IEEE, 2008.

Critical Review - Closing

In this part of the thesis, we have formalised a strategy to acquire, store and recall abstractions with the purpose of bootstrapping robotic behaviour. We started arguing the ease of acquiring motions, and their suitability, to bootstrap robotic behaviour on varied tasks. Obtained behavioural samples are stored in a library of motions with semantic attachment to the task they relate to. Then, on deployment, such relational database enables retrieving a sample of a motion solving a problem related to the task at hand. To conclude this part, we provide a summary of the key results and discuss how the findings form a coherent piece of this thesis.

6.1 Results

The presented strategy of bootstrapping problem-relevant robotic behaviour via motion-based relational databases has been developed in two stages. Each iteration has been supported with experimental evaluation, in both synthetic environments and real-world robotic platforms. In particular, we provided the following results:

Chapter 4   We demonstrated our framework to enable the bootstrapping of relevant robot behaviour in accordance with a given planning problem. We showed that considering trajectories as behavioural abstractions allows acquiring samples from human demonstrations, at different reference frames. Additionally, we explored the possibility of concatenating sequentially and simultaneously to generate more complex behaviours than those encoded in the libraries.

Chapter 5   We demonstrate our framework generality to extend the autonomy capabilities of a robotic agent. Linking our framework to affordances enables a robot agent to cope with planning problems as "pour the content of the cup into the blue bowl"; the framework determines the relevant behavioural abstraction to pour, whereas the link to affordances deal with goal and grasp identification. In other words, the framework's linkage to affordances bridges high-level problem definitions to planning-understandable objectives.

6.2 Discussion

Our efforts in the direction of bootstrapping robotic behaviour via abstractions to support efficient motion synthesis result in the following contribution: a framework that leverages intuition on the behavioural requirements of a task to provide some bias, in form of a trajectory, to the motion synthesis process. Such capability is relevant to the prospect of robot autonomy, as the framework, altogether with its tight link to affordances, is capable of dealing with more generic, less human-processed, planning problems.

We exemplified the applicability of the framework by leveraging the extracted behavioural bias with DMPs. Importantly, the generality of the proposed framework allows its deployment on different robotic tasks and robotic platforms. In fact, this part of the thesis has demonstrated the usability of the framework in several tasks, and two different robotic platforms (iCub and PR2). Results in this project consolidate the assumption that the remaining of the thesis builds on; the existence of a strategy capable of identifying a behavioural abstraction (set of trajectories or, equivalently, a set of DMPs) relevant to a particular problem, if any.

In the remaining of the thesis, we investigate methods to leverage the extracted behavioural abstractions in differing tasks contexts (e.g., different start and goal configurations, or presence of obstacles). In particular, in Part III, we hold on the assumption that we have some information a-priori about the task to explore strategies to generalise the motion samples to more varied tasks contexts. Then, in Part IV, we explore the applicability of behavioural abstractions in complex, ever-changing planning problems, from which there is not a-priori information.

We have discussed some directions for future work in the corresponding manuscripts. Among them, we are particularly keen on the possibilities that such a framework offers for task planning.

Part III

ENRICHING BEHAVIOUR FROM EXPERIENCE GENERALISATION

Critical Review - Opening

Building on the capability of bootstrapping robot behaviour from a library of experiences presented in Part II [123, 5], in this part of the thesis, we focus on the challenge of generalising those experiences to novel task contexts. Enabling a robot to generalise prior information to varied task contexts is essential as the prior might not always be tightly relevant to the current planning problem. Thus, our aim is to formulate planning strategies capable of leveraging any prior experience to guide robot behaviour in significantly dissimilar task contexts.

7.1 Objectives and Contribution

There are many aspects that might differentiate instances of the same task. In the scope of this thesis, we focus on the generalisation needs involved with changes on the required start and goal configuration of the robotic system, and variability on the number of obstacles, their pose and geometric constitution. Therefore, in order to enable a robot leveraging from a prior experience across task instances, we need to (1) identify relevant features from the observed data, to then (2) reason about how these features extrapolate to new tasks. We present two independent approaches to these objectives, which lead to two different notable contributions:

Chapter 8   A hierarchical framework which safely modulates an ongoing DMP-encoded policy to avoid obstacles. The proposed approach follows a multilayered perception-decision-action analysis which (i) extracts unified system-obstacle low-dimensional geometric descriptors, then (ii) exploits them to rapidly reason about the environment with a combination of heuristics and learning techniques, and finally (iii) guides and regulates the obstacle avoidance behaviour with a conjunction of coupling terms modulating the task policy encoded as a DMP.

Chapter 9   Two new experience-based planners: the uni-directional experience-driven random trees (ERT) and its bi-directional version (ERTConnect). Both methods are tree sampling-based planners that iteratively exploit a single prior path experience to

ease the capture of connectivity of the space. At each iteration, a segment of the prior is extracted and semi-randomly modulated to generate a task-relevant motion. The obtained motions are sequentially concatenated to compose a task-relevant tree, such that a trace along the edges constitutes a solution to a given planning problem.

7.2   Methodology

The library of experiences presented in Part II [123, 5] employed DMPs to encode observed robotic behaviour, and generalise it to novel task instances. Whilst the concept of the library is not restricted to any particular motion policy, we initially adopted the DMP motion descriptor to pursue the aims of this part of the thesis; their generalisation capabilities already extend to different start and goal configurations, task duration, as well as point-based obstacles using the coupling term formulated in [61]. Such a coupling term creates a parameter-dependent repulsive force that modulates the underlying DMP for the system to steer away from obstacles. The main limitation of such an approach, however, is the assumption that an obstacle can be reduced to a point-mass, as well as the obstacle avoidance behaviour being parameter dependant.

Early work [120, 121] of this thesis mitigated the obstacle avoidance limitations of DMPs with a learning from demonstration-based approach. In particular, we proposed teaching the robot varied obstacle avoidance styles, each being suitable to circumnavigate obstacles of different sizes. We captured each obstacle avoidance style by estimating the parametrisation of the coupling term in [61] that explained the contrast between a perturbationless task, and one with an obstacle; their dynamical difference was assumed to be purely due to the presence of an obstacle. Then, the estimate of the best-approximating parameterisation was computed applying least mean squares (LMS) regression on the log-linearised coupling term.

We tested such approach with the real iCub humanoid robot (see Figure 7.1). We demonstrated, with respect to a symbolic point-mass obstacle (red sphere), two different obstacle avoidance styles: reckless (see Figure 7.1a) and conservative (see Figure 7.1b). While the former steers around the point-mass obstacle closely (behaviour potentially suitable to avoid small obstacles), the latter keeps a larger distance to it (behaviour potentially suitable for bigger obstacles). The recorded raw proprioception data of these two kinaesthetic demonstrations is respectively portrayed in Figure 7.1c and Figure 7.1d. The retrieved trajectories were pre-processed by (i) filtering out outliers and high-frequency noise, and (ii) projecting the resulting information to the 2D space defined by the two principal components of the data (i.e., we assumed that avoiding the obstacle was intended along the same direction during the whole demonstration). Figure 7.1e and Figure 7.1f show the preprocessed data (red trajectories), later used to learn the

(a)

(b)

(c)

(d)

(e)

(f)
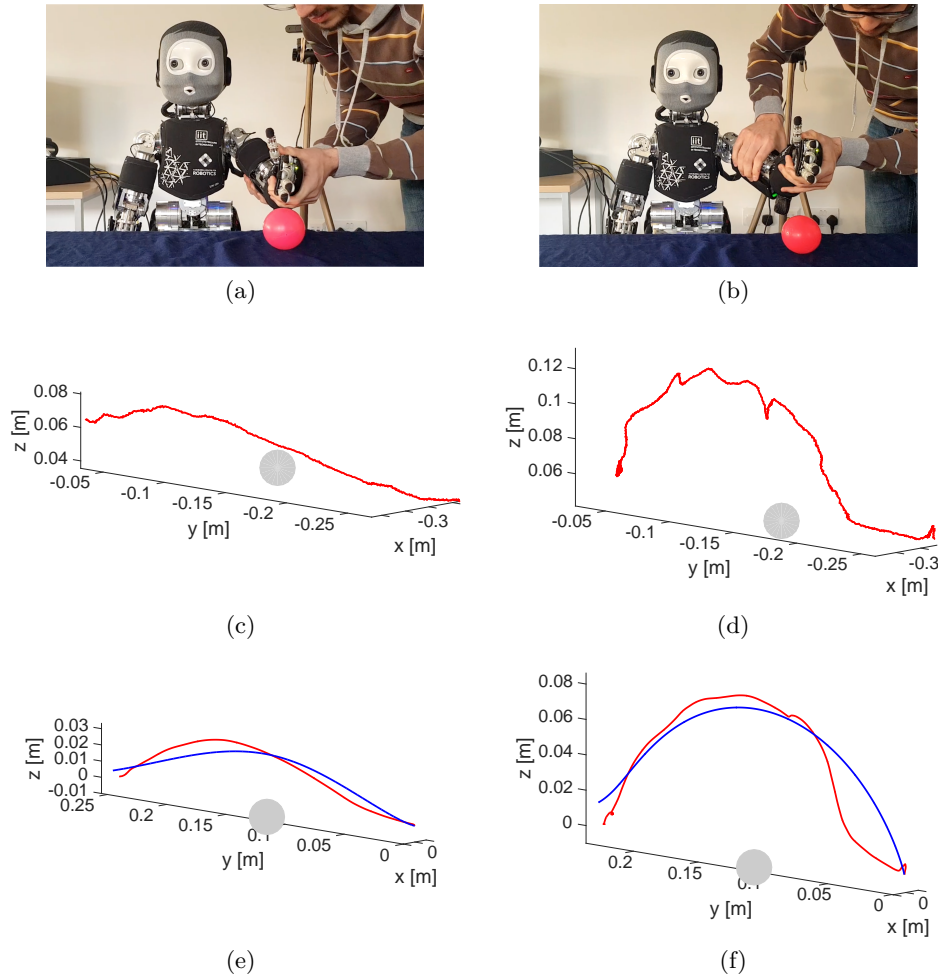
Figure 7.1: iCub humanoid robot learning the primitive skill of obstacle avoidance with two different behaviours: reckless (first column) and convervative (second column). (a)-(b) Human demonstrations to avoid an obstacle (red sphere). (c)-(d) iCub's proprioception data. (e)-(f) Processed proprioception data (red trajectory) and learnt behaviour (blue trajectory).
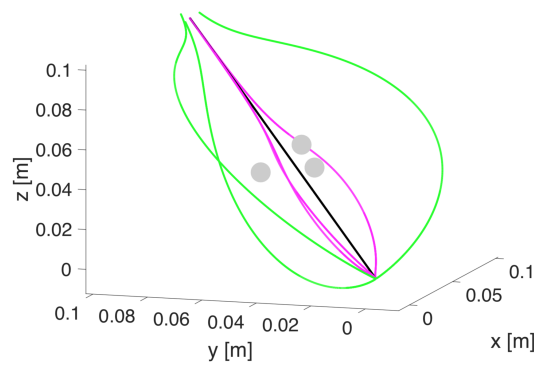
Figure 7.2: Generalisation capabilities to multiple obstacles and in 3D scenarios of the learnt reckless (magenta trajectory) and conservative (green trajectory) obstacle avoidance behaviours.

parameters defining the demonstrator's obstacle avoidance behaviour. The encoded reckless and conservative styles are respectively depicted in Figure 7.1e and Figure 7.1f (blue trajectories). Overall, learning the parameters of the coupling term in [61] from human demonstrations avoids blindly hand-tuning the obstacle avoidance behaviour, allows reproducing the observed styles to cope with obstacles of different geometric constitution, and enables the robot to generalise such behaviour to scenes with multiple obstacles (see Figure 7.2).

Our learning by demonstration-based approach showed to be suitable to capture, reproduce and generalise the demonstrator's style in avoiding obstacles. However, we did not pursue such direction due to the dependency on high-quality data, being a time-consuming teaching process, and requiring an additional layer of reasoning to select the most suitable style given a particular obstacle geometric constitution. Instead, we used these challenges to motivate the research project detailed in Chapter 8 [124]. In there, we designed a semi-supervised learning strategy to learn suitable coupling term parametrisations subject to obstacles of different shapes and sizes. To make the learning and generalisation tractable, the obstacle avoidance problem was studied in the workspace, particularly on a 2D plane, whose orientation in the workspace was guided by some heuristics. Overall, such an approach to generalise DMP-encoded behaviours showed high reliability, even in completely unknown environments, while offering nearly real-time performance. These desirable properties, however, were achieved at the cost of not providing kinematic feasibility guarantees on the computed path due to planning in the workspace.

Bearing these limitations in mind, we adopted the strengths of our research into a sampling-based planning scheme, as detailed in in Chapter 9 [125]. Namely, we thought of the malleable nature of a DMP under the influence of a coupling term as that of an affine transform. With that insight, we proposed two experience-based planners, the ERT and its bi-directional version ERTConnect, which repeatedly apply semi-random affine transforms onto parts of a prior path experience to build a tree of locally modulated micro-experiences. Overall, such a strategy demonstrated to allow efficiently leveraging a single prior experience to compute motion plans quickly. We also showed how to select such unique prior path experience from a library of task-relevant experiences. In other words, we designed this research on the capabilities developed earlier in this thesis, but instead of a library of DMPs (each task being encoded with a unique DMP), we envisioned a library of libraries, where each task's prior is a library of task-relevant prior paths.

Learning Generalizable Coupling Terms for Obstacle Avoidance

In this chapter, we focus on extending the generalisation capabilities of dynamic movement primitives (DMPs) to also cope with obstacles of any geometric constitution. We design a hierarchical framework that generates reactive yet bounded obstacle avoidance DMP modulations through a multi-layered analysis. The framework leverages the strengths of learning techniques and the versatility of DMPs to efficiently unify perception, decision, and action levels via low-dimensional geometric descriptors of the environment. Experimental evaluation on synthetic environments and a real anthropomorphic manipulator proves that our method's robustness and generalisation capabilities, regardless of the obstacle avoidance scenario, makes our approach suitable for robotic systems in real-world environments.

All proposed work is described in detail in the following published journal article:

Learning Generalizable Coupling Terms for Obstacle Avoidance via Low-Dimensional Geometric Descriptors

Èric Pairet [ID], Paola Ardón [ID], Michael Mistry [ID], and Yvan Petillot [ID]

Abstract—**Unforeseen events are frequent in the real-world environments where robots are expected to assist, raising the need for fast replanning of the on-going policy to guarantee operational safety. Inspired by human behavioral studies of obstacle avoidance and route selection, this letter presents a hierarchical framework that generates reactive yet bounded obstacle avoidance behaviors through a multi-layered analysis. The framework leverages the strengths of learning techniques and the versatility of dynamic movement primitives to efficiently unify perception, decision, and action levels via environmental low-dimensional geometric descriptors. Experimental evaluation on synthetic environments and a real anthropomorphic manipulator proves the robustness and generalization capabilities of the proposed approach regardless of the obstacle avoidance scenario.**

Index Terms—**Collision avoidance, reactive and sensor-based planning, autonomous agents.**

Fig. 1. Proposed hierarchical framework for learning and producing generalisable obstacle avoidance behaviours. Pre-planned start-go-goal (blue) and modulated policy (red).

I. INTRODUCTION

ROBUST reactive behaviours are essential to ensure the safety of robots operating in unstructured environments. For instance, the on-going pick-and-place policy of a robotic system sorting and storing items in a home environment might be interrupted by the sudden appearance of an obstacle in the middle of a pre-planned trajectory. In this scenario, the robot must be able to modulate its behaviour online to succeed in its task while providing some safety guarantees. Given the expertise of humans in dealing with these conditions, it is natural to adopt human behaviour for robotic control.

 Human behavioural studies of obstacle avoidance and route selection [1] have shown that the dynamics of perception and action consist of (i) identifying the informational variables useful to guide behaviour and to regulate action, and (ii) interacting with the environment using a particular set of

dynamic behaviours. One possible policy descriptor allowing for this hierarchical control are dynamic movement primitives (DMPs) [2]. dynamic movement primitives (DMPs) are differential equations encoding kinematic control policies towards a goal attractor. Their transient behaviour can be shaped via a non-linear forcing term, which can be initialised via imitation learning and used to reproduce an observed motion while generalising to different start and goal locations, as well as task durations.

A key feature of DMPs is that they allow for online modulation via coupling term functions that create a forcing term. Coupling terms have been exploited for many applications, such as avoidance of joint and workspace limits [3], force control for environment interaction [4], [5], dual-arm manipulation [4], [6] and reactive obstacle avoidance [7]–[10]. This work focuses on the latter challenge, which historically has been approached using potential fields [7], analytical [8] and learning methods [9], [10] (see Section II). As further discussed in Section II-C, analytical formulations become less reactive for imminent collisions (dead-zone problem). Moreover, these approaches do not provide any guidance to the reactive behaviour, thus limiting their applicability to free-floating obstacles. Additionally, analytical formulations uniquely deal with point-mass obstacles and systems. In an attempt to address this latter issue, recent proposals learn coupling terms for a small set of obstacle geometries described by an array of markers on their surface [9], [10], but they fail to generalise actions to novel obstacles. These works are notable in learning the coupling terms from human demonstration. Nonetheless, providing a rich set of demonstrations involving various obstacles geometries can be time-consuming and prone to measurement noise.

This letter presents the hybrid DMP-learning-based obstacle avoidance framework schematised in Figure 1. The proposed approach addresses the limitations of the precedent works with a layered perception-decision-action analysis [1]. The main contributions at the action level (see Section III) are (i) reformulating the coupling terms to provide dead-zone free behaviours, and (ii) guiding the obstacle avoidance reactivity to satisfy task-dependant constraints, while the main contributions at the perception-decision level (see Section IV) are (iii) regulating action according to the extracted unified system-obstacle low-dimensional geometric descriptor, and (iv) learning to regulate the action level via exploration of the parameter space. The experimental evaluation reported in Section V demonstrates that the overall proposed approach generalises obstacle avoidance behaviours to novel scenarios, even when those involve multiple obstacles, or are uniquely described by partial visual-depth observations.

II. RELATED WORK

This letter proposes a reactive approach that endows a system with the ability to modulate its policy to avoid unexpected obstacles. The selected strategy uses DMPs for encoding any desired policy and defining an obstacle avoidance behaviour as a coupling term. This section introduces DMPs and coupling terms for obstacle avoidance as they constitute the fundamentals of this work.

A. Dynamic Movement Primitives

DMPs are a versatile framework that encode primitive motions or policies as nonlinear functions called forcing terms [2]. The DMPs equations define the system's state transition, which can be converted into actuator commands by means of inverse kinematics and inverse dynamics. For a one-degree of freedom, long-plural-form = degrees of freedom (DoF) system, the system's state transition is described by the following set of nonlinear differential equations, known as the transformation system:

$$\tau \dot{z} = \alpha_x(\beta_x(g_x - x) - z) + f(\cdot) + C(\cdot), \quad (1)$$

$$\tau \dot{x} = z, \quad (2)$$

where τ is a scaling factor for time, x is the system's position, z and $\dot{z}$ respectively are the scaled velocity and acceleration, α_x and β_x are constants defining the attraction dynamics towards the model's attractor g_x, and $f(\cdot)$ and $C(\cdot)$ are the forcing and coupling term, respectively.

The forces generated by the forcing and coupling terms define the transient behaviour of the transformation system. It is common to model the forcing term $f(\cdot)$ as a weighted linear combination of nonlinear radial basis functions (RBFs). The evaluation of $f(\cdot)$ at phase $k \in \mathbf{k}$ is defined as:

$$f(k) = \frac{\sum_{i=1}^{N} w_i \Psi_i(k)}{\sum_{i=1}^{N} \Psi_i(k)} k, \quad (3)$$

$$\Psi_i(k) = \exp\left(-h_i(k - c_i)^2\right), \quad (4)$$

where c_i and $h_i > 0$ are the centres and widths, respectively, of the $i \in [1, N]$ RBFs, which are weighted by w_i and distributed along the trajectory. The weights can be initialised via imitation
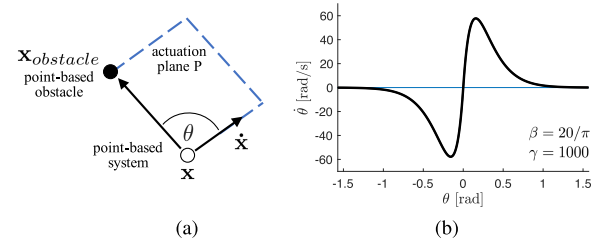
Fig. 2. Original coupling terms for obstacle avoidance [8]. (a) Heading angle θ according to velocity vector $\dot{x}$ and the relative obstacle-system position in P-plane. (b) Change of steering angle $\dot{\theta}$ subject to heading angle θ as defined by (7).

learning and used to reproduce the motion with some generalisation capabilities to changes in start and goal positions. The duration of the motion can be adjusted by the scaling factor τ, which modifies the canonical system defining the transient behaviour of the phase variable k as:

$$\tau \dot{k} = -\alpha_k k, \quad (5)$$

where the initial value of the motion's phase $\mathbf{k}(0) = 1$ and α_k is a positive constant.

A common strategy to extend the spatial generalisation capabilities of DMPs is to reference them in a local frame, whose pose in the space is task-dependent [2], [10]. In this work's context, the unit vectors of the local frame are defined as follows: the x-axis points from the start position towards the goal position, the z-axis points upwards and is orthogonal to the local x-axis, and the y-axis is orthogonal to both local x-axis and z-axis following the right-hand convention.

A robot with multiple DoFs uses a transformation system for each DoF, but they all share the same canonical system.

B. Coupling Terms for Obstacle Avoidance

Early coupling terms for obstacle avoidance were formulated as repulsive potential fields [7]. Potential fields suffer from local minima and can be computationally expensive to calculate on the fly. Alternatively, some coupling terms analytically formalise the influence of an obstacle on the system's behaviour [8]. As depicted in Figure 2a, a point-mass system with position $\mathbf{x} \in \mathbb{R}^3$ and velocity $\dot{x} \in \mathbb{R}^3$ has a heading $\theta \in SO(2)$ towards a point-mass obstacle. To avoid a collision, the coupling term generates a repulsive force:

$$C(\cdot) = \mathbf{R} \, \dot{x} \, \dot{\theta}, \quad (6)$$

where $\mathbf{R} \in SO(3)$ is a $\pi/2$ rotation matrix around the vector $\mathbf{r} = (\mathbf{x}_{obstacle} - \mathbf{x}) \times \dot{x}$. The respective obstacle-system position $\mathbf{x}_{obstacle} - \mathbf{x}$ and the system's velocity $\dot{x}$ define the plane $P \in \mathbb{R}^2$ where the system is desired to steer away from the obstacle with a turning velocity $\dot{\theta}$ defined as:

$$\dot{\theta} = \gamma \, \theta \exp(-\beta \, |\theta|), \quad (7)$$

where γ and β respectively scale and shape the mapping $\theta \rightarrow \dot{\theta}$ defined in (7) and represented in Figure 2b.

Building on (6)–(7), human demonstrations were used to retrieve the required parameters to circumvent two non-point obstacles, particularly a sphere and a cylinder [9]. More recently, coupling terms were formulated as independent neural networks
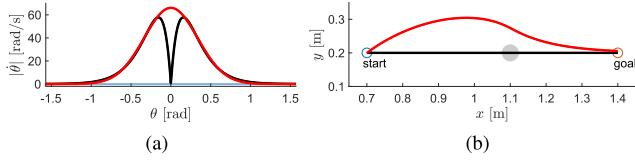
Fig. 3.   Dead-zone issue in the original (6)–(7) (black) and proposed (8) (red) coupling terms. (a) (8) reacts for narrow headings towards the obstacle. (b) (6)–(7) fails where (8) smoothly circumvents the point-mass obstacle (grey circle).

(NNs) modelling the desired obstacle avoidance behaviour for a sphere, a cylinder and a cube [10]. These methods do not provide any strategy to avoid obstacles not observed in training time, and they rely on markers identifying an obstacle's boundaries. Their evaluations are conducted either in simulation or in single-obstacle scenarios. Hence, their performance in realistic scenarios is yet to be tested.

C. Discussion and Contribution

State-of-the-art on coupling terms modelling obstacle avoidance behaviours suffers from four major limitations. First, as illustrated in Figure 3, the analytical term (6)–(7) has a dead-zone where the system becomes less reactive as the heading towards the obstacle narrows, thus compromising the method's reliability. Second, there is no strategy to guide the behaviour's reactivity towards a preferred route to circumnavigate an obstacle. For example, in the scenario depicted in Figure 1, there is no constraint on the reactive behaviour preventing the system from hitting the table. Third, when attempting to deal with non-point obstacles, their performance drastically decreases for novel scenarios due to the absence of global features identifying the obstacle geometry during the learning process. Fourth, these works learn the coupling terms from demonstration, which can be time-consuming and prone to measurement noise.

All these issues are jointly addressed within the proposed hierarchical framework, which hybridises the versatility of DMPs and the strengths of learning techniques. Specifically, in Section III, (6) (7) is reformulated at the action level as a conjunction of coupling terms whose obstacle avoidance behaviour is dead-zone free and can be guided. Then, in Section IV, the formalised action level is exploited to learn via exploration of the parameter space how to regulate the behaviour subject to both the end-effector's and obstacle's geometric properties. This work considers a unified system-obstacle low-dimensional geometric descriptors identifying the relevant features to the action level, thus allowing for enhanced generalisation even in novel real-world scenarios.

III. COUPLING TERMS FOR DEAD-ZONE FREE AND GUIDED OBSTACLE AVOIDANCE

The proposed hierarchical framework to learn and produce generalisable obstacle avoidance behaviours regardless of the scenario comprises three layers. The DMP-based action level is formalised as a composition of two coupling terms which (i) generate robust obstacle avoidance behaviours, and (ii) guide these in a particular direction of the task space. The parametrisation needs of these terms allow for regulating their actuation scope via reasoning at the decision level.

A. Inherently Robust Obstacle Avoidance

Current coupling terms for obstacle avoidance in the literature suffer from dead-zones, i.e. a heading range towards the obstacle for which the system becomes incoherently less reactive. Ideally, the expected behaviour of those terms would be to become more reactive as (i) the heading of the system is more aligned towards an obstacle, and (ii) the system-obstacle distance is smaller. Bearing these conditions in mind, the coupling term in (6)–(7) is reformulated as:

$$C_{\mathrm{OA}}(\cdot) = \mathbf{R}\,\dot{\mathbf{x}}\,\alpha\,\mathrm{sign}(\theta)\exp\!\left(-\frac{\theta^2}{\psi^2}\right)\exp\!\left(-\kappa\,d^2\right), \quad (8)$$

where $\alpha\,\mathrm{sign}(\theta)\exp\!\left(-\theta^2/\psi^2\right)$ addresses the first issue by shaping the absolute change of steering angle as a zero-mean Gaussian-bell function, and $\exp\!\left(-\kappa\,d^2\right)$ tackles the second requirement by regulating the coupling term effect according to a parameter k and the system-obstacle distance d.

Figure 3 highlights the increase in robustness of the formulated coupling term (8) in contrast to the original term (6)–(7). While the original coupling term (black curves) produces low reactivity for narrow headings towards an obstacle, the dead-zone free proposal (red curves) reacts the most (see Figure 3a). This reformulation has a significant impact in the task space, where (8) succeeds on a scenario where (6)–(7) fails to generate an obstacle avoidance behaviour which does not collide with the point-mass obstacle (see Figure 3b).

B. Guiding the Obstacle Avoidance Reactivity

The velocity vector $\dot{\mathbf{x}}$ of a point-mass system also represents the system's orientation. Consequently, $\dot{\mathbf{x}}$ plays a critical role in determining both the actuation P-plane and the direction of turning $\dot{\theta}$. Overall, the behaviour encapsulated in (8) consists of turning to the opposite direction where the obstacle is with respect to the system's heading or velocity vector $\dot{\mathbf{x}}$. Although this reactive motion might be the safest behaviour in front of an obstacle, there are many situations where guiding the system towards a particular route might be of interest, such as in constrained environments or when aiming for a trajectory providing a minimum cost.

Given the influence of the system's heading $\dot{\mathbf{x}}$ on the overall obstacle avoidance reaction, it is natural to modulate $\dot{\mathbf{x}}$ to guide the reactivity of (8) through a preferred route. Within the DMP motion descriptor, this can be formulated through a coupling term that creates an attractive forcing term to reduce the heading error $\hat{\theta}$ between the current $\dot{\mathbf{x}}$ and a desired $\dot{\mathbf{x}}_d$ system's direction as:

$$C_{\mathrm{HG}}(\cdot) = \mathbf{R}'\,\dot{\mathbf{x}}\,\alpha\,\hat{\theta}\,\exp\!\left(1 + \kappa\,d^2\right) \quad (9)$$

where $\mathbf{R}' \in SO(3)$ is a $\pi/2$ rotation matrix around the vector $\mathbf{r}' = \dot{\mathbf{x}} \times \dot{\mathbf{x}}_d$, and the term $\alpha\exp\!\left(1 + \kappa\,d^2\right)$ ensures that (8) and (9) act in counterphase when parameterised for the same α and κ. This is, (9) uniquely modifies the system's heading when not in proximity to obstacles, where (8) takes over the control to ensure the system's safety.

C. Coupling Terms Composition

Figure 4 depicts the significance of using (8) in conjunction with (9) to perform route selection of obstacle avoidance. This is formalised within the DMP in (1)–(2) as the composition of
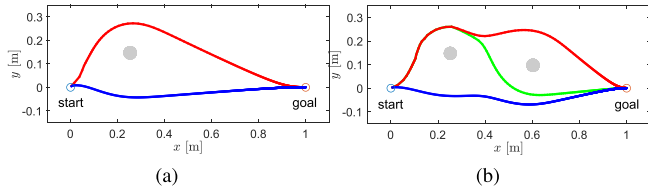
Fig. 4. Route selection for obstacle avoidance in (a) single and (b) multi-object setups. Reactive avoidance (blue), one-time decision (red), two-times decision (green).

coupling terms $C(\cdot) = \sum_i C_{\text{OA}}^i(\cdot) + C_{\text{HG}}^i(\cdot)$, where $C_{\text{OA}}^i(\cdot)$ and $C_{\text{HG}}^i(\cdot)$ generate the corresponding forcing terms with respect to the ith obstacle in the scenario. This composition allows, in a single-obstacle scenario (see Figure 4a), to guide the reactive behaviour (blue trajectory) in a different direction (red trajectory) by temporarily defining the initial desired heading towards the upper part of the task space. The same applies to multi-obstacle environments (see Figure 4b), where the system's heading can be modified at multiple decision points to obtain a preferred route (green trajectory). In both scenarios, the actuation scope of the coupling term for guiding the system was set manually for illustration purposes. Alternatively, these decision points could be defined by a task-dependant module.

D. Proof of Lyapunov's Stability

The addition of coupling terms can imperil the inherent stability properties of DMPs [2]. Authors in [9] proved with Lyapunov's theory that the overall dynamical system remains stable when the coupling terms generate a forcing term orthogonal to the system's velocity vector. The coupling terms formulated in (8) and (9) satisfy this condition, therefore proving the global stability of the proposed action level.

IV. LEARNING OBSTACLE AVOIDANCE FOR NON-POINT OBJECTS

The set of coupling terms formalised in the previous section efficiently generates guided collision-free trajectories for point-mass objects, i.e. obstacles and systems. Nonetheless, objects in real-world scenarios present different shapes and sizes. This section details the encoding of objects as low-dimensional geometric descriptors, which allows for (i) the design of a learning module that regulates the action level to generalise over different obstacle geometries while considering the system's geometry, and (ii) the use of heuristics to rapidly perform route selection in constrained environments.

A. Superquadrics as Geometric Approximates

Objects obstructing the execution of a policy might present different shapes and dimensions. This geometric diversity complicates the design of an intelligent module able to generalise obstacle avoidance behaviours across geometries [10]. This work considers global features to approximate the geometric properties of an object. One possible encoding strategy are superquadrics [11], which have been used, among others, to ease the computation of system-obstacle distances [12], and to generate repulsive potential fields [13]. Alternatively to these task space applications, this work is interested in the low-dimensional
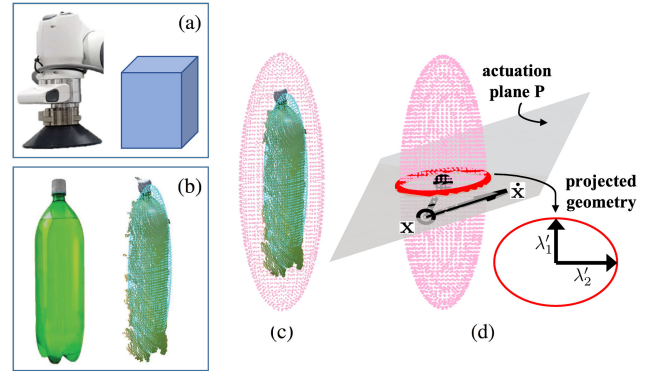
Fig. 5. Extraction of unified low-dimensional descriptors accounting for the (a) end-effector's and (b) obstacle's geometry. (c) Ellipsoid (rose) fitting the dilated obstacle cloud. (d) Relevant descriptor $\boldsymbol{\lambda}'$ along the P-plane (red ellipse).

parametric encoding of such geometric approximate, which is defined as:

$$F(x, y, z, \boldsymbol{\lambda}) : \left(\left(\frac{x}{\lambda_1}\right)^{\frac{2}{\lambda_5}} + \left(\frac{y}{\lambda_2}\right)^{\frac{2}{\lambda_5}} \right)^{\frac{\lambda_5}{\lambda_4}} + \left(\frac{z}{\lambda_3}\right)^{\frac{2}{\lambda_4}}, \quad (10)$$

where $F(\cdot)$ defines whether a given 3D point (x, y, z) lies inside ($F < 1$), outside ($F > 1$), or on the surface ($F = 1$) of a superquadric described by $\boldsymbol{\lambda} = [\lambda_1, \dots, \lambda_5]$. In particular, $(\lambda_1, \lambda_2, \lambda_3)$ set the superquadric semi-axes lengths, and (λ_4, λ_5) parameters define the superquadric shape.

The parameter vector $\boldsymbol{\lambda}$ can be estimated from a discrete representation of the obstacle's surface by minimisation of:

$$\min_{\boldsymbol{\lambda}} \sum_{i=1}^{N} \left(\sqrt{\lambda_1 \lambda_2 \lambda_3} \left(F(x_i, y_i, z_i, \boldsymbol{\lambda}) - 1\right) \right), \quad (11)$$

where $\sqrt{\lambda_1 \lambda_2 \lambda_3}$ penalises the fitting of large superquadrics.

B. Unified Low-Dimensional Geometric Descriptors

The process in (10)–(11) provides a geometrical descriptor $\boldsymbol{\lambda}$ from a discrete representation of an object. However, it is of interest to obtain a descriptor accounting for both the system's and obstacle's geometry. Figure 5 schematises the extraction of a unified obstacle-system low-dimensional geometric descriptor. An approximate of the system's geometry (see blue prism in Figure 5a) is used to dilate [14], [15] the obstacle's discrete representation (see Figure 5b). The dilated obstacle representation is then encoded using (11) while imposing $\lambda_4 = \lambda_5 = 1$, i.e. restricting the superquadric to shape as an ellipsoid. Figure 5c portrays the significance on the descriptor's difference when considering the raw obstacle representation (blue ellipsoid) and its dilated version (rose ellipsoid). Interestingly, ellipsoids hold the property that any random projection or section of these results in an ellipse, providing a strategy to extract the unified geometric features relevant to the obstacle avoidance coupling term. This is, the P-plane defined by the respective obstacle-system position $\mathbf{x}_{obstacle} - \mathbf{x}$ and the system's heading $\dot{\mathbf{x}}$, intersects the unified geometric approximation. Thus, the descriptor $\boldsymbol{\lambda}$ can be further reduced to $\boldsymbol{\lambda}' = (\lambda_1', \lambda_2') \in \mathbb{R}^2$ such that $\boldsymbol{\lambda}' = g(\boldsymbol{\lambda})$ where $g(\cdot) : \mathbb{R}^5 \to \mathbb{R}^2$ maps an arbitrary vector onto the P-plane. The resulting low-dimensional descriptor is

an ellipse laying on the P-plane with semi-axis lengths $\boldsymbol{\lambda}'$ (see Figure 5d).

C. Geometry-Conditioned Parameter Regressor

Leveraging the unified low-dimensional descriptor $\boldsymbol{\lambda}'$ from Section IV-B, this section proposes a method to learn the correspondence between $\boldsymbol{\lambda}'$ and the non-independent parameters (α, ψ, κ) of the coupling term, subject to a user-defined clearance Δ, i.e. the minimum distance between the end-effector and the obstacle. This multiple target regression problem is formulated as a regressor chain (RC) [16], which defines an ordered chain $U = (Y_1, Y_2, Y_3)$ of single target regressions. This is, given an input vector $\mathbf{h} = \{\boldsymbol{\lambda}', \Delta\}$, the proposed RC-based learning module is composed of three models: $Y_1 \colon \mathbf{h} \to \kappa$ adjusts the actuation span of the coupling term, $Y_2 \colon (\mathbf{h}, \kappa) \to \psi$ regulates the relevance of the relative system-obstacle heading, and finally $Y_3 \colon (\mathbf{h}, \kappa, \psi) \to \alpha$ tunes the strength of the behaviour. Each regressor Y_i is modelled as a NN which provides a powerful strategy to learn and represent approximations to non-linear mappings, and is suitable for reactive decisions due to its rapid response. Considering the relevance of the input features, each NN regressor is arranged with four layers; the hidden layers are hyperbolic tangent sigmoid units, and the output layer is a log-sigmoid to avoid negative settings of the targets.

It should be noted that the regulation of the action level formalised in Section III is conducted along the P-plane. As explained previously in Section IV-A, this sub-space contains all essential information to circumnavigate an obstacle and is efficiently defined using the relative system-obstacle state. Namely, changes in the obstacle avoidance scene such as different start and goal positions, obstacle location and geometries do not alter the encoding of the problem in the P-plane. Therefore, the prediction capabilities of the designed RC-based learning module extend to a wide range of setups, including in the presence of multiple obstacles in the scene.

D. Route Selection via Heuristic Cost Rings

Real-world environments and physical systems constrain the amount of feasible reactive behaviours. Exhaustively evaluating all possible directions in $\mathrm{SO}(3)$ which satisfy these additional constraints can slow the decision response. To ease the reasoning complexity of the route selection problem, this work proposes a twofold heuristic analysis called cost rings which (i) considers an orthographic projection of the obstacle onto the YZ-plane $\in \mathbb{R}^2$ of the local frame, i.e. confining the direction space $\boldsymbol{\omega} \in \mathrm{SO}(2)$, to then efficiently (ii) find the obstacle avoidance direction ω_d minimising a metric $\eta(\boldsymbol{\omega})$. The resulting direction ω_d is used with the coupling terms composition formulated in Section III-C to guide the obstacle avoidance behaviour towards ω_d.

The advantage of route selection via heuristic cost rings is exemplified in Figure 6, where the path cost $\eta(\boldsymbol{\omega})$ is determined according to three metrics: (i) the physical constraints imposed by the table $\eta_{table}(\boldsymbol{\omega})$, (ii) the length of the trajectory $\eta_{length}(\boldsymbol{\omega})$, and (iii) the robot's workspace limit $\eta_{limits}(\boldsymbol{\omega})$, such that ω_d can be found by minimisation of:

$$\min_{\boldsymbol{\omega}} \eta_{table}(\boldsymbol{\omega}) + \eta_{length}(\boldsymbol{\omega}) + \eta_{limits}(\boldsymbol{\omega}), \qquad (12)$$

where $\eta_{table}(\boldsymbol{\omega}) = 1$ if the end-effector would collide with the table and 0 otherwise, $\eta_{length}(\boldsymbol{\omega}) \in [0, 1]$ is the normalised
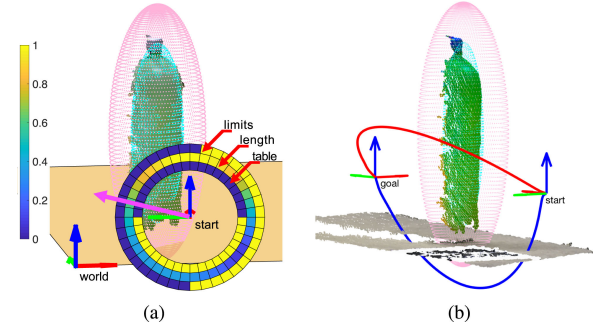
Fig. 6.   Route selection via heuristic rings. (a) Cost evaluation on the local YZ-plane to penalise workspace limits, long trajectories, and collisions with the table. Overall best direction is marked in magenta. (b) A reactive behaviour (blue) would lead the system colliding with the table, whereas the guided behaviour (red) generates the route with lowest cost.

trajectory length, and $\eta_{limits}(\boldsymbol{\omega}) = 1$ if the end-effector would move outside of its workspace and 0 otherwise. Figure 6a illustrates these estimated costs rings and the resulting direction $\omega_d \in \boldsymbol{\omega}$ (magenta) with minimum cost. As depicted in Figure 6b, using this reasoning to initially guide the behaviour enables the system to avoid the obstacle in the direction with lowest cost (red), whereas the non-guided reactive behaviour leads with collision with the table (blue).

E. Convergence to Goal

The required path $\hat{\pi}$ to avoid obstacles may be longer than the pre-planned trajectory π, thus needing more time to finalise the encoded task. This fact is especially critical when dealing with non-point objects as failing to account for this can imperil convergence to the desired goal [10]. To address this issue, this work regulates the DMP duration by scaling $\tau = length(\hat{\pi})/length(\pi)$, i.e. an approximate of the increase of trajectory length. Here, $length(\hat{\pi})$ is estimated with linear interpolation of the finite sequence of $\mathbb{R}^3$ points $\{\mathbf{x}_s, \mathbf{x}_p^1, \ldots, \mathbf{x}_p^N, \mathbf{g}_x\}$, where $\mathbf{x}_s$ and $\mathbf{g}_x$ are the start and goal positions, and $\mathbf{x}_p^i$ is the extreme point of the ellipse encoding the $i \in [1, \ N]$ dilated obstacle's geometry along its P-plane.

V. EXPERIMENTAL EVALUATION

The proposed framework has been evaluated in simulated environments and on a physical system. This section first explains the training of the RC model via exploration of the parameter space. Thereafter, it reports the performance and generalisation capabilities of the proposed approach in familiar and novel obstacle avoidance settings. Finally, this section details the deployment of the proposed framework on an anthropomorphic Franka Emika Panda arm engaged in a start-to-goal policy in the presence of unplanned obstacles.

An extended illustration of the experimental evaluation is documented in: `https://youtu.be/lym5cCbjI3k` , and the corresponding source code can be found in: `https://github.com/ericpairet/ral_2019` .

A. Training the RC-Based Learning Module

This work has designed a RC-based learning module to regulate the action level according to a unified obstacle-system

TABLE I
PREDICTION ERROR ON EVERY SINGLE TARGET OF THE TWO MODELLED RC
ARCHITECTURES FOR THE TRAINING AND TEST DATASETS

	NMSE(Y_1)		NMSE(Y_2)		NMSE(Y_3)	
	train	test	train	test	train	test
RC($\boldsymbol{\lambda}'$)	0.539	0.543	0.802	0.802	0.893	0.897
RC($\boldsymbol{\lambda}', \Delta$)	0.251	0.253	0.244	0.243	1.6e-4	1.7e-4

descriptor $\boldsymbol{\lambda}'$ and a possible clearance constraint Δ. The un-constrained model is denoted as RC($\boldsymbol{\lambda}'$), while the constrained model is referred to as RC($\boldsymbol{\lambda}', \Delta$). The training of these models is conducted leveraging the knowledge of the action level to create a synthetic dataset via exploration of the parameter space. This is, given different obstacle avoidance scenarios, training explores the parameters $\{\alpha, \psi, \kappa\}$ of the coupling term (8) generating a collision-free trajectory.

Bearing in mind that the learning module uniquely regulates the action level along its plane of actuation, 100 synthetic scenarios were created to simulate possible intersections between a unified system-obstacle ellipsoid approximation and the actuation plane P $\in \mathbb{R}^2$. This resulted in 100 ellipses parameterised with semi-axis values $\boldsymbol{\lambda}' = (\lambda'_1, \lambda'_2)$ uniformly sampled in the range 2.5 to 25 cm. Each of these sections was placed in the middle of a one-metre length start-goal baseline. For each scenario, a set of trajectories were generated using (8) with a $50 \times 50 \times 50$ grid of the parameters $\{\alpha, \psi, \kappa\}$. Only those input-target $\{(\lambda'_1, \lambda'_2), (\alpha, \psi, \kappa)\}$ pairs involving a collision-free trajectory were integrated into the dataset along with the resulting clearance.

The RC architectures were trained using a 70% of the synthetic dataset. Each NN was trained independently using the Levenberg-Marquardt algorithm with a random initialisation of the weights and biases. The remaining 30% of the dataset was used to test the performance of the trained RC models. Since the aim of a RC model is to reduce the prediction error on every single target [16], each model Y_i was validated by computing the normalised mean squared error (NMSE) on the training and testing sets. As shown in Table I, the parameter prediction error of the models reduces significantly when considering the clearance in the input vector $\mathbf{h}$. This is because the clearance allows differentiating the influence of the targets among all possible collision-free trajectories. It is worth noting that the performance of the RC does not deteriorate when being evaluated on the test set.

B. Experiments on Familiar Scenarios

The performance of both RC($\boldsymbol{\lambda}'$) and RC($\boldsymbol{\lambda}', \Delta$) models in the P-plane space was evaluated for the same obstacle geometries as in the training dataset, i.e. 100 ellipses. For the RC($\boldsymbol{\lambda}', \Delta$) model, the considered constraints on the clearance were $\Delta = \{0.05, 0.1, 0.15, 0.2, 0.25\}$ metres. All six models were evaluated with and without scaling the trajectory duration τ according to its estimated length as explained in Section IV-E. Overall, this led to the testing of the RC architecture under 12 different settings. Performance in the P-plane space was evaluated for the metrics (i) number of collisions, (ii) minimum distance to an obstacle (clearance), and (iii) distance to goal (convergence). The obtained results over the 1,200 scenarios are illustrated in Figure 7.
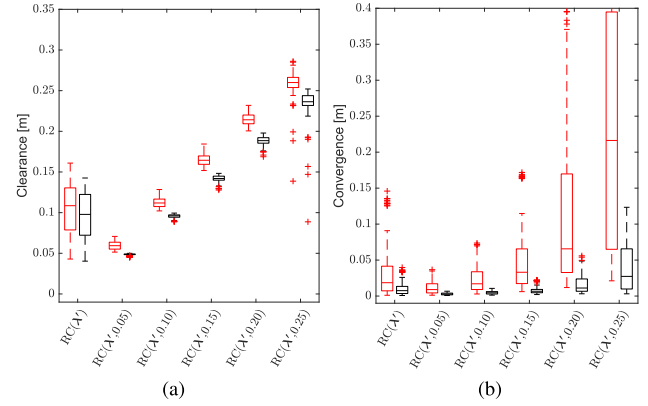
Fig. 7. (a) Clearance and (b) convergence of the avoidance behaviours generated in familiar scenarios, when scaling the trajectory duration (black plots) and when not (red plots).

TABLE II
CLEARANCE, CONVERGENCE AND NUMBER OF COLLISIONS OF THE TRAINED
RC($\boldsymbol{\lambda}', 0.15$) MODEL FOR 4,000 NOVEL SETTINGS

	Clearance to obstacle [m]		Convergence to goal [m]		n° of collisions
	mean	min	mean	max	
Goal at 0.5m	0.144	-5.01e-4	4.41e-4	0.017	2
Goal at 1.0m	0.184	0.060	4.23e-4	0.017	0
Goal at 1.5m	0.196	0.068	5.22e-4	0.023	0
Goal at 2.0m	0.202	0.076	6.49e-4	0.027	0
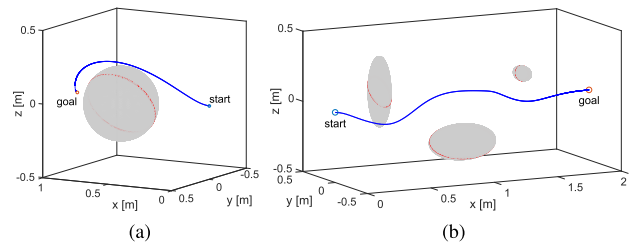
Fig. 8. Generalisation capabilities of the trained RC($\boldsymbol{\lambda}', 0.15$) model in novel settings. Parameters $\boldsymbol{\lambda}'$ are extracted from the relevant section (red ellipse) where the coupling term acts.

Figure 7a and Figure 7b respectively represent the clearance to the obstacle and convergence to the goal for the 1,200 scenarios evaluated across the 12 settings of the RC architecture. Overall, constraining the model with a desired clearance leads to more bounded behaviours. However, as the clearance constraint increases, the convergence rapidly deteriorates for those models not scaling the trajectory duration (red boxes). Instead, when scaling the time (black boxes), the convergence is at most of 3 cm for the most constrained model RC($\boldsymbol{\lambda}', 0.25$). This fact highlights the importance of scaling the time when larger trajectories are required. Indifferently from the model setup, none of the 1,200 conducted tests resulted with a trajectory colliding with an obstacle. The remainder of the experimental evaluation is conducted with the RC($\boldsymbol{\lambda}', 0.15$) model and scaling the trajectory duration according to its estimated length.
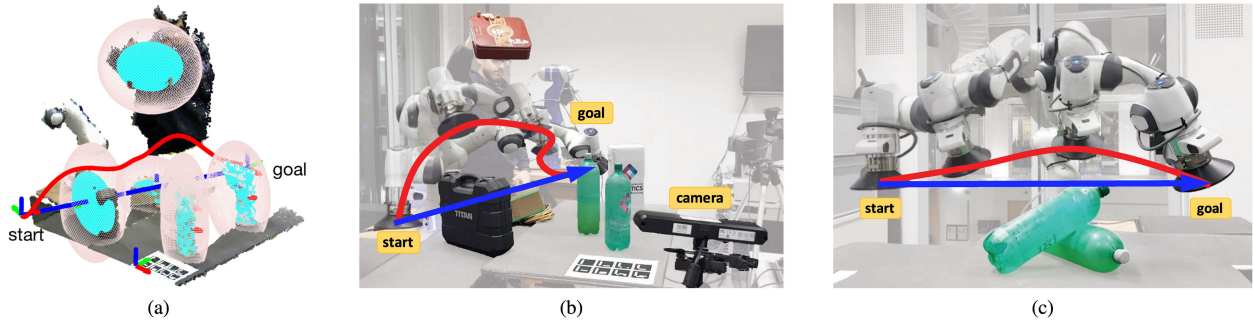
Fig. 9.   Panda arm engaged in a start-to-goal policy (blue trajectories) while modulating its behaviour (red trajectories). (a) Environment perception with unified low-dimensional encoding of the system's and obstacle's geometry (rose ellipsoids). Proposed hierarchical framework dealing with (b) multiple obstacles in a cluttered environment, and (c) an irregular obstacle.

C. Experiments on Novel Scenarios

Given the variety of obstacle avoidance scenarios that a system may face in the real-world, the proposed $RC(\lambda', 0.15)$ model was evaluated for its performance and generalisation capabilities on scenarios not seen during the training process. Notably, the approach was tested for its suitability to deal with three-dimensional (3D) obstacles via the extraction of relevant unified low-dimensional geometrical features laying on the P-plane as described in Section IV-A.

Novel 3D scenarios were created by sampling the location and dilated geometry of the obstacle randomly. The obstacle was arbitrarily located along the x-axis between the start and goal configurations preserving 5 cm of margin, and around the baseline between -0.4 and 0.4 m along both the y-axis and z-axis. The unified system-obstacle ellipsoid approximation had random width, height and length within the spectrum 5 to 50 cm, leading to representative candidates of possible object geometries in real-world environments. This spectrum corresponds to semi-axis values λ_1, λ_2 and λ_3 laying in the range 2.5 to 25 cm. These boundaries also ensured that none of the extracted low-dimensional features λ' would result beyond the limits for which the RC model was trained for.

A 1,000 novel 3D scenarios were created for the different start-to-goal baselines of 0.5, 1.0, 1.5 and 2.0m along the x-axis of the local frame, adding up to a total of $4,000$ evaluations. The semi-axis λ_1 was limited to a maximum of 20 cm for the baseline of 0.5 m to be consistent with the 5cm margin across experiments. All environments required the action level to modulate a start-to-goal policy to avoid collision and preserve the desired clearance. Out of the 4,000 tests, 1,296 environments already had the baseline in collision with the obstacle. The performance of $RC(\lambda', 0.15)$ on the unseen settings was evaluated for the metrics (i) number of collisions, (ii) clearance to an obstacle, and (iii) convergence to goal. Table II summarises the extracted metrics across the evaluation, and Figure 8 depicts the performance of the proposal on some novel single and multi-obstacle settings.

Results in Table II reflect the performance of the designed $RC(\lambda', 0.15)$ model when dealing with 3D scenarios via their section on the P-plane. The overall success rate is of 99.95% on novel scenarios while providing, in average, a clearance similar to the requested one of 0.15m and a close convergence to the goal. This implies an enhancement of 31.75 times over the success rate reported on known objects in [10]. However,

the performance of the approach is slightly compromised in some scenarios, obtaining clearances of 6cm and convergences up to 2.7cm. The proposed approach could not cope uniquely with two scenarios out of $4,000$, where the generated trajectory penetrated 0.501 mm an obstacle of 40cm along the x-axis and 50cm along the y-axis and z-axis placed in the middle of a 0.5m long baseline. Albeit these extreme scenarios for which more data could be provided at training time, the proposed approach has proved to generalise not only to different object sizes and locations, but also to different start-to-goal baselines. Further experimentation also showed the suitability of the framework to deal with multi-obstacle scenarios (see Figure 8b). Since the action level is referenced in a local frame (see Section II-A), the performance of the framework does not deteriorate regardless of the local frame's pose in the task space. Within the local frame, the outstanding generalisation capabilities are mainly due to regulating action according to the relative system-obstacle state defining the P-plane, and extracting relevant system-obstacle low-dimensional geometrical descriptors.

D. Experiments on a Robotic Platform

The proposed hierarchical framework for obstacle avoidance has been deployed on an anthropomorphic 7-DoF Franka Emika Panda arm operated with OROCOS [17]. The DMP-encoded system's transient behaviour is converted to joint configurations using a Cartesian inverse dynamic controller with null space optimisation. The environment is partially observed with a depth camera ASUS Xtion previously calibrated with Aruco markers [18]. The acquired point cloud is processed applying standard filtering techniques to segment the clusters describing obstacles and the table. The partial observation of each obstacle is dilated to also account for the system's geometry (see Section IV-B). The location of the table is used to constrain the reactive behaviour along the upper part of the task space (see Section IV-D).

As in [9], [10], the test-bed consisted of obstacles interrupting a straight trajectory underlying a start-to-goal policy. However, differently than [9], [10], the assortment of considered obstacles had not been seen before. This included, but was not limited to, regular objects, such as the cardboard box in Figure 1, irregular objects, such as the pile of plastic bottles in Figure 9c, and also aleatory combinations of them, such as the cluttered environment with six obstacles in Figure 9a and Figure 9b. As summarised in Table III, the robot engaged in the pre-planned

TABLE III
CLEARANCE, CONVERGENCE, NUMBER OF COLLISIONS AND UNSATISFIED
CLEARANCES FOR THE PRE-PLANNED (PP.) AND MODULATED (MOD.)
START-TO-GOAL POLICIES ON THE REAL ROBOT

		Clearance to obstacle [m]	Convergence to goal [m]	$n°$ collisions & $\Delta < 0.15m$
regular	pp.	fail	fail	1 & 1
(Fig. 1)	mod.	0.168	9.58e-3	0 & 0
irregular	pp.	fail	fail	1 & 1
(Fig. 9c)	mod.	0.163	4.17e-3	0 & 0
multi-obs	pp.	fail	fail	2 & 5
(Fig. 9b)	mod.	0.172	1.29e-2	0 & 0

policy (blue trajectories) would impact with the obstacles. Instead, endowing the robot with the ability to modulate such policy, allows the system to successfully circumvent all obstacles with the desired 0.15m clearance while converging to the goal (red trajectories).

The presented results demonstrate that the proposed hierarchical framework which (i) extracts relevant geometric descriptors, (ii) uses them in the designed RC-based learning module to (iii) regulate the DMP-based action level, endows a system with the ability to modulate its behaviour in settings never seen before, while stably converging to the goal.

VI. FINAL REMARKS

This letter has presented a biologically-inspired hierarchical framework which safely modulates an on-going policy to avoid obstacles. The proposed approach follows a multi-layered perception-decision-action analysis which (i) extracts unified system-obstacle low-dimensional geometric descriptors, then (ii) exploits them to rapidly reason about the environment with a combination of heuristics and learning techniques, and finally (iii) guides and regulates the obstacle avoidance behaviour with a conjunction of coupling terms modulating a DMP-encoded policy. Experimentation conducted in synthetic environments highlights this method's generalisation capabilities to confront novel scenarios at the same time of ensuring the convergence of the system to the goal. Additionally, real-world trials on an anthropomorphic manipulator demonstrated the framework's suitability to successfully modulate a policy in the presence of multiple novel obstacles described by partial visual-depth observations, while satisfying a user-defined clearance constraint.

The proposed framework is not restricted to the presented experimental evaluation nor platform. Any robotic system following a DMP-encoded policy can benefit from this work to safely modulate its behaviour in the presence of unexpected obstacles. Similarly to [7], collisions of the links can also be considered by finding the closest geometric section on the robot to the obstacle, and then modulating the kinematic null-space movement with the proposed approach. An interesting venue for future work

is to modulate the system's orientation policy to overcome an obstacle, which, for instance, might have a significant impact on a manipulator carrying a large bulk. Another interesting extension of this work is learning route selection priorities in cluttered environments, so systems can autonomously reason about the most convenient direction to avoid an obstacle.

REFERENCES

[1] B. R. Fajen and W. H. Warren, "Behavioral dynamics of steering, obstacle avoidance, and route selection," *J. Exp. Psychol., Human Perception Perform.*, vol. 29, no. 2, pp. 343–362, 2003.

[2] A. J. Ijspeert, J. Nakanishi, H. Hoffmann, P. Pastor, and S. Schaal, "Dynamical movement primitives: learning attractor models for motor behaviors," *Neural Comput.*, vol. 25, no. 2, pp. 328–373, 2013.

[3] A. Gams, A. J. Ijspeert, S. Schaal, and J. Lenarčič, "On-line learning and modulation of periodic movements with nonlinear dynamical systems," *Auton. Robots*, vol. 27, no. 1, pp. 3–23, 2009.

[4] A. Gams, B. Nemec, A. J. Ijspeert, and A. Ude, "Coupling movement primitives: Interaction with the environment and bimanual tasks," *IEEE Trans. Robot.*, vol. 30, no. 4, pp. 816–830, Aug. 2014.

[5] G. Sutanto, Z. Su, S. Schaal, and F. Meier, "Learning sensor feedback models from demonstrations via phase-modulated neural networks," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2018, pp. 1142–1149.

[6] È. Pairet, P. Ardón, F. Broz, M. Mistry, and Y. Petillot, "Learning and generalisation of primitives skills towards robust dual-arm manipulation," in *Proc. AAAI Fall Symp. Reasoning Learn. Real-World Syst. Long-Term Auton.*, 2018, pp. 62–69.

[7] D.-H. Park, H. Hoffmann, P. Pastor, and S. Schaal, "Movement reproduction and obstacle avoidance with dynamic movement primitives and potential fields," in *Proc. IEEE-RAS Int. Conf. Humanoid Robots*, 2008, pp. 91–98.

[8] H. Hoffmann, P. Pastor, D.-H. Park, and S. Schaal, "Biologically-inspired dynamical systems for movement generation: Automatic real-time goal adaptation and obstacle avoidance," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2009, pp. 2587–2592.

[9] A. Rai, F. Meier, A. Ijspeert, and S. Schaal, "Learning coupling terms for obstacle avoidance," in *Proc. IEEE-RAS Int. Conf. Humanoid Robots*, 2014, pp. 512–518.

[10] A. Rai, G. Sutanto, S. Schaal, and F. Meier, "Learning feedback terms for reactive planning and control," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2017, pp. 2184–2191.

[11] A. Barr, "Superquadrics and angle-preserving transformations," *IEEE Comput. Graph. Appl.*, vol. 1, no. 1, pp. 11–23, Jan. 1981.

[12] V. Perdereau, C. Passi, and M. Drouin, "Real-time control of redundant robotic manipulators for mobile robot obstacle avoidance," *Robot. Auton. Syst.*, vol. 41, no. 1, pp. 41–59, 2002.

[13] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," in *Autonomous Robot Vehicles*. New York, NY, USA: Springer, 1986, pp. 396–404.

[14] T. Lozano-Perez, "Spatial planning: A configuration space approach," in *Autonomous Robot Vehicles*. New York, NY, USA: Springer, 1990, pp. 259–271.

[15] L. Huber, A. Billard, and J.-J. Slotine, "Avoidance of convex and concave obstacles with convergence ensured through contraction," *Robot. Autom. Lett.*, vol. 4, no. 2, pp. 1462–1469, 2019.

[16] E. Spyromitros-Xioufis, G. Tsoumakas, W. Groves, and I. Vlahavas, "Multi-target regression via input space expansion: Treating targets as inputs," *Mach. Learn.*, vol. 104, no. 1, pp. 55–98, 2016.

[17] H. Bruyninckx, "Open robot control software: The Orocos project," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2001, vol. 3, pp. 2523–2528.

[18] S. Garrido-Jurado, R. Muñoz-Salinas, F. J. Madrid-Cuevas, and M. J. Marín-Jiménez, "Automatic generation and detection of highly reliable fiducial markers under occlusion," *Pattern Recognit.*, vol. 47, no. 6, pp. 2280–2292, 2014.

8.2   Supplementary Material

In this section, we provide some in-detail discussion about the novelty of the proposed approach in comparison to related methods in the literature (see Section 8.2.1), as well as prove the overall stability of our approach (see Section 8.2.2).

8.2.1   Comparison to Previous DMP-based Obstacle Avoidance Methods

To better understand the contribution of the presented work, we first briefly revisit the previous DMP-based obstacle avoidance methods in the literature [130, 82, 61, 142, 143], and then we provide a theoretical and experimental comparison among approaches in Table 8.1.

The first DMP-based obstacle avoidance method in the literature formulated a coupling term as a dynamic potential field, and controlled the kinematic null-space of the robot to also avoid obstacles with the links of a robotic arm [130]. Importantly, their approach to consider the manipulator's links can be integrated in any of the coupling terms presented in the literature, including our method. All posterior works, however, have just focused on the end-effector. The local minima issue and the point-based assumption in [130] were tackled by authors in [82], by extracting the obstacles' bounding volume to shape the harmonic potential fields driving the system. Heuristics were considered to overcome local minima in certain situations, yet without guarantees. Another common concern of potential fields is that they require fine parametrisation.

To overcome the main limitations of potential field-based approaches, authors in [61] presented an analytical form of a repulsive term in the heading space for point-based obstacles and point-based systems. To address the point-based obstacle assumption while preserving the low computational burden of such analytical formulation, recent works have leveraged from a pre-processing stage. Authors in [142] learnt from human demonstrations the required coupling term to avoid two different obstacles (a sphere and a cylinder). The generalisation capabilities of this approach were not reported, and its applicability was not proven on a real platform. Similarly, authors in [143] used human demonstrations to build independent NN-based coupling terms for each of the three considered obstacles (a sphere, a cylinder and a cube). This work was only tested in single obstacle scenarios, and similarly to the precedent approach, the evaluation was limited to the obstacles used in the training set. A significant drawback of these methods is their dependency on a huge set of high-quality data; a marker-based motion capture system was used to accurately extract human-demonstrated motions (human-hand as point-mass system) and identify the obstacles' shape. More importantly, these approaches do not demonstrate generalisation to novel obstacle geometries.

Differently from previous approaches, the method we present is not restricted to the obstacles observed in the training set because we approximate both the obstacle and the end-effector shape as a low-dimensional geometric descriptor. We apply dilation [105, 66] to project the end-effector geometric descriptor onto the obstacles' descriptors. These dilated descriptors are used to infer suitable action modulations with a learning-based technique, particularly a RC of NNs. We ease the training of this NN-based RC by exploring the parameter space of the coupling term, thus avoiding dependency on the time-consuming process of gathering quality data from human demonstrations. Additionally, we propose a heuristic optimisation process to find the preferred direction to avoid constraints violation. We evaluate our overall approach in novel scenarios, i.e., obstacle's locations and geometries not observed at training time, in both single and multi-obstacle setups, and in both simulated and real-world experiments.

Table 8.1 provides a theoretical and experimental comparison among methods. From the theoretical side, the analysed features are: (i) the type of coupling term, (ii) if they suffer from local minima (LM) or dead-zone (DZ), (iii) the geometrical representation of the obstacle (OBS) and end-effector (EE), and (iv) the relevant computation requirements, pre-execution (PE) and during execution (DE). From the experimental side, most of these works do not provide any source code, training data, nor trained models. In fact, the most recent and related approaches [142, 143] do not have any online resource. As previously discussed, these approaches depend upon a great amount of quality data (they employed a total of 1,900 human demonstrations) and a marker-based identification of the obstacle geometry (yet limited to three known obstacles). On top of that, one of these approaches has

	THEORETICAL DETAILS				EXPERIMENTAL DETAILS				
	(i)	(ii)	(iii)	(iv)	(v)				(vi)
	coupling	LM/DZ	OBS/EE	PE/DE	experiments				failure
	term		geometry	computation	SS	MS	SR	MR	rate
[130]	dynamic PF	LM	point/point	none/PF	✓		✓		NR
[82]	harmonic PF	LM[2]	BV/point	none/PF	✓	✓	✓	✓	NR
[61]	analytic	DZ	point/point	none/none	✓	✓	✓		NR
[142]	analytic	DZ	marker/point	LbD/none	✓	✓			NR[4]
[143]	NN	none[3]	marker/point	LbD/none	✓		✓		3/189[4]
This work	hybrid[1]	none	SQ/SQ	FS/none	✓	✓	✓	✓	2/4000

Table 8.1: Comparison of the proposed approach with previous DMP-based obstacle avoidance methods. PF: potential field. BV: boundary volume. SQ: superquadric. LbD: learning by demonstration. FS: forward simulation of the parameter space. NR: not reported. [1]analytical + NN + heuristics to satisfy extra constraints. [2]overcomes LM in certain conditions via heuristics. [3]subject to the demonstrations. [4]Uniquely evaluated with, at most, three known obstacles.

not been even tested on a real platform. Consequently, taking everything into account, reproducing their approaches for a thorough benchmark pointed out to be an unfeasible, cumbersome task. Alternatively, we take as a comparison point the reported experimental evaluation, namely: (v) the documented experiments (single obstacle in simulation (SS), multiple obstacles in simulation (MS), single obstacle in real platform (SR), multiple obstacles in real platform (MR) and (vi) the reported failure rate.

It is worth noticing that the latest work in the literature [143], despite limiting the evaluation of their method to generalise to different obstacle locations with the three obstacles observed in the training set, reported a failure rate of 3/189. This failure rate is 31.75 times higher than the one we achieved (2/4000) with our overall hierarchical framework when testing it on novel scenarios, i.e., not only different locations but also different obstacles geometries. Moreover, our approach has been proven to work in real-world scenarios, for novel obstacle geometries and locations, and for single and multi-obstacle setups.

8.2.2   Proof of Lyapunov's Stability

The overall stability of the second-order time-invariant linear system driven by a forcing term and several coupling terms (see (1)-(2)) is guaranteed as far as each of the terms fulfils the stability criterion [104, 133, 69]. In this regard, stability proofs of DMP-based dynamical system under the effect of both discrete and rhythmic forcing terms $f(\cdot)$ was presented in [69], while its stability when applying coupling terms $C(\cdot)$ was proved with the Lyapunov stability theorem in [142]. As shown next, the latter proof applies to both presented coupling terms in (8) and (9).

A DMP-based dynamical system in $\mathbb{R}^3$ driven by a forcing term $f(\cdot)$ and the two proposed coupling terms $C_{\mathrm{OA}}(\cdot)$ and $C_{\mathrm{HG}}(\cdot)$ can be written in a spring-mass-damper form as:

$$\ddot{\mathbf{x}} = \mathbf{K}_x(\mathbf{g} - \mathbf{x}) - \mathbf{D}_x\dot{\mathbf{x}} + f(\cdot) + C_{\mathrm{OA}}(\cdot) + C_{\mathrm{HG}}(\cdot), \tag{8.1}$$

where as the system approaches the equilibrium point, i.e., $t \to \infty$, the phase term $k \to 0$, thus reducing the magnitude of the forcing term $f(\cdot) \to 0$ (see (3) and (5) in the paper):

$$\ddot{\mathbf{x}} = \mathbf{K}_x(\mathbf{g} - \mathbf{x}) - \mathbf{D}_x\dot{\mathbf{x}} + C_{\mathrm{OA}}(\cdot) + C_{\mathrm{HG}}(\cdot). \tag{8.2}$$

The equilibrium point for this system is $(\mathbf{x}, \dot{\mathbf{x}}, \ddot{\mathbf{x}}) = (\mathbf{g}, \mathbf{0}, \mathbf{0})$. To prove the stability of the dynamical system, a Lyapunov function is formulated according the energy function of the mass-spring-damper system:

$$V(\mathbf{x}, \dot{\mathbf{x}}, \ddot{\mathbf{x}}) = \frac{1}{2}(\mathbf{g} - \mathbf{x})^T\mathbf{K}_x(\mathbf{g} - \mathbf{x}) + \frac{1}{2}\dot{\mathbf{x}}^T\dot{\mathbf{x}}. \tag{8.3}$$

According to the Lyapunov's stability theorem, the system is guaranteed to be stable if (i) $V(\mathbf{x}, \dot{\mathbf{x}}, \ddot{\mathbf{x}}) > 0$ for any state, (ii) $V(\mathbf{x}, \dot{\mathbf{x}}, \ddot{\mathbf{x}}) = 0$ at the equilibrium point, i.e., $(\mathbf{x}, \dot{\mathbf{x}}, \ddot{\mathbf{x}}) = (\mathbf{g}, \mathbf{0}, \mathbf{0})$, and (iii) $\dot{V}(\mathbf{x}, \dot{\mathbf{x}}, \ddot{\mathbf{x}}) < 0$ for any state. While it is straightforward to see that the formulated Lyapunov function satisfies the first two requirements, proving the latter requires deriving $V(\mathbf{x}, \dot{\mathbf{x}}, \ddot{\mathbf{x}})$ and substituting for the system at the equilibrium point (8.2):

$$
\begin{aligned}
\dot{V}(\mathbf{x}, \dot{\mathbf{x}}, \ddot{\mathbf{x}}) &= \frac{\partial}{\partial \mathbf{x}} V(\mathbf{x}, \dot{\mathbf{x}}, \ddot{\mathbf{x}})^T \dot{\mathbf{x}} + \frac{\partial}{\partial \dot{\mathbf{x}}} V(\mathbf{x}, \dot{\mathbf{x}}, \ddot{\mathbf{x}})^T \ddot{\mathbf{x}}, \\
&= -(\mathbf{g} - \mathbf{x})^T \mathbf{K}_x \dot{\mathbf{x}} + \dot{\mathbf{x}}^T \ddot{\mathbf{x}}, \\
&= \dot{\mathbf{x}}^T (-\mathbf{K}_x (\mathbf{g} - \mathbf{x}) + \ddot{\mathbf{x}}), \\
&= \dot{\mathbf{x}}^T (-\mathbf{D}_x \dot{\mathbf{x}} + C_{\mathrm{OA}}(\cdot) + C_{\mathrm{HG}}(\cdot)), \\
&= -\dot{\mathbf{x}}^T \mathbf{D}_x \dot{\mathbf{x}} + \dot{\mathbf{x}}^T C_{\mathrm{OA}}(\cdot) + \dot{\mathbf{x}}^T C_{\mathrm{HG}}(\cdot), \\
&= -\dot{\mathbf{x}}^T \mathbf{D}_x \dot{\mathbf{x}},
\end{aligned}
\tag{8.4}
$$

where given that $\mathbf{D}_x > 0$ and the negative quadratic nature of $\dot{V}(\mathbf{x}, \dot{\mathbf{x}}, \ddot{\mathbf{x}})$, the third Lyapunov requirement for stability is also proven. The latest step of the derivation is based on the fact that $\dot{\mathbf{x}}^T C_{\mathrm{OA}}(\cdot) = 0$ and $\dot{\mathbf{x}}^T C_{\mathrm{HG}}(\cdot) = 0$; both coupling terms are composed of $\mathbf{R}\dot{\mathbf{x}}$ (see (8) and (9)), where by definition $\mathbf{R} \in \mathrm{SO}(3)$ is a $\pi/2$ rotation matrix, thus making the proposed coupling terms orthogonal to the velocity vector $\dot{\mathbf{x}}$.

Chapter

9

Path Planning for Manipulation using Experience-driven Random Trees

In this chapter, we tackle the challenge of generalising a prior experience with the proposition that experiences are "decomposable" and "malleable", i.e., parts of an experience are suitable to relevantly explore the connectivity of the robot-task space even in non-experienced regions. Two new planners result from this insight: ERT and its bi-directional version ERTConnect. These planners adopt a tree sampling-based strategy that incrementally extracts and modulates parts of a single path experience to compose a valid motion plan. We demonstrate our method on task instances that significantly differ from the prior experiences, and compare with related state-of-the-art experience-based planners. While their repairing strategies fail to generalise priors of tens of experiences, our planner, with a single experience, significantly outperforms them in both success rate and planning time.

All proposed work is described in detail in the following published journal article:

Path Planning for Manipulation using Experience-driven Random Trees

Èric Pairet[1,2], Constantinos Chamzas[2], Yvan Petillot[1], Lydia E. Kavraki[2]

Abstract—Robotic systems may frequently come across similar manipulation planning problems that result in similar motion plans. Instead of planning each problem from scratch, it is preferable to leverage previously computed motion plans, i.e., experiences, to ease the planning. Different approaches have been proposed to exploit prior information on novel task instances. These methods, however, rely on a vast repertoire of experiences and fail when none relates closely to the current problem. Thus, an open challenge is the ability to generalise prior experiences to task instances that do not necessarily resemble the prior. This work tackles the above challenge with the proposition that experiences are "decomposable" and "malleable", i.e., parts of an experience are suitable to relevantly explore the connectivity of the robot-task space even in non-experienced regions. Two new planners result from this insight: experience-driven random trees (ERT) and its bi-directional version ERTConnect. These planners adopt a tree sampling-based strategy that incrementally extracts and modulates parts of a single path experience to compose a valid motion plan. We demonstrate our method on task instances that significantly differ from the prior experiences, and compare with related state-of-the-art experience-based planners. While their repairing strategies fail to generalise priors of tens of experiences, our planner, with a single experience, significantly outperforms them in both success rate and planning time. Our planners are implemented and freely available in the the Open Motion Planning Library.

Index Terms—Manipulation Planning; Motion and Path Planning; Learning from Experience; Autonomous Agents

I. INTRODUCTION

A long-envisioned requisite for fully-autonomous robotic manipulation is to endow robots with the ability to learn from and improve through experiences. For example, consider a robot on a shelf stacking task (see Figure 1). Such a robot may frequently come across similar task instances that result in similar motion plans. Despite the resemblance among problems, the most common approach is to plan from scratch; neither prior information nor recurrent computations are leveraged to aid in solving related queries. This strategy can lead to unnecessary long planning times. Instead, the commonalities between instantiations should be considered as prior knowledge at the planning stage. However, this is not a trivial problem. The planner must reason over the relevant features that allow for the generalisation of prior knowledge even to notably different task instances.

Fig. 1: Our planner can leverage a single path (experience) computed in a particular task instance, e.g., "fetch the red object" (left), to efficiently solve novel task instances (right) that remarkably differ from the experience, e.g., in obstacles (blue objects), shelf structural geometry and target locations.

Related work. Leveraging prior experiences for building motion plans efficiently has drawn special attention to the learning and planning communities. Learning-based approaches infer the underlying task policy from a given set of demonstrations, which is then used to retrieve task-related motion plans (e.g., [1]–[4]). Relevant features are extracted from the demonstrations such that the learnt policy can generalise to novel task instances. Although these methods are capable of computing plans quickly by learning from experience, they typically generalise poorly to task instances that significantly differ from those observed a priori [5].

On coping with varying task instances while leveraging experiences, sampling-based planning offers a promising venue to generalise the a priori knowledge. Such a strategy is known as experience-based planning. There are mainly two orthogonal approaches: (1) biasing the sampling into task-relevant areas, and (2) exploiting previously computed motions. This work is, in spirit, closer to the latter: leveraging prior motions. Related work is discussed for both alternatives.

(1) Biasing the sampling involves guiding the exploration towards task-relevant regions of the configuration space. A common approach is to take advantage of geometric features of the workspace to guide the sampling in the configuration space (e.g., [6]–[10]). Strategies that bias the sampling can significantly speed up queries, but they rely on identifying familiar workspace features to infer relevant samples in the configuration space. Therefore, their applicability is mainly limited to task instances that resemble those observed a priori, leading to a lack of generalisation to new environments.

(2) Using previously computed motions consists of storing experienced motions in a library (e.g., [11], [12]) or jointly as a graph (e.g., [13], [14]). These methods recall exact prior expe-

A visual aid about the experience-driven random trees planners can be found in: https://youtu.be/kD3A3Xs_psI.

riences to solve the current planning query. In Lightning [11], the most relevant experience is retrieved based on the start-goal proximity of a experience and the current query. The nearest path is chosen to be repaired. The repair step employs the bi-directional rapidly-exploring random tree (RRTConnect) to reconnect the end-points of segments originated by variant constraints, e.g., obstacles. Differently, Experience Graphs [13] build a roadmap of experiences to then search it using some heuristics. In a similar vein, Thunder [14] creates a sparse roadmap from all experiences, which is repeatedly queried via A^* until a valid path is found. If the graph does not contain a valid path, candidate paths, if any, are considered for repairing. The repairing invokes RRTConnect to reconnect the disconnected states along the candidate paths. All these path-centric approaches exploit prior motion plans in the exact configuration they were experienced, i.e., "rigidly". This leads to poor performance when planning in non-experienced regions of the robot-task space. Therefore, for these methods to work, the library must contain a prior path that already resembles a valid motion plan for the current planning problem. Consequently, current experienced-based planners are dependant on a vast and extremely relevant set of prior experiences to counteract their lack of generalisation capabilities.

Contribution. In this work, we change the paradigm in which prior path experiences stored in libraries of motions are being used. Instead of exploiting prior motions "rigidly" to preserve the invariant constraints, we use them in a "decomposable" and "malleable" way to infer the next move given a particular state of the robot in a task. With this proposition, we present experience-driven random trees (ERT) and its bi-directional version ERTConnect, two experience-based planners capable of generalising a single prior motion plan across significantly varied task instances. These planners leverage a path experience by parts to iteratively build a tree of micro-experiences, i.e., segments that resemble those in the prior experience. Suitable micro-experiences result from semi-randomly morphing different parts of the experience. Such a strategy proves to be useful to efficiently explore the connectivity of the robot-task space even in non-experienced regions. Additionally, we discuss how to select the best candidate for our planner given a library of path experiences. Such experience selection strategy enables the use of our planners in frameworks that incrementally build libraries of experiences by adding newly computed motion plans (e.g., [11], [14]), as well as in systems that gather experiences from human demonstrations (e.g., [15]–[17]).

The key insight of our approach is that prior experiences are of a better use when leveraged in a "malleable" fashion, oppositely to the common "rigid" usage of experiences. Thus, contrary to prior work, the applicability of our planner goes beyond task instances that closely resemble those observed a priori. Empirical analysis demonstrates our planner's ability to leverage prior experiences efficiently and to generalise them to distinguishably dissimilar task instances. In these challenging conditions, while related state-of-the-art experience-based planners fail to exploit vast repertoires of prior path experiences, our planner, with a single path experience, significantly outperforms them in both success rate and planning time.

II. PROBLEM DEFINITION AND APPROACH OVERVIEW

In this manuscript, we are interested in families of motion planning problems that involve similar task instances and thus, seek similar motion plans. The commonalities between instantiations are of interest because they open the possibility for a robot to leverage prior information about the task. Enabling the robot to exploit such similarities would allow it to efficiently solve tasks related to those seen a priori.

Consider a robot with configuration space $\mathcal{Q} \in \mathbb{R}^n$ conducting a particular task, e.g., shelf stacking (see Figure 1). Let $\mathcal{Q}_{\text{obst}} \subset \mathcal{Q}$ be the region of the configuration space occupied by obstacles, and $\mathcal{Q}_{\text{free}} = \mathcal{Q} \setminus \mathcal{Q}_{\text{obst}}$ be the collision-free region. Let $\mathbf{q} \in \mathcal{Q}$ denote a particular robot configuration, and $\alpha \in [0, 1]$ be a phase variable that indicates the progress on the execution of a collision-free motion plan. Then, the state of the robot in a motion plan is defined in the configuration-phase space $\mathcal{S} = \mathcal{Q} \times \mathbb{R}_{[0,1]}$ as $\mathbf{s} = \langle \mathbf{q}, \alpha \rangle \in \mathbb{R}^{n+1}$. The valid regions in the configuration-phase are defined as:

$$\mathcal{S}_{\text{free}} = \{\langle \mathbf{q}, \alpha \rangle \in \mathcal{S} \mid \mathbf{q} \in \mathcal{Q}_{\text{free}}\}. \tag{1}$$

Let $\mathcal{A}$ be some prior information about the task. In this work, we consider prior knowledge defined by a library of path experiences $\mathcal{A} = \{\xi_{\mathcal{D}1}, \xi_{\mathcal{D}2}, \ldots, \xi_{\mathcal{D}j}\}$, where each $\xi_{\mathcal{D}}$ is a path (prior experience) solving a particular task instance. Paths as priors are of particular interest since they can be acquired over time from the robot's planning solutions on similar task instances, or from external sources, such as from a human kinaesthetically guiding a robot through a task. Note that the focus of this manuscript is experience-based planning, where a set of prior path experiences $\mathcal{A}$ relevant to the current problem is assumed to be provided. Therefore, given a library $\mathcal{A}$, and the start $\langle \mathbf{q}_{\text{start}}, 0 \rangle \in \mathcal{S}_{\text{free}}$ and goal $\langle \mathbf{q}_{\text{goal}}, 1 \rangle \in \mathcal{S}_{\text{free}}$ states, the motion planning problem considered in this work seeks a planning process $\mathcal{J} : \mathcal{A} \to \xi$ capable to leverage $\mathcal{A}$ to efficiently find a collision-free continuous path $\xi : \alpha \in [0, 1] \to \mathcal{S}_{\text{free}}$ that connects $\xi(0) = \mathbf{q}_{\text{start}} \in \mathcal{S}_{\text{free}}$ to $\xi(1) = \mathbf{q}_{\text{goal}} \in \mathcal{S}_{\text{free}}$.

Our approach to take advantage of a library of experiences $\mathcal{J} : \mathcal{A} \to \xi$ is twofold. First, as discussed in Section IV-A, we select a path experience $\xi_{\mathcal{D}} \in \mathcal{A}$ suitable for the current planning problem. Then, we exploit the selected prior $\mathcal{L} : \xi_{\mathcal{D}} \to \xi$ via our contribution: the experience-driven random trees planners ERT and ERTConnect presented in Section III. We empirically demonstrate that, when using our planner, a unique prior path suffices to solve other instances of the same task.

III. EXPERIENCE-DRIVEN RANDOM TREES

The ERT and ERTConnect planners are inspired by tree sampling-based methods [18], [19]. Our planners, however, iteratively leverage a single task-relevant prior path experience by parts (segments, a.k.a., micro-experiences) to ease the capture of connectivity of the space. Such micro-experiences are semi-randomly morphed to generate task-relevant motions, i.e., segments that resemble those in the prior (e.g., dotted lines in Figure 2). The obtained motions are sequentially concatenated to compose a task-relevant tree (see green tree in Figure 3). This exploratory strategy aims at finding a trace

along the tree edges, i.e., a sequence of local modifications on the prior, that constitutes a continuous path ξ which satisfies $\xi : \alpha \in [0, 1] \rightarrow \mathcal{S}_{\text{free}}$, $\xi(0) = \mathbf{q}_{\text{start}}$ and $\xi(1) = \mathbf{q}_{\text{goal}}$.

Noteworthy, our planners are designed to be agnostic to distance metrics, as capturing proximity between two robot configurations in a task is not trivial. Moreover, such metric would potentially need to be designed for each task. Therefore, instead of iteratively growing a tree from the nearest configuration to a random sample (RRT-like [19]), our experience-driven random trees iteratively branch-off (expand, EST-like [18]) by concatenating the inferred motions. Likewise, to generate resembling motions, we deform the micro-experiences such that no similarity metric is needed.

The core routine through which the planner exploits the prior experience to generate task-relevant micro-experiences is detailed in Section III-A, and its usage in a uni- and bi-directional sampling-based planning strategy is presented in Section III-B and Section III-C, respectively.

A. Inferring Task-relevant Motions from a Single Experience

For planning efficiently, we are particularly interested in generating motions that are task-relevant in $\mathcal{S}$, i.e., coherent according to the robot state in a task. To that purpose, our planners leverage an experience by parts in a "malleable" fashion, as opposed to the common "rigid" usage, to infer motions that are likely to be relevant to different task instantiations.

Initially, our planners pre-process the given experience $\xi_{\mathcal{D}}$ before exploiting it iteratively. Specifically, $\xi_{\mathcal{D}}$ is mapped onto the current planning problem to obtain $\xi'_{\mathcal{D}}$, a path whose initial and final configurations match the start and goal of the current planning problem (see Figure 2). The computation of such mapping $\xi_{\mathcal{D}} \rightarrow \xi'_{\mathcal{D}}$ is detailed within the description of the planners. Then, at each iteration, our planners leverage a part (micro-experience) of the mapped experience $\xi'_{\mathcal{D}}$ to infer suitable motions for the task. Generally, let $\psi_{\mathcal{D}} : \alpha \in [\alpha_{\text{ini}}, \alpha_{\text{end}}]$ be a micro-experience from the prior spanning from α_{ini} to α_{end} such that $\psi_{\mathcal{D}}(\alpha) = \xi'_{\mathcal{D}}(\alpha) \; \forall \; \alpha \in [\alpha_{\text{ini}}, \alpha_{\text{end}}]$ (e.g., red segment in Figure 2). We denote the extraction of a micro-experience from a prior as $\psi_{\mathcal{D}} = \xi'_{\mathcal{D}}(\alpha_{\text{ini}}, \alpha_{\text{end}})$, and say that such segment has a phase span $|\psi_{\mathcal{D}_\alpha}| \in (0, 1]$.

Extracted micro-experiences are exploited to create task-relevant motions. Formally, let $\nu : \psi_{\mathcal{D}} \rightarrow \psi \in \mathbb{R}^{(n+1) \times (n+1)}$ be a function that morphs a sequence of states onto another region of $\mathcal{S}$. We formulate the support of this operation to be that of an affine transformation of the form $\psi = A\psi_{\mathcal{D}} + B$, where $\psi_{\mathcal{D}(n+1) \times k} = \langle \bar{\mathbf{q}}_{\mathcal{D}_{n \times k}}, \bar{\alpha}_{\mathcal{D}_{1 \times k}} \rangle$ is a prior micro-experience with k states, and $\psi_{(n+1) \times k} = \langle \bar{\mathbf{q}}_{n \times k}, \bar{\alpha}_{1 \times k} \rangle$ is the generated task-relevant segment. Specifically, we design A to be a shear transform for its shape-preserving properties, and B to be a translation of the segment into a region of interest. Formally, then, this affine transformation modulates $\psi_{\mathcal{D}}$ as:

$$\begin{bmatrix} \bar{\mathbf{q}} \\ \bar{\alpha} \end{bmatrix} = \begin{bmatrix} \mathbb{I}_{n \times n} & \boldsymbol{\lambda}_{n \times 1} \\ \mathbb{0}_{1 \times n} & |\psi_{\mathcal{D}_\alpha}| \end{bmatrix} \begin{bmatrix} \bar{\mathbf{q}}_{\mathcal{D}} \\ \boldsymbol{\rho} \end{bmatrix} + \begin{bmatrix} \mathbf{b}_{n \times 1} & \dots & \mathbf{b}_{n \times 1} \\ \alpha_{\text{ini}} & \dots & \alpha_{\text{ini}} \end{bmatrix}_{(n+1) \times k}, \quad (2)$$

where $\boldsymbol{\lambda}_{n \times 1}$ is the shearing coefficient, $\mathbf{b}_{n \times 1}$ is a shifting vector, and $\boldsymbol{\rho} = [0, ..., 1]_{1 \times k}$ is a local reparametrisation of $\bar{\alpha}_{\mathcal{D}}$. Note that the phase of the generated segment ψ remains $\bar{\alpha} = \bar{\alpha}_{\mathcal{D}}$. Informally, Equation 2 translates and smoothly
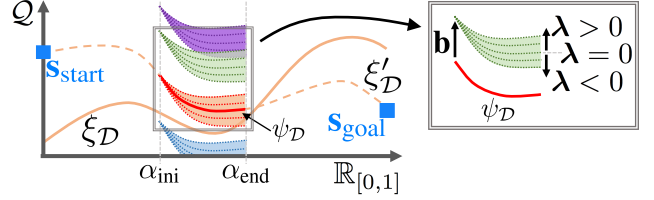
Fig. 2: Illustrative example of Equation 2: generation of resembling motions (dotted lines) by morphing the micro-experience $\psi_{\mathcal{D}}$ with semi-random $\mathbf{b}$ (shift) and $\boldsymbol{\lambda}$ (shear) pairs.

deforms the micro-experience by adding up the increasing amount of noise $\boldsymbol{\lambda}\boldsymbol{\rho} + \mathbf{b}$, such that $\psi(\alpha_{\text{ini}}) - \psi_{\mathcal{D}}(\alpha_{\text{ini}}) = \mathbf{b}$ and $\psi(\alpha_{\text{end}}) - \psi_{\mathcal{D}}(\alpha_{\text{end}}) = \boldsymbol{\lambda} + \mathbf{b}$. Therefore, specifying $\mathbf{b}$ and $\boldsymbol{\lambda}$ enables the generation of new micro-experiences and their mapping onto any region of interest in $\mathcal{S}$. Figure 2 exemplifies the affine morphing in Equation 2 with different parameters. We detail the implementation of Equation 2 in Algorithm 1.

Algorithm 1: MORPH_SEGMENT($\psi_{\mathcal{D}}$, $\boldsymbol{\lambda}$, $\mathbf{b}$)

Input:
  $\psi_{\mathcal{D}}$: micro-experience of phase-span $|\psi_{\mathcal{D}_\alpha}|$ from α_{ini}
  $\boldsymbol{\lambda}$: shearing coefficient
  $\mathbf{b}$: shifting vector
Output:
  ψ: morphed motion

1 **for** $\rho \leftarrow 0$ **to** 1 **do**   // Equation 2
2     $\alpha = \rho|\psi_{\mathcal{D}_\alpha}| + \alpha_{\text{ini}}$
3     $\psi(\alpha) \leftarrow \psi_{\mathcal{D}}(\alpha) + \rho\boldsymbol{\lambda} + \mathbf{b}$
4 **return** ψ

We exploit the ability to morph parts of the mapped prior path experience $\xi'_{\mathcal{D}}$ to infer motions that are suitable to either *connect* two particular states $\mathbf{s}_{\text{init}}$ and $\mathbf{s}_{\text{targ}}$, or *explore* the best way to continue the task from a given state $\mathbf{s}_{\text{init}}$. These two processes, and their interaction with Algorithm 1, are detailed in Algorithm 2 and illustrated in Figure 3.

Connect (line 3 to 6): given two task-related configuration-phase states $\mathbf{s}_{\text{init}} = \langle \mathbf{q}_{\text{init}}, \alpha_{\text{init}} \rangle$ and $\mathbf{s}_{\text{targ}} = \langle \mathbf{q}_{\text{targ}}, \alpha_{\text{targ}} \rangle$, we hypothesise that a suitable connection may result from mapping the micro-experience $\psi_{\mathcal{D}} : \alpha \in [\alpha_{\text{init}}, \alpha_{\text{targ}}]$ between $\mathbf{s}_{\text{init}}$ and $\mathbf{s}_{\text{targ}}$. Thus, after extracting the relevant micro-experience from $\xi'_{\mathcal{D}}$ (line 4), the parameters $\mathbf{b}$ and $\boldsymbol{\lambda}$ of the mapping in Equation 2 are calculated such that the resulting micro-experience ψ satisfies $\psi(\alpha_{\text{init}}) = \mathbf{q}_{\text{init}}$ and $\psi(\alpha_{\text{targ}}) = \mathbf{q}_{\text{targ}}$ (line 5 and 6).

Explore (line 8 to 11): given one task-related configuration-phase state $\mathbf{s}_{\text{init}} = \langle \mathbf{q}_{\text{init}}, \alpha_{\text{init}} \rangle$, we hypothesise that a suitable continuation of the task is to apply a micro-experience similar to that $\psi_{\mathcal{D}} \in \xi'_{\mathcal{D}}$ starting at α_{init}. For that, we first determine which span of $\xi'_{\mathcal{D}}$ to exploit by defining α_{targ} (line 8). An appropriate α_{targ} depends on the direction in which $\xi'_{\mathcal{D}}$ is being exploited; we call it `forward` when exploiting the prior from $\xi'_{\mathcal{D}}(0)$ to $\xi'_{\mathcal{D}}(1)$, and `backward` otherwise. Correspondingly,

(a) ERT: first iteration example   (b) ERT: *explore* example   (c) ERT: *connect* example

Fig. 3: Experience-driven random trees iteratively build a tree (green) of micro-experiences. At each iteration, an existing node in the tree is randomly selected to either *explore* the most suitable continuation of the task (e.g., snapshots in (a) and (b)), or *connect* to another known state (e.g., the goal state as in (c) (ERT), or a state in the other tree (ERTConnect)). In both cases, relevant motions (dotted red) are generated by morphing micro-experiences (red) of the prior path experience $\xi'_{\mathcal{D}}$ (see Figure 2).

SAMPLE_SEGMENT_END(α_{init}) defines α_{targ} as:

$$\alpha_{\text{targ}} = \begin{cases} \min(\alpha_{\text{init}} + \mathbb{U}(\omega_{\min}, \omega_{\max}), 1), & \text{if } \texttt{forward} \\ \max(0, \alpha_{\text{init}} - \mathbb{U}(\omega_{\min}, \omega_{\max})), & \text{if } \texttt{backward} \end{cases} \quad (3)$$

where $\mathbb{U}(\omega_{\min}, \omega_{\max})$ draws a sample from a uniform distribution to determine the phase span of the extracted segment. The bounds $\omega_{\min}$ and $\omega_{\max}$ are discussed in Section IV-B. Next, the corresponding segment $\psi_{\mathcal{D}} : \alpha \in [\alpha_{\text{init}}, \alpha_{\text{targ}}]$ is extracted from the mapped prior path experience $\xi'_{\mathcal{D}}$ (line 9), and $\mathbf{b}$ is computed for the resulting micro-experience ψ to start at $\mathbf{s}_{\text{init}}$, i.e., to satisfy $\psi(\alpha_{\text{init}}) = \mathbf{q}_{\text{init}}$ (line 10). Finally, to generate a task-relevant motion from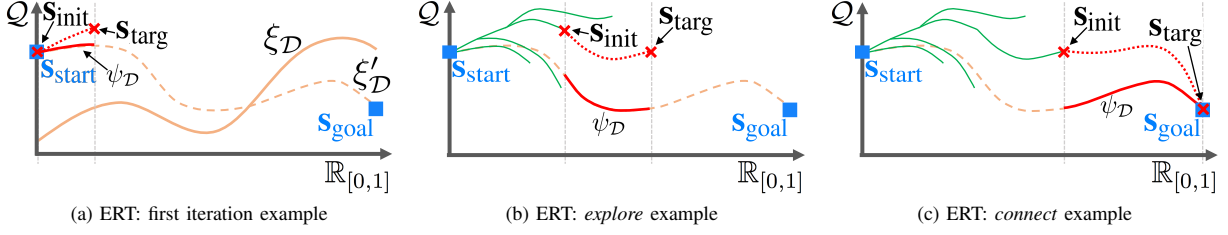 $\mathbf{q}_{\text{init}}$, the shearing coefficient $\boldsymbol{\lambda}$ is sampled randomly within some bounds to morph the micro-experience $\psi_{\mathcal{D}}$ into a similar motion (line 11). In particular, $\boldsymbol{\lambda}$ is drawn from a uniform distribution $\mathbb{U}(-\boldsymbol{\epsilon}|\psi_{\mathcal{D}\alpha}|, \boldsymbol{\epsilon}|\psi_{\mathcal{D}\alpha}|)$ such that, at each iteration, the maximum allowed deformation is proportional to the segment's phase span. This implies that the accumulated deformation along any possible path ξ found

by our planners does not exceed, with respect to $\xi'_{\mathcal{D}}$, the user-defined malleability bound ϵ (see discussion in Section IV-B).

Overall, the method GENERATE_SEGMENT($\cdot$) enables the presented experience-guided random tree planners to leverage a single path experience at different levels of granularity, and map task-relevant segments onto any region of interest in the configuration-phase space. In that way, our planner aims at composing a valid path from a suitable sequence of morphed micro-experiences. The remaining of this section discusses the usage of such routine in our uni-directional (ERT) and a bi-directional (ERTConnect) tree sampling-based techniques.

B. Uni-directional Experience-driven Random Trees (ERT)

Algorithm 3 provides the pseudo-code of the uni-directional version of our planner. The algorithm seeks finding a continu-

---

Algorithm 2: GENERATE_SEGMENT($\mathbf{s}_{\text{init}}$, $\mathbf{s}_{\text{targ}}$, $\xi'_{\mathcal{D}}$)

Input:
  $\mathbf{s}_{\text{init}}$: required segment configuration-phase start
  $\mathbf{s}_{\text{targ}}$: required (if any) segment configuration-phase end
  $\xi'_{\mathcal{D}}$: prior experience
Output:
  ψ: generated segment
  $\mathbf{s}_{\text{end}}$: end configuration-phase of the segment ψ

1 $\langle \mathbf{q}_{\text{init}}, \alpha_{\text{init}} \rangle = \mathbf{s}_{\text{init}}$
2 **if not** $s_{targ} = \varnothing$ **then**   // connect
3     $\langle \mathbf{q}_{\text{targ}}, \alpha_{\text{targ}} \rangle = \mathbf{s}_{\text{targ}}$
4     $\psi_{\mathcal{D}} \leftarrow \xi'_{\mathcal{D}}(\alpha_{\text{init}}, \alpha_{\text{targ}})$
5     $\mathbf{b} \leftarrow \mathbf{q}_{\text{init}} - \psi_{\mathcal{D}}(\alpha_{\text{init}})$
6     $\boldsymbol{\lambda} \leftarrow \mathbf{q}_{\text{targ}} - (\psi_{\mathcal{D}}(\alpha_{\text{targ}}) + \mathbf{b})$
7 **else**   // explore
8     $\alpha_{\text{targ}} \leftarrow$ SAMPLE_SEGMENT_END(α_{init})
9     $\psi_{\mathcal{D}} \leftarrow \xi'_{\mathcal{D}}(\alpha_{\text{init}}, \alpha_{\text{targ}})$
10    $\mathbf{b} \leftarrow \mathbf{q}_{\text{init}} - \psi_{\mathcal{D}}(\alpha_{\text{init}})$
11    $\boldsymbol{\lambda} \leftarrow \mathbb{U}(-\boldsymbol{\epsilon}|\psi_{\mathcal{D}\alpha}|, \boldsymbol{\epsilon}|\psi_{\mathcal{D}\alpha}|)$
12 $\psi \leftarrow$ MORPH_SEGMENT($\psi_{\mathcal{D}}$, $\boldsymbol{\lambda}$, $\mathbf{b}$)
13 $\mathbf{s}_{\text{end}} \leftarrow \langle \psi(\alpha_{\text{targ}}), \alpha_{\text{targ}} \rangle$
14 **return** $\langle \psi, \mathbf{s}_{\text{end}} \rangle$

Algorithm 3: ERT($\mathbf{s}_{\text{start}}$, $\mathbf{s}_{\text{goal}}$, $\xi_{\mathcal{D}}$)

Input:
  $\mathbf{s}_{\text{start}}$ and $\mathbf{s}_{\text{goal}}$: start and goal configuration-phase
  $\xi_{\mathcal{D}}$: prior experience
Output:
  ξ: collision-free path

  /* map $\xi_{\mathcal{D}}$ onto current problem   */
1 $\langle \xi'_{\mathcal{D}}, \varnothing \rangle \leftarrow$ GENERATE_SEGMENT($\mathbf{s}_{\text{start}}$, $\mathbf{s}_{\text{goal}}$, $\xi_{\mathcal{D}}$)
2 **if** IS_VALID($\xi'_{\mathcal{D}}$) **then**
3   **return** $\xi'_{\mathcal{D}}$

  /* sampling-based $\xi'_{\mathcal{D}}$ exploitation   */
4 $\mathcal{T}$.init($\mathbf{s}_{\text{start}}$)
5 **while not** STOPPING_CONDITION() **do**
6   /* node selection   */
      $\mathbf{s}_{\text{init}} \leftarrow \mathcal{T}$.select_node()
7   /* micro-experience generation   */
      $\mathbf{s}_{\text{targ}} \leftarrow \varnothing$
8   **if** ATTEMPT_GOAL() = *True* **then**
9        $\mathbf{s}_{\text{targ}} \leftarrow \mathbf{s}_{\text{goal}}$
10    $\langle \psi, \mathbf{s}_{\text{targ}} \rangle \leftarrow$ GENERATE_SEGMENT($\mathbf{s}_{\text{init}}$, $\mathbf{s}_{\text{targ}}$, $\xi'_{\mathcal{D}}$)
11   /* tree extension   */
      if EXTEND($\mathcal{T}$, ψ, $\mathbf{s}_{\text{init}}$, $\mathbf{s}_{\text{targ}}$) $\neq$ *Failed* **then**
12   **if** GOAL_REACHED($\mathbf{s}_{\text{targ}}$) **then**
13   **return** PATH($\mathcal{T}$)

Algorithm 4: EXTEND($\mathcal{T}$, ψ, $\mathbf{s}_\text{init}$, $\mathbf{s}_\text{targ}$)

Input:

   $\mathcal{T}$: tree of previously generated micro-experiences

   ψ: new generated micro-experience

   $\mathbf{s}_\text{init}$ and $\mathbf{s}_\text{targ}$: start and end configuration-phase of ψ

Output:

   outcome of the tree extension attempt

1 **if** IS_VALID(ψ) **then**

2     $\mathcal{T}$.add_vertex($\mathbf{s}_\text{targ}$)

3     $\mathcal{T}$.add_edge(ψ, $\mathbf{s}_\text{init}$, $\mathbf{s}_\text{targ}$)

4   **return** *Advanced*

5 **return** *Failed*

---

ous path from a start $\mathbf{s}_\text{start}$ to a goal $\mathbf{s}_\text{goal}$ configuration, given a related path experience $\xi_\mathcal{D}$. The planner firstly maps the entire prior experience $\nu : \xi_\mathcal{D} \rightarrow \xi'_\mathcal{D}$ onto the current planning problem (line 1); note that the output of GENERATE_SEGMENT($\cdot$) (Algorithm 2) is a segment ψ that spans from $\alpha_\text{ini} = 0$ to $\alpha_\text{end} = 1$, thus we rename it $\xi'_\mathcal{D}$. If $\xi'_\mathcal{D}$ is not valid (line 2), the planner proceeds to exploit $\xi'_\mathcal{D}$ to generate task-relevant micro-experiences. The planner follows a three-step procedure (node selection, segment sampling, and tree extension) until the stopping condition is met (line 5). A node $\mathbf{s}_\text{init}$ is selected from the tree $\mathcal{T}$ with probability $P(node) = \frac{1}{w(node)+1}$ (line 6), where $w(\cdot)$ is a weighting function that penalises the selection of a node according to the number of times that it has already been selected. This weighted sampling strategy seeks a uniform selection of all nodes over time, thus promoting a first depth exploration of the task phase α. From the selected node $\mathbf{s}_\text{init}$, the tree is expanded using segments that resemble those in the prior experience $\xi'_\mathcal{D}$. With probability p, the expansion of the tree attempts to *connect* $\mathbf{s}_\text{init}$ with $\mathbf{s}_\text{goal}$, whereas with probability $(1 - p)$ an *explore* expansion is done towards a semi-random configuration $\mathbf{s}_\text{targ}$ (line 7 to 10). Algorithm 2, previously explained in Section III-A, details the extraction of suitable segments under these two different cases. The extracted segment is used to attempt expanding the tree (line 11) following Algorithm 4. If the segment is valid, it is integrated into the tree. Note that, as discussed in Section III-A, the appended segment is a motion whose shape resembles that of the related micro-experiences in the prior experience, not a straight line. Finally, if the incorporated (valid) segment reaches the goal, the path is returned (line 12 and 13).

C. Bi-directional ERT (ERTConnect)

The principles of leveraging from a prior experience by generation of task-relevant micro-experiences can also be employed in a bi-directional fashion. The proposed bi-directional planning scheme resembles, in spirit, that of the RRTConnect [19], i.e., to simultaneously grow two trees, one from the start configuration and the other from the goal configuration, aiming to find a solution by connecting both trees. ERTConnect, however, includes the peculiarities of our experience-based planning approach. As shown in Algorithm 5, the planner firstly maps the prior path experience $\nu : \xi_\mathcal{D} \rightarrow \xi'_\mathcal{D}$ onto the current planning problem (line 1). If $\xi'_\mathcal{D}$ is not valid

---

Algorithm 5: ERTConnect($\mathbf{s}_\text{start}$, $\mathbf{s}_\text{goal}$, $\xi_\mathcal{D}$)

Input:

   $\mathbf{s}_\text{start}$ and $\mathbf{s}_\text{goal}$: start and goal configuration-phase

   $\xi_\mathcal{D}$: prior experience

Output:

   ξ: collision-free path

   `/* map `$\xi_\mathcal{D}$` onto current problem     */`

1 $\langle\xi'_\mathcal{D}, \varnothing\rangle \leftarrow$ GENERATE_SEGMENT($\mathbf{s}_\text{start}$, $\mathbf{s}_\text{goal}$, $\xi_\mathcal{D}$)

2 **if** IS_VALID($\xi'_\mathcal{D}$) **then**

3   **return** $\xi'_\mathcal{D}$

   `/* sampling-based `$\xi'_\mathcal{D}$` exploitation   */`

4 $\mathcal{T}_a$.init($\mathbf{s}_\text{start}$)

5 $\mathcal{T}_b$.init($\mathbf{s}_\text{goal}$)

6 **while not** STOPPING_CONDITION() **do**

     `/* node selection                 */`

7     $\mathbf{s}_\text{init} \leftarrow \mathcal{T}_a$.select_node()

     `/* micro-experience generation    */`

8     $\langle\psi, \mathbf{s}_\text{targ}\rangle \leftarrow$ GENERATE_SEGMENT($\mathbf{s}_\text{init}$, $\varnothing$, $\xi'_\mathcal{D}$)

     `/* tree extension                 */`

9   **if** EXTEND($\mathcal{T}_a$, ψ, $\mathbf{s}_\text{init}$, $\mathbf{s}_\text{targ}$) $\neq$ *Failed* **then**

10   **if** OTHER_EXTREME_REACHED($\mathbf{s}_\text{targ}$) **then**

11   **return** PATH($\mathcal{T}_a$)

       `/* micro-experience generation    */`

12       $\mathbf{s}_\text{near} \leftarrow \mathcal{T}_b$.nearest_neighbour($\mathbf{s}_\text{targ}$)

13       $\langle\psi, \mathbf{s}_\text{targ}\rangle \leftarrow$ GENERATE_SEGMENT($\mathbf{s}_\text{near}$, $\mathbf{s}_\text{targ}$, $\xi'_\mathcal{D}$)

       `/* tree connection                */`

14   **if** EXTEND($\mathcal{T}_b$, ψ, $\mathbf{s}_\text{near}$, $\mathbf{s}_\text{targ}$) $\neq$ *Failed* **then**

15   **return** PATH($\mathcal{T}_a$, $\mathcal{T}_b$)

16   SWAP($\mathcal{T}_a$, $\mathcal{T}_b$)

(line 2), the planner proceeds to exploit $\xi'_\mathcal{D}$ to compute a solution. The planner simultaneously grows two trees, one rooted at $\mathbf{s}_\text{start}$ and the other at $\mathbf{s}_\text{goal}$ (line 4 and 5). At each iteration, until the stopping criterion is met (line 6), a node of the active tree $\mathcal{T}_a$ is selected via weighted selection (line 7) to *explore* the space via a task-relevant micro-experience (line 8). If the active tree is extended successfully with the generated segment (line 9), we first check whether the end of the motion has reached the other extreme (line 10). This implies that a path has been found before the trees connected, either by $\mathcal{T}_a$ reaching the root of $\mathcal{T}_b$ or the other way around, in which case the path is returned as a solution (line 11). Otherwise, the node of $\mathcal{T}_b$ nearest to $\mathbf{s}_\text{targ}$ is selected (line 12) to attempt to *connect* both trees with a task-relevant segment (line 13). For such nearest neighbour query, we consider the Euclidean distance between the configuration components (without the phase). If the extension is successful, the corresponding path is returned (line 14 and 15).

IV. USING ERT AND ERTCONNECT

In this section we discuss some details on using the proposed experience-driven random trees planners.

A. Selecting a Prior from a Library of Experiences

A robot might have at its disposal a library $\mathcal{A}$ of task-relevant path experiences. As our planners exploit a unique prior path experience $\xi_{\mathcal{D}}$ to solve other instances of the same task, our current selection criteria $\xi_{\mathcal{D}} \in \mathcal{A}$ is to pick the prior that resembles the current planning query the most. Intuitively, if a solution to the current planning problem lies in the neighbourhood of a prior experience, the invariant robotic constraints encoded in the experience itself, such as self-collisions and joint limits, are more likely to prevail and thus, ease the planner's computations. Inspired by the experience selection in [11], we estimate such resemblance by ranking the experiences for their similarity to the start and goal of the planning query. This is, the prior experience $\xi_{\mathcal{D}}$ selected to feed the proposed experience-based planner is such that:

$$\xi_{\mathcal{D}} = \arg\min_{\xi_{\mathcal{D}i} \in \mathcal{A}} \mathrm{dist}(\xi_{\mathcal{D}i}(0), \mathbf{q}_{\mathrm{start}}) + \mathrm{dist}(\xi_{\mathcal{D}i}(1), \mathbf{q}_{\mathrm{goal}}), \quad (4)$$

where $\mathbf{q}_{\mathrm{start}}$ and $\mathbf{q}_{\mathrm{goal}}$ are the start and goal configurations of the current planning problem, and $\mathrm{dist}(\cdot)$ is a function that estimates the Euclidean distance between configurations in $\mathcal{Q}$.

We verified the approach to select a unique prior from a library of experiences described in Equation 4 experimentally; despite it led to good results, the topic merits further attention.

B. Planner's Parameterisation

Next, we review the parameters of the presented planners:

- p - probability of attempting to *connect* the tree to the goal (only in ERT). This parameter should be set small to allow the planner *explore* the space. Default: $p = 0.05$.
- $\omega_{\min}$ and $\omega_{\max}$ - lower and upper phase span bounds of the extracted segments. Indirectly, these parameters delimit the length of the motions added in the tree to *explore* the space. Default: $\omega_{\min} = 0.05$ and $\omega_{\max} = 0.1$.
- ϵ - malleability bound to delimit the amount of morphing applied to the micro-experiences. Intuitively, this parameter defines a volume (tube) around $\xi'_{\mathcal{D}}$ where the planner can *explore* for a solution. Default: $\epsilon = \mathbf{5}_{1 \times n}$ (large enough to cover the entire robot's kinematic range).

Our planners' default parameters are non-optimised for a particular planning problem, but left generic to succeed in many scenarios provided a relevant path experience. The planners' behaviour can be adjusted by tuning, namely, $\omega_{\min}$, $\omega_{\max}$ and ϵ. As our planner discards motions that are not entirely valid, large phase spans might endanger the ability to build a tree. However, in scenarios with few obstacles, $\omega_{\min}$ and $\omega_{\max}$ can be set large to speed up planning computations. Also, lowering ϵ can speed up computations, as the tree growth would be more guided around the mapped experience $\xi'_{\mathcal{D}}$. The lower ϵ, the more dependant the planner is on the suitability of the provided experience as the probabilistic completeness is compromised. Knowing the level of dissimilarity between the experience and the current problem might aid in tuning ϵ to trade growth guidance and space exploration.

V. EXPERIMENTAL EVALUATION

The proposed experience-guided random trees have been implemented in the Open Motion Planning Library

(OMPL) [20] and evaluated on the Fetch robot [21] in a shelf-stocking task. Fetch is a humanoid robot with a 7-DoF arm attached on a sliding torso, thus requiring to plan in an 8-DoF configuration space. Our experiments are designed to measure the generalisation capabilities of our planner in scenarios that involve different levels of dissimilarity between prior experiences and task instances (see Section V-A). The considered task instances include synthetic and real-world scenarios (see Section V-B).

A. Experimental Setup

Our experimental setup considers a varied instance set of the challenging problem of reaching a target object in a shelving unit, specifically in a synthetic 4-tier (see Figure 4) and a narrower real 5-tier shelving unit (see Figure 5). Task instances in these scenarios not only present variability on the location of the shelving unit ($\pm 90°$ around the robot) and the robot's initial position (± 10cm), but also on the location of the target object and the obstacles within the shelving unit.

To further evaluate the generalisation capabilities of the proposed experience-based planner, we introduce some additional variability across the experimental setup. This is, we compute with RRTConnect a total of 100 experiences from different task instances; them all at the synthetic shelving unit, with target objects in the middle shelf and no obstacles. These scenarios are discarded for the rest of the evaluation. Then, these experiences are used to evaluate the planner in four scenario sets that involve increasing dissimilarity levels between experiences and planning queries:

- *Set 1*: 200 instances with target objects in the middle shelf without obstacles (synthetic). Note that these instances resemble those used to compute experiences.
- *Set 2*: 200 instances with target objects in the middle shelf with the presence of obstacles (synthetic).
- *Set 3*: 200 instances with target objects in three different shelves with the presence of obstacles (synthetic).
- *Set 4*: 120 instances with target objects and obstacles in the middle shelf (real-world).

The four sets of task instances are used to benchmark our bi-directional ERTConnect planner against RRTConnect [19] and the most representative experience-based planners that employ motions as prior information of the task, i.e., Thunder [14] and Lightning [11]. Note that these two frameworks are double-threaded with a bi-directional 'retrieve and repair' (RR) and 'plan from scratch' (PFS) module. Similarly, for a fair comparison, we embed our ERTConnect in a double-threaded framework which runs RRTConnect in parallel to PFS. The reported results indicate the contribution of the RR (plain bar) and PFS (stripped bar) modules in solving the planning queries separately, and the required planning time jointly (plain bar).

In this work's context, where we consider novel tasks instances in varied scenarios, optimising each planner's parameters across queries is not possible. Optimal parametrisation requires extensive testing in each scenario set, and thus knowing the scenarios in advance, among other planning aspects. Therefore, all planners are used in their default OMPL settings, and ours is set to the non-optimised default parameters
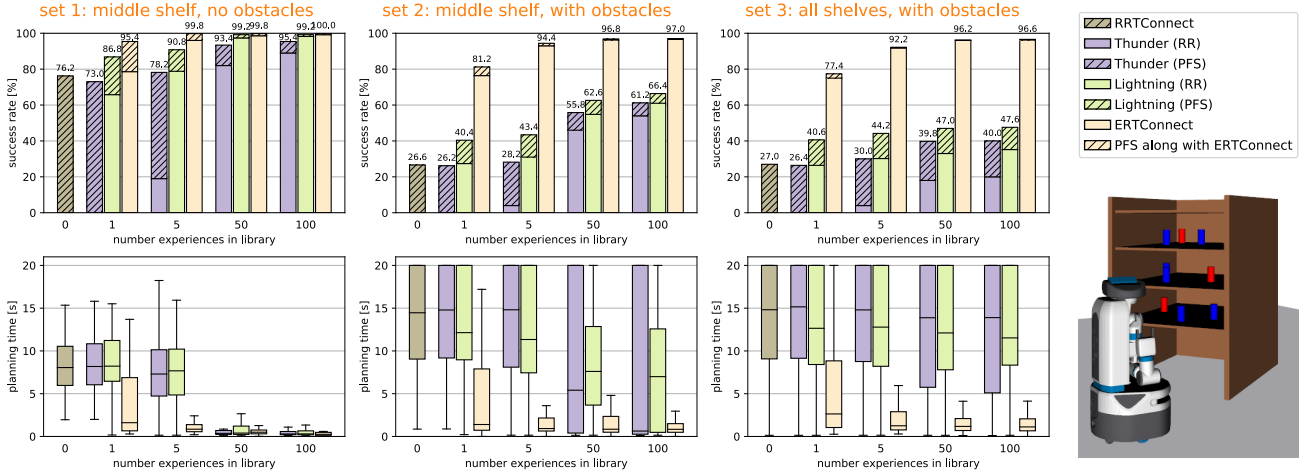
Fig. 4: Success rate and solving time results for the benchmark on synthetic scenarios, where the Fetch Robot needs to reach a target object in the shelving unit subject to multiple variations of the task instances. From left to middle-right column, case studies from less to more experience-instance dissimilarity: *Set 1*, *Set 2* and *Set 3*. The picture on the right depicts a particular instance of *Set 3* which, differently from the considered prior experiences, involves target objects (red cylinders) located at any of the three shelves, a different relative location of the shelving unit, as well as obstacles (blue cylinders).

specified in Section IV-B. The benchmark is run on an Intel i7 Linux machine with 4 3.6GHz cores and 16GB of RAM. The performance of the three experience-based planners in each instance set is assessed under libraries with $\{1, 5, 50, 100\}$ prior experiences. The experiences provided to each planner are the same. Each query is repeated 50 times with a planning timeout of 20 seconds. All in all, the conducted benchmark involves a total of 468,000 planning queries.

B. Results on Synthetic and Real-world Scenarios

The results of the benchmark on *Set 1*, *Set 2* and *Set 3* are summarised in Figure 4, whilst those in the real-world *Set 4* are depicted in Figure 5. As it can be observed, provided a high number of experiences that are close to the current planning problem (i.e., *Set 1* with the library of 100 experiences), all planners achieve a high success rate with solving time of the order of milliseconds. This behaviour is expected as, given the experience-query similarity and the library size, it is likely that there exists a prior experience that nearly resembles the current query, thus involving minimum repairing.

As the dissimilarity between experiences and queries increases, experience-based planners need to generalise the prior information more broadly to succeed. Intuitively, the need of generalisation arises when a reduced number of demonstrations in the library needs to cover varied task instances (x-axis within each experimental set), or when the current planning requirements differ significantly from the set of available experiences (variability across experimental sets). The outcome of our benchmark points out that the performance of Thunder's RR module drops abruptly by either dissimilarity factor, whereas the Lighting's RR is not as affected by the lack of experiences as it is when dealing with significantly different task instances. The poor generalisation of these frameworks across instances is due to the rigid usage of prior experiences. Our approach, instead, by leveraging experiences

in a malleable way, achieves a success rate and solving time that significantly improves that of Lightning and Thunder.

The importance of generalising prior information is particularly noticeable in the real-world task instances in *Set 4*, where the queries differ from the experiences not only on the location of the shelving unit and the target object, but also on the narrower geometry of the whole shelving unit, the height of the shelf where the target object is located at, and the presence of obstacles. Under these challenging task variations and when accounting with only one demonstration, our approach outperforms by a factor of approximately 3.7
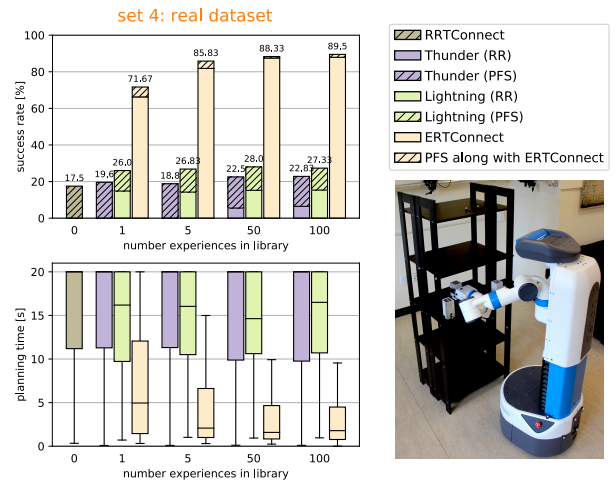

Fig. 5: Success rate and solving time results for the benchmark on real-world scenarios (*Set 4*), where the Fetch Robot needs to reach a target object by generalising prior experiences to a narrower shelving unit geometry, to different locations of the shelving unit, robot's initial position and target object, as well as to the presence of obstacles.

and 37.1 times the RR module of the Lightning and Thunder frameworks, respectively. Similarly, those frameworks are respectively outperformed by our approach by a factor of 4.7 and 8.9 times when considering 100 experiences. Notably, while these planners time out most of the trials, our method required only a quarter (with one demonstration) and less than an eight (with multiple demonstrations) of the planning time budget to find a solution.

The ability to generalise prior information not only limits the level of experience-query dissimilarity that a planner can cope with, but also the number of experiences that are required to achieve high performance. As an example, providing 50 experiences to Thunder's RR, 5 to Lightning's RR and 1 to our ERTConnect leads to approximately the same success rate of 80% in *Set 1*. Achieving such a performance in the other experimental sets with Lightning and Thunder is not possible even with a library of 100 experiences, whereas our approach surpasses such performance when selecting a unique experience from a library of only 5 experiences. This implies that our ERTConnect planner, by generalising prior experiences more efficiently, significantly outperforms current experienced-based planners using libraries of motions in the literature even when provided with notably fewer experiences.

VI. DISCUSSION

In this manuscript, we have presented two new experience-based planners: the uni-directional experience-driven random tress (ERT) and the bi-directional ERT (ERTConenct). These two methods are tree sampling-based planners that iteratively exploit a single prior path experience to ease the capture of connectivity of the space. At each iteration, a segment of the prior is extracted and semi-randomly morphed to generate a task-relevant motion. The obtained motions are sequentially concatenated to compose a task-relevant tree, such that a trace along the edges constitutes a solution to a given task-related planning problem. Thorough experimentation against current experienced-based planners using libraries of motions in the literature [11], [14] demonstrates our planner's significant superior performance in a wide range of task instances.

We have shown that, similarly to related work [11], [14], our planner can be used in parallel with a planning from scratch strategy to guarantee probabilistic completeness. Therefore, when multi-threading is an option, a planning from scratch thread should be considered, as well as multiple instantiations of our planner with a set of varied experiences that maximises space coverage. In the future, we plan to explore the convenience of different transformation supports to infer relevant micro-experiences subject to intrinsic task and robot constraints; for instance, early tests demonstrate our planners' suitability to leverage experiences in $SO(3)$ using quaternions. Another promising line for future work is the extension of our planner to leverage multiple experiences simultaneously, such that the local exploration is conducted with the most suitable segment in the library. Likewise, such a strategy would potentially allow the planner adapting to changes in the planning context, e.g., dynamic obstacles and moving goal configurations, as well as solving novel tasks by combining experiences of multiple different tasks.

REFERENCES

[1] F. Meier, D. Kappler, and S. Schaal, "Online learning of a memory for learning rates," in *2018 IEEE International Conference on Robotics and Automation*, pp. 2425–2432, IEEE, 2018.

[2] È. Pairet, P. Ardón, M. Mistry, and Y. Petillot, "Learning generalizable coupling terms for obstacle avoidance via low-dimensional geometric descriptors," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3979–3986, 2019.

[3] S. Stark, J. Peters, and E. Rueckert, "Experience reuse with probabilistic movement primitives," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1210–1217, IEEE, 2019.

[4] È. Pairet, P. Ardón, M. Mistry, and Y. Petillot, "Learning and composing primitive skills for dual-arm manipulation," in *Annual Conference Towards Autonomous Robotic Systems*, pp. 65–77, Springer, 2019.

[5] H. Ravichandar, A. S. Polydoros, S. Chernova, and A. Billard, "Recent advances in robot learning from demonstration," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 3, 2020.

[6] M. Zucker, J. Kuffner, and J. A. Bagnell, "Adaptive workspace biasing for sampling-based planners," in *2008 IEEE International Conference on Robotics and Automation*, pp. 3757–3762, IEEE, 2008.

[7] B. Ichter, J. Harrison, and M. Pavone, "Learning sampling distributions for robot motion planning," in *2018 IEEE International Conference on Robotics and Automation*, pp. 7087–7094, IEEE, 2018.

[8] P. Lehner and A. Albu-Schäffer, "The repetition roadmap for repetitive constrained motion planning," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3884–3891, 2018.

[9] C. Chamzas, A. Shrivastava, and L. E. Kavraki, "Using local experiences for global motion planning," in *2019 International Conference on Robotics and Automation*, pp. 8606–8612, IEEE, 2019.

[10] D. Molina, K. Kumar, and S. Srivastava, "Learn and link: learning critical regions for efficient planning," in *IEEE International Conference on Robotics and Automation*, 2020.

[11] D. Berenson, P. Abbeel, and K. Goldberg, "A robot path planning framework that learns from experience," in *2012 IEEE International Conference on Robotics and Automation*, pp. 3671–3678, IEEE, 2012.

[12] N. Jetchev and M. Toussaint, "Fast motion planning from experience: trajectory prediction for speeding up movement generation," *Autonomous Robots*, vol. 34, no. 1-2, pp. 111–127, 2013.

[13] M. Phillips, B. J. Cohen, S. Chitta, and M. Likhachev, "E-graphs: bootstrapping planning with experience graphs.," in *Robotics: Science and Systems*, 2012.

[14] D. Coleman, I. A. Şucan, M. Moll, K. Okada, and N. Correll, "Experience-based planning with sparse roadmap spanners," in *2015 IEEE International Conference on Robotics and Automation*, pp. 900–905, IEEE, 2015.

[15] Y. Wang, K. Harada, and W. Wan, "Motion planning through demonstration to deal with complex motions in assembly process," in *2019 IEEE-RAS 19th International Conference on Humanoid Robots*, pp. 544–550, IEEE, 2019.

[16] J. DelPreto, J. I. Lipton, L. Sanneman, A. J. Fay, C. Fourie, C. Choi, and D. Rus, "Helping robots learn: a human-robot master-apprentice model using demonstrations via virtual reality teleoperation," in *IEEE International Conference on Robotics and Automation*, 2020.

[17] P. Ardón, È. Pairet, Y. Petillot, R. P. Petrick, S. Ramamoorthy, and K. S. Lohan, "Self-assessment of grasp affordance transfer," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, 2020.

[18] D. Hsu, J.-C. Latombe, and R. Motwani, "Path planning in expansive configuration spaces," in *Proceedings of International Conference on Robotics and Automation*, vol. 3, pp. 2719–2726, IEEE, 1997.

[19] J. J. Kuffner and S. M. LaValle, "RRT-connect: an efficient approach to single-query path planning," in *2000 International Conference on Robotics and Automation*, vol. 2, pp. 995–1001, IEEE, 2000.

[20] I. A. Şucan, M. Moll, and L. E. Kavraki, "The Open Motion Planning Library," *IEEE Robotics & Automation Magazine*, vol. 19, pp. 72–82, December 2012.

[21] M. Wise, M. Ferguson, D. King, E. Diehr, and D. Dymesich, "Fetch and Freight: standard platforms for service robot applications," in *Workshop on autonomous mobile service robots*, 2016.

9.2   Supplementary Material

In this section, we provide some additional discussion on the results presented in the paper (see Section 9.2.1), as well as some details on employing experience-driven random trees in SO(3) (see Section 9.2.2) and optimising the usage of our planner in multiple threads (see Section 9.2.3).

9.2.1   Path Quality

Besides the advantages on computation time and success rate that our planner offers in contrast to traditional and other experience-based planning techniques, the presented experience-driven random trees ERT and ERTConnect planners also compute paths of better quality. Similar to the experiments in the paper, we benchmark our bi-directional ERTConnect planner against RRTConnect [91] and the most representative experience-based planners that employ motions as prior information of the task, i.e., Lightning [17] and Thunder [37]. We consider the solutions obtained in the experimental set 1 of the paper when using 100 experiences in the library. This analysis includes a total of 37,086 trajectories across the four planners. As shown in Figure 9.1, we analyse the average length and smoothness of the solutions computed by each planner. RRTConnect, Thunder and Lightning show a similar performance; this behaviour is expected as the underlying planning process of all these planners is the RRTConnect. In contrast, the solutions computed with our ERTConnect are significantly both shorter and smoother.
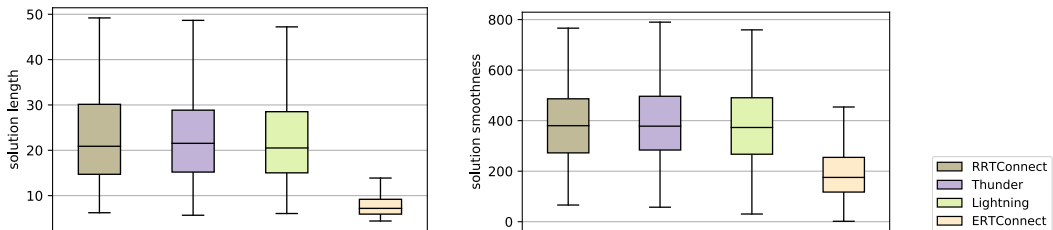
Figure 9.1: Comparison of the solutions computed by RRTConnect [91], Lightning [17] and Thunder [37] in contrast to our ERTConnect planner.

9.2.2   Micro-experience Malleability in SO(3)

We originally presented the experience-driven random trees ERT and ERTConnect planners for robotic manipulation planning. However, the concept of micro-experience (de)composability and malleability is not strictly restricted to $\mathbb{R}^n$ planning spaces. We detail our approach in SO(3) using quaternions as a generic representation of any rotation.

Let the orientation of a robot in a motion plan be defined in the configuration-phase space $\mathcal{S} = \mathrm{SO}(3) \times \mathbb{R}_{[0,1]}$ as $\mathbf{s} = \langle \mathbf{q}, \alpha \rangle$. Then, the affine transformation described in (2) and Algorithm 1 for a SO(3) space is:

$$
\begin{aligned}
\bar{\mathbf{q}}_{4 \times k} &= \bar{\mathbf{q}}_{\mathcal{D} 4 \times k} \circ [\mathbf{b}_{4 \times 1} \dots \mathbf{b}_{4 \times 1}]_{4 \times k} \circ f(\mathbf{q}_{\mathbb{1}}, \ \lambda, \ \boldsymbol{\rho})_{4 \times k}, \\
\bar{\alpha}_{1 \times k} &= \bar{\alpha}_{\mathcal{D}},
\end{aligned}
\tag{9.1}
$$

where $\mathbf{b}_{4 \times 1}$ is a quaternion to rotate the whole prior micro-experience $\psi_{\mathcal{D}}$, $\boldsymbol{\lambda}_{4 \times 1}$ is a quaternion to indicate the desired shearing, $\circ$ denotes the Hadamard (element-wise) quaternion product, and $f(\mathbf{q}_{\mathbb{1}}, \ \lambda, \ \boldsymbol{\rho})$ computes a quaternion trajectory by interpolating from the identity quaternion $\mathbf{q}_{\mathbb{1}} = [0, 0, 0, 1]^T$ to $\boldsymbol{\lambda}$ at each local phase $\boldsymbol{\rho} = [0, ..., 1]_{1 \times k}$. Analogously to (2), (9.1) deforms the micro-experience by incrementally applying a $\mathbf{b} \circ f(\mathbf{q}_{\mathbb{1}}, \ \lambda, \ \boldsymbol{\rho})$ rotation, such that the difference between $\psi(\alpha_{\mathrm{ini}})$ and $\psi_{\mathcal{D}}(\alpha_{\mathrm{ini}})$ is $\mathbf{b}$, and that between $\psi(\alpha_{\mathrm{end}})$ and $\psi_{\mathcal{D}}(\alpha_{\mathrm{end}})$ is $\mathbf{b} \circ f(\mathbf{q}_{\mathbb{1}}, \ \lambda, \ 1)$.

Then, generating new micro-experiences in Algorithm 2 consists in determining $\mathbf{b}$ and $\boldsymbol{\lambda}$ to map segments of the prior experience onto any region of interest in $\mathcal{S}$. For the *connect* routine, this is:

$$
\begin{aligned}
\mathbf{b} &= \mathbf{q}_{\mathrm{init}} \psi_{\mathcal{D}}(\alpha_{\mathrm{init}})^{\text{-}1}, \\
\boldsymbol{\lambda} &= \mathbf{q}_{\mathrm{targ}} (\psi_{\mathcal{D}}(\alpha_{\mathrm{targ}}) \mathbf{b})^{\text{-}1},
\end{aligned}
\tag{9.2}
$$

whereas for the *explore* routine, this is:

$$
\begin{aligned}
\mathbf{b} &= \mathbf{q}_{\mathrm{init}} \psi_{\mathcal{D}}(\alpha_{\mathrm{init}})^{\text{-}1}, \\
\boldsymbol{\lambda} &= \mathbb{U}(\boldsymbol{\epsilon} | \psi_{\mathcal{D}_{\alpha}} |),
\end{aligned}
\tag{9.3}
$$

where $\boldsymbol{\epsilon}$ is a user-defined malleability bound that, intuitively, determines the volume (neighbourhood) around the provided $\psi_{\mathcal{D}}$ that the planner can explore, and $\mathbb{U}(\boldsymbol{\epsilon} | \psi_{\mathcal{D}_{\alpha}} |)$ samples a unit quaternion uniformly within distance $\boldsymbol{\epsilon} | \psi_{\mathcal{D}_{\alpha}} |$ of the identity quaternion. The malleability upper bound $\boldsymbol{\epsilon}$ in (9.3) is scaled by the segment's phase span $| \psi_{\mathcal{D}_{\alpha}} |$ such that the accumulated deformation along the resulting path does not exceed $\boldsymbol{\epsilon}$.

Note that all operators employed in this section refer to quaternion operators. For the interpolation denoted as $f(\cdot)$ in (9.1), as it starts from the identity quaternion, we use a simplified version of the spherical linear interpolation (SLERP) [153]. For the uniform bounded sampling $\mathbb{U}(\cdot)$ in (9.3), we use a similar strategy as that used in the open motion planning library (OMPL) [161] to sample a quaternion within a certain distance of another.

9.2.3 Selecting Multiple Priors from a Library of Experiences

As discussed in the paper, when multi-threading is an option, not only a planning from scratch module should be considered to ensure completeness guarantees, but also multiple instantiations

of our planner with a set of varied instances. Ideally, this set of experiences should capture distinctive topological cases of the same task such that minimum repairing is required and, consequently, reduce the required planning time as well as increasing the success rate. Formally, let $g(\cdot)$ be a function that measures the performance of the overall multi-thread planning system. Then, given a library of prior path experiences $\mathcal{A} = \{\xi_{\mathcal{D}_1}, \xi_{\mathcal{D}_2}, ..., \xi_{\mathcal{D}_j}\}$ relevant to the current problem, we seek the subset $\mathcal{B}_k \subset \mathcal{A}$ that maximises the performance of the overall multi-thread planning system with k threads. We denote the optimisation objective as:

$$\underset{\mathcal{B}_k}{\arg\max}\, g(\mathcal{B}_k) \quad s.t.\ |\mathcal{B}_k| = k. \tag{9.4}$$

As explicitly computing (9.4) for novel planning problems would prohibitively require solving the problems in advance, we instead attempt to approximate $g(\cdot)$ with metrics. In particular, we analyse four strategies that aim at retrieving the optimal $\mathcal{B}_k$ by instantiating k threads with:

- **M1**: the best experience as ranked by the selection criteria (4) in the paper.

- **M2**: the first k best experiences as ranked by the selection criteria (4) in the paper.

- **M3**: the best k experiences as seen across prior motion plans.

- **M4**: the best experience as ranked by the selection criteria (4) in the paper and its $k-1$ best partnering experiences as seen across prior motion plans.

The methods **M3** and **M4** guide the selection of $\mathcal{B}_k$ with the performance of the prior experiences in the prior motion plans. This approach builds on the hypothesis that the set of task instances E used to retrieve the path experiences in $\mathcal{A}$ are representative to future instantiations. Therefore, as part of the offline process of building the library of experiences $\mathcal{A}$, we include, for each experience, a reference to the $k-1$ partnering experiences that maximise $g(\cdot)$. For the particular case of $g(\cdot)$ measuring success rate, (9.4) is computed as:

$$\underset{\mathcal{B}_k}{\arg\max}\, \frac{1}{|E|} \sum_{e \in E} \mathrm{P}(\mathrm{mt\_planner}(\mathcal{B}_k, e) == \mathrm{success}) \tag{9.5}$$

$$= \underset{\mathcal{B}_k}{\arg\max}\, \frac{1}{|E|} \sum_{e \in E} \mathrm{P}\left(\left(\bigcup_{\xi_{\mathcal{D}_i} \in \mathcal{B}_k} \mathrm{planner}(\xi_{\mathcal{D}_i}, e)\right) == \mathrm{success}\right) \tag{9.6}$$

$$= \underset{\mathcal{B}_k}{\arg\max}\, \frac{1}{|E|} \sum_{e \in E} \left(1 - \prod_{\xi_{\mathcal{D}_i} \in \mathcal{B}_k} \mathrm{P}(\mathrm{planner}(\xi_{\mathcal{D}_i}, e) == \mathrm{failure})\right), \tag{9.7}$$

where mt_planner is the multi-thread planning system with k instance of our experience-driven random trees ERT or ERTConnect planners.

We benchmark the different approaches for selecting multiple priors from the library following the same experimental setup as in the paper. We provide the library with a total of 10 experiences. Then, with these 10 experiences, we evaluate the performance achieved with single-threaded planning as described in the paper (**MO**), and the different experience selection methods for multi-threaded planning (**M1**, **M2**, **M3** and **M4**). Each method is evaluated in the four sets of scenarios defined in the paper (*Set 1*, *Set 2*, *Set 3* and *Set 4*), which involve increasing dissimilarity levels between experiences and planning queries. Figure 9.2 depicts the comparison between methods across the four sets of scenarios when employing a different number of threads. As it can be observed, **M1** is generally the worst approach to select multiple priors from the library; despite using k times the best-ranked experience, all planner instances explore the same region of the space. A boost of performance on the overall multi-thread planning system is achieved when using the k first best-ranked experiences (**M2**), as it allows each planner instance to explore different, yet similar to the problem, parts of the space. Leveraging the pre-computed references to guide the selection of experiences (**M3** and **M4**) also provides some extra performance for the overall system, especially when using **M4** in a higher number of threads.
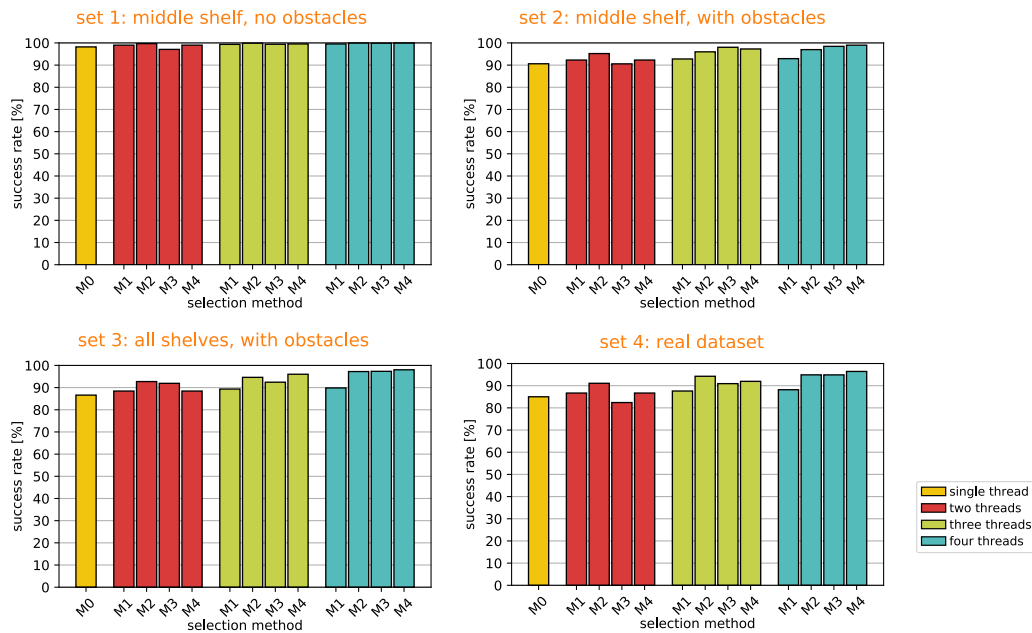

Figure 9.2: Performance comparison of single-thread planning (**M0**) and multi-thread planning with different approaches to select multiple experiences from the library (**M1**, **M2**, **M3** and **M4**). Each bar plot depicts the performance of the methods in the scenario sets (*Set 1*, *Set 2*, *Set 3* and *Set 4*) described in the paper.

Critical Review - Closing

In this part of the thesis, we have investigated two different strategies that generalise prior information, in the form of trajectories, to varied and significantly dissimilar task contexts. We started exploring strategies to extend the generalisation capabilities of policy-encoded behaviours to the presence of obstacles. Then, motivated by our findings, we formulated a sampling-based planning scheme that iteratively exploits a single prior path experience to ease the capture of connectivity of the space. To conclude this part, we provide a summary of the key results and discuss how the findings compare to those in the other parts of this thesis.

10.1   Results

Our efforts on exploiting prior information for efficient motion synthesis has resulted in two pieces of work. Each has been supported with experimental evaluation, in both synthetic environments and real-world robotic platforms. In particular, we provided the following results:

Chapter 8   We demonstrated our method extends the capabilities of DMP-encoded policies to safely adapt behaviour in the presence of obstacles in real-time. Extensive benchmark results highlighted the robustness and generalisation capabilities of the proposed approach regardless of the obstacle avoidance scenario. Finally, we demonstrated the suitability of our approach for robotic systems operating in challenging real-world environments, generalising obstacle avoidance behaviours to novel scenarios, even when those involve multiple obstacles, or are uniquely described by partial visual-depth observations.

Chapter 9   We demonstrated our ERT and ERTConnect planners offer a novel strategy to experience-based planning, which is able to cope with task instances that significantly differ from the prior experiences. We benchmarked our approach against relevant state-of-the-art planners; while related work fails to generalise priors of tens of experiences, our planner, with a single experience, significantly

outperforms them in both success rate and planning time. In this benchmark, we also included challenging real-world scenarios that showed the suitability of our approach for real-world robotic systems.

10.2 Discussion

Our work in the direction of generalising experiences to enrich the range of behaviour encoded in a library has resulted in two notable contributions:

Chapter 8   A hierarchical framework capable of safely modulating an ongoing DMP-encoded policy to avoid obstacles. Such capability is significantly relevant to the robotics community, given that DMPs are widely adopted in many robotic systems to learn task-relevant motions and to reproduce them with minimal generalisation capabilities. Thus, our work provides a strategy for these systems to extend their autonomous operation in contexts that include obstacles.

Chapter 9   Two new planners: the uni-directional experience-driven random trees (ERT) and its bi-directional version ERTConnect. These planners are extremely relevant to the community as, to the best of our knowledge, their underlying modus operandi is the less prior-requiring, and best well-performing experience-based planners in the literature. Also, we have discussed how to extract a suitable experience for such planners from a library of task-relevant motions.

Importantly, the proposed algorithms are not restricted to the presented experimental evaluation nor robotic platform. Any robotic system can benefit from our contributing algorithms to compute motion plans quickly by generalising prior knowledge to varied task instances. Both pieces of work have been open-sourced. Specifically, the ERT and the ERTConnect algorithms have been contributed in the OMPL for them to be of use to the robotics community at large.

Up to this point, the developments in this thesis frame robotic applications from which there exists a-priori information. In Part II [123, 5], we assumed similar enough task contexts that could be addressed with off-the-shelf policy encoded behaviours. In this part, we have enhanced the motion synthesis capabilities to succeed in leveraging prior information in significantly dissimilar tasks contexts. Next, in Part IV, we investigate the applicability of our findings in challenging, ever-changing planning problems that lack prior information.

We have discussed some directions for future work in the corresponding manuscripts. Among them, we are particularly keen on the possibility of extending our contributions to cope with multiple behavioural samples simultaneously. We elaborate on this thought in Chapter 15.

Part IV

GUIDING SYNTHESIS VIA ONLINE BEHAVIOURAL ABSTRACTIONS

Critical Review - Opening

Leveraging from behavioural abstractions has proved to be a powerful approach to rapid motion synthesis for tasks experienced a priori. At this point, we have explored such a strategy to address (i) similar tasks by directly bootsrapping behaviour (see Part II [123, 5]), and (ii) significantly dissimilar tasks by generalising behaviour (see Part III [124, 125]). The aim in this part of the thesis is to study the applicability of some of the findings introduced earlier to solve efficiently challenging path planning problems from which, mostly, there is no available prior knowledge.

11.1 Objectives and Contribution

There are some classes of planning problems whose instantiations are highly variable, thus complicating the pre-computation of relevant behavioural features. To motivate this fact, we consider a challenging ever-changing planning problem: mobile base navigation, in unknown environments, and in face of uncertainties. From a planning point of view, this is a remarkably hard problem as (i) the initially undiscovered environment is uniquely revealed in the proximity of the robot's exteroceptive sensors, (ii) the robot's manoeuvrability is constrained by its dynamics, and (iii) the robot's localisation, sensing and motion model suffer from uncertainties. All these constraints pose a complex, uncertain and high-dimensional problem in the belief space that is intractable with traditional planning from scratch strategies. Therefore, to enable a robot to navigate in such hostile environments, we need to formalise efficient online motion synthesis strategies. We present two notable contributions to this objective:

Chapter 12 A multi-layered planning strategy capable of solving complex, uncertain, and high-
(first)   dimensional problems online. We formalise a whole abstraction pipeline that projects
            the problem on a low-dimensional continuous support to retrieve a lead trajectory
            that hints at relevant navigation behaviour in the belief space.

Chapter 12 Efficient evaluation of uncertainties that, along with the multi-layered planner,
(second)   aids in making the problem tractable with: (1) recurrent pre-computation of
            environmental uncertainties, and (2) efficient state validation in the belief space.

11.2   Methodology

To tackle the challenge described above, we propose an uncertainty-based framework for mapping and planning feasible motions online with probabilistic safety guarantees. The proposed approach deals with the motion, probabilistic safety, and online computation constraints by: (i) incrementally mapping the surroundings to build an uncertainty aware representation of the environment, and (ii) iteratively (re)planning trajectories to a goal state that are kinodynamically feasible and probabilistically safe through a multi-layered sampling-based planner in the belief space. As described next, all contributions in this interdisciplinary framework are related to motion synthesis.

The multi-layered planning scheme within the proposed framework is built on the idea of problem abstractions. In particular, abstracting the problem online onto a known support allows us to induce some structure in such a highly-unstructured problem. Our planning scheme is two-layered. A first layer abstracts the problem on a low-dimensional and deterministic space to rapidly compute a lead, i.e., a trajectory that hints at relevant navigation behaviour. Specifically, we consider the support of this projection to be the workspace, enabling for on-the-fly computation of the lead using the RRT* planner. Then, in a similar vein as our planners in Chapter 9 [125], our scheme's second layer exploits the computed lead to guide the connectivity search towards relevant parts of the belief space. This exploration includes evolutionary heuristics to induce an adaptive lead-driven bias to the SST planner's search. Such heuristics ensure completeness guarantees despite biasing the search.

The multi-layered planning strategy enhances the focus of the search on relevant areas of the high-dimensional ever-changing space. However, the amount of time needed by a planner to find a solution is also highly dependant on the ability to quickly assess the validity of a state. State validity checking is generally a highly demanding computation process, and even more in face of uncertainties. To support the proposed multi-layered planner, we pose the stochastic state validity checking challenge as a process that can be approximated via abstraction. In particular, we project the continuous belief-space on a discrete support of variable resolution to efficiently compute the probabilistic collision between an uncertain map and uncertain states.

Online Mapping and Motion Planning under Uncertainty for Safe Navigation in Unknown Environments

In this chapter, we tackle the challenge of safe autonomous navigation in completely unknown environments. This is not a trivial task as robotic systems operating in these hostile environments must account for motion, localisation and sensory constraints, and their associated uncertainties, to plan safe navigation actions. Our approach is an uncertainty-based framework that incrementally maps and plans online, via an abstraction of the ongoing problem, feasible motions with probabilistic safety-guarantees. We demonstrate the properties of this framework via in-depth empirical analyses in simulation. Furthermore, real-world in-water experimental evaluation on a nonholonomic torpedo-shaped autonomous underwater vehicle and simulated trials in the DARPA Subterranean Challenge 2019 scenario on a quadrotor unmanned aerial vehicle illustrate the efficacy of the method, as well as its suitability for real-world systems with limited on-board computational power.

All proposed work is described in detail in the following published journal article:

Online Mapping and Motion Planning Under Uncertainty for Safe Navigation in Unknown Environments

Èric Pairet , *Associate Member, IEEE*, Juan David Hernández , *Senior Member, IEEE*, Marc Carreras , *Member, IEEE*, Yvan Petillot , *Member, IEEE*, and Morteza Lahijanian , *Member, IEEE*

Abstract—Safe autonomous navigation is an essential and challenging problem for robots operating in highly unstructured or completely unknown environments. Under these conditions, not only robotic systems must deal with limited localization information but also their maneuverability is constrained by their dynamics and often suffers from uncertainty. In order to cope with these constraints, this article proposes an uncertainty-based framework for mapping and planning feasible motions online with probabilistic safety guarantees. The proposed approach deals with the motion, probabilistic safety, and online computation constraints by: 1) incrementally mapping the surroundings to build an uncertainty-aware representation of the environment and 2) iteratively (re)planning trajectories to goal that is kinodynamically feasible and probabilistically safe through a multilayered sampling-based planner in the belief space. In-depth empirical analyses illustrate some important properties of this approach, namely: 1) the multilayered planning strategy enables rapid exploration of the high-dimensional belief space while preserving asymptotic optimality and completeness guarantees and 2) the proposed routine for probabilistic collision check-

ing results in tighter probability bounds in comparison to other uncertainty-aware planners in the literature. Furthermore, real-world in-water experimental evaluation on a nonholonomic torpedo-shaped autonomous underwater vehicle and simulated trials in an urban environment on an unmanned aerial vehicle demonstrate the efficacy of the method and its suitability for systems with limited onboard computational power.

Note to Practitioners—Emergent robotic applications require operating in previously unmapped scenarios. This article presents a unified mapping–planning strategy that enables robots to navigate autonomously and safely in harsh environments.

Index Terms—Field robotics, online mapping, online motion planning under uncertainty, safe autonomous navigation in unknown environments, sampling-based motion planning.

I. Introduction

AUTONOMOUS robots have been increasingly employed to assist humans notably in hazardous or inaccessible environments in recent years. Examples include rescue missions in disaster response scenarios [7], in-water ship hull [31] and wind turbine inspections [51], and deep underwater and space exploration [4], [74], among many others. A fundamental requirement for a robot engaged in any of these applications is to be adept at navigating autonomously through highly unstructured and hostile environments. However, this is not a trivial task due to a limited or complete lack of prior knowledge about the environment in which the robot has to operate. This implies that the robot has to base its decision-making on onboard sensors despite their limited accuracy. In addition, the robot itself might suffer from poor localization, as well as restricted and uncertain maneuverability. Therefore, even though challenging, it is essential to jointly consider all these motion and sensory constraints, as well as their associated uncertainties, when planning for navigation actions. This problem becomes particularly more challenging in safety–critical missions where the robot's safety must be ensured at all times.

Although there exist alternative methodologies addressing each of the abovementioned issues individually, limited attention has been devoted to the autonomous navigation problem in unknown environments as a whole [44]. The classical algorithms known as simultaneous localization and mapping (SLAM) enable a mobile robot to concurrently build and use a map to estimate its location [17]. These algorithms rely on identifying distinctive landmarks, which can bound the

uncertainty of both the environment representation and the robot localization. Nonetheless, even for scenarios rich in features, there are always some residual uncertainties. More recently, online motion planning frameworks have been developed to empower a mobile robot to compute navigation actions in unexplored environments while accounting for the system's motion capabilities, e.g., [11], [21], [27]–[29], [68], [76], [80]. These approaches, however, do not cope with any source of uncertainty and employ *ad hoc* heuristics that lack quantified safety guarantees. The few attempts to ensure safety through probabilistic methods, such as [12], [35], [71], are generally computationally expensive, built on strong assumptions, and commonly suppose a complete prior environment knowledge. Therefore, they are unsuitable for applications requiring online computations to deal with unknown environments.

In this context, our previous framework guaranteed (in compliance with a user-defined minimum probability of safety) the robot's safety when navigating through unexplored environments [55]. The underlying strategy consisted of an iterative mapping–planning scheme capable of continuously modifying the vehicle's motion plan toward the desired goal according to the incremental environmental awareness. At any time, the resulting motion plan was guaranteed to be feasible and safe in face of localization, mapping, and motion uncertainties. Despite the promising results achieved with this iterative mapping–planning scheme, its underlying formalization had some limitations. Namely, the framework was exclusively tailored to cope with a low-dimensional robot (three *degrees of freedom* (DoFs)) navigating in an unknown, symmetrically structured, 2-D workspace. The initially proposed mapping–planning scheme and its constituent components would scale poorly when dealing with systems and scenarios of higher complexity. More demanding problems exacerbate the curses of dimensionality and computational load to guarantee probabilistic safeness in face of uncertainties. Such a challenge motivated the development of this follow-up work to extend the framework's capabilities to suit the requirements of a larger group of robotic systems and environments.

Building on our previous mapping–planning scheme [55], the main contribution of this article is threefold.

1) Multilayered planning strategy capable of rapid search in high-dimensional belief spaces, with asymptotical optimality and probabilistic completeness guarantees.

2) Probabilistic map fusion that efficiently retrieves environmental uncertainties in form of a cumulative map, while dealing with overlapping local submaps.

3) Probabilistic collision checking routine, which rapidly evaluates the validity of a state subject to uncertainties by trading the tightness of the safety bound for computational efficiency, while accounting for the tail events.

Our new contributions in the framework's key constituent components are supported with rigorous theoretical development and thorough experimental evaluations. These novel advancements allow for faster online motion planning and more efficient evaluation of uncertainties. Consequently, the improved framework is now capable to compute navigation actions online for high-dimensional systems and more challenging unknown environments while providing safety guarantees. To the best of our knowledge, this is the first generic architecture capable of jointly dealing with kinodynamic and probabilistic constraints in unknown environments online. Both the precedent and new framework are analyzed and compared in multiple scenarios with different interesting real-world[1] and simulated[2] physical systems. The experimental results demonstrate the suitability of the proposed method to address the challenge of probabilistically safe autonomous navigation in unknown environments while being suitable for systems with limited onboard computational power.

The remainder of this article is organized as follows. Section II provides a comprehensive review of the literature and the corresponding contribution of this article. Then, Section III formally defines the considered problem. In Section IV, an overview of the framework is presented, and then, the mapping and planning components are detailed in Sections V and VI, respectively. The description of the framework is followed by a thorough analysis of its key constituent features and its performance and capabilities as a whole in Section VII. Finally, this article concludes with a discussion in Section VIII.

II. RELATED WORK

This section gives a brief overview of prior work on planning under kinodynamic constraints and planning under uncertainty, as well as frameworks for online mapping–planning. Finally, this section discusses all contributions of this work with respect to the latest related literature.

A. Planning Under Kinodynamic Constraints

Planning under kinodynamic constraints deals with the challenge of computing trajectories that are feasible according to the vehicle's motion capabilities. This problem is commonly formulated as finding a trajectory between two points through the system's state space. The robotics literature offers various approaches to tackle this problem.

One strategy is to represent the continuous state space as a lattice space, i.e., a graph where edges correspond to a reduced set of precomputed motion primitives. Then, the motion planning problem can be efficiently solved using graph search algorithms. For the particular case of a car-like system, the motion primitives can be defined as a set of lines and arcs to build a geometric state lattice [16], [67]. These approaches can find the shortest path, but the transition between segments presents abrupt changes in angular velocity, which could only be achieved by a system capable of infinite angular acceleration. More complex lattice space definitions allow the consideration of more restrictive concatenation rules and richer sets of primitive motions, e.g., [20], [62], at the cost of more memory usage and more computationally expensive queries. Even though planning in lattice spaces has proven to be suitable for many applications, it requires the crafting of a set of motions such that the resulting lattice offers,

---

[1] A mission through a real breakwater structure with an autonomous underwater vehicle (AUV) can be seen in: https://youtu.be/dTejsNqNC00

[2] A mission in the DARPA Subterranean Challenge 2019 scenario with an unmanned aerial vehicle (UAV) can be seen in https://youtu.be/I5X_QFKDpeI

at least, one suitable solution to the planning problem. Some works in the learning community have addressed this difficult and time-consuming task with data-driven techniques [14]. However, the resulting set of motions still represents a very limited range of the real dynamic capabilities of the robot. This is undesirable in applications where the environment is not known in advance, and where having the entire dynamic range of motions available for planning can be critical to finding a suitable solution. All in all, lattice-based methods struggle with planning in high-dimensional state spaces.

To deal with kinodynamic constraints, sampling-based motion planners offer great opportunities, e.g., [32], [36], [38]. Most sampling-based planners, however, lose their asymptotic optimality guarantees when a steering function does not exist in the system's kinodynamically constrained state space. To cope with this limitation, there are different assumptions and heuristics that can be applied at the expense of longer computational times. For example, Webb and van den Berg [78] proposed a version of the asymptotic optimal RRT (RRT*) that can deal with kinodynamic constraints of systems with linearisable dynamics [78]. If the system's dynamics are not linearisable, asymptotic optimality can be obtained in any planner by augmenting the dimensionality of the state space to account for the search cost [25]. However, this strategy implies solving the planning problem repeatedly to improve the cost of the solution at each iteration, consequently being unsuitable for applications with online requirements. Finally, the stable sparse RRT (SST) planner offers asymptotically near-optimal guarantees by means of a shooting approach, which consists of expanding the tree from the node with the lowest cost within a neighborhood of predefined δ-radius [41].

Planning in high-dimensional spaces with multiple constraints poses a challenge for classical planners and, typically, results in long computation times if a solution can be found at all. In such problems, a common approach to boost performance is via a multilayered planning scheme. The key idea is to leverage from a lead to guide (warm-start) the search. In this regard, an interesting approach is the incremental trajectory optimization for motion planning (ITOMP) algorithm, which interleaves planning and optimization; the planner is given a fixed time budget to find a solution, which is then used as a warm-start for the optimizer [58]. Work in [63] and [64] introduced a synergistic three-layered planner: the high-level planner uses discrete search to initially determine those candidate regions (from a decomposed representation of the environment), which might contain part of the final solution; a low-level planner employs a sampling-based motion planner to find a solution; and a middle layer updates the candidate regions according to the considered constraints. However, the proposed combination of planners does not guarantee asymptotic optimality, and the discrete planner becomes slow for high-dimensional problems. Palmieri *et al.* [57] presented the Theta*-rapidly exploring random tree (RRT) scheme, which first uses the Theta* path planner to compute a lead path, which is then employed to bias the search of the RRT planner [57]. This approach, however, lacks asymptotic optimality guarantees, given that the second planner is an RRT.

More recently, a multilayered approach based on the RRT* as a lead planner and the SST as the final planner has been proposed in [76]. The final planner's search space is strictly constrained around the lead path, raising concerns about the completeness guarantees of the overall architecture.

B. Planning Under Uncertainty

An essential capability for any autonomous robot is to operate in the presence of uncertainty [13]. Sources of uncertainty relevant to autonomous systems fall into four types [39].

1) *Uncertainty in Localization:* The robot's location is uncertain with respect to the environment. This issue is particularly critical in robots operating in GPS-denied environments or for systems suffering from low-accuracy state estimation.
2) *Uncertainty in Motion (Dynamics):* The future robot state cannot be predicted accurately, either because of discrepancies between the considered and the real system's dynamic behavior or due to limited precision in the system's command tracking.
3) *Uncertainty in the Environmental Awareness:* The robot has inexact or incomplete information about its surroundings (e.g., obstacle location). This issue can arise from inaccuracies in the *a priori* map, or imperfect and noisy exteroceptive sensory capabilities.
4) *Disturbances in the Operational Environment:* The robot is subject to external factors, such as wind, atmospheric turbulences, or water currents, which makes the robot deviate from the planned trajectory, thus compromising the reliability of deterministic path planning techniques.

This section scrutinizes relevant planning strategies dealing with any of the three first sources of uncertainty. Given the scope of our work, terrain traversability analysis methods (e.g., [19], [24], [54]) are excluded from this review.

One approach that is popular among existing planners is based on discrete Markov processes. This strategy models the evolution of the system in the environment and generates a policy over the approximated Markov states. Examples of such motion planners include stochastic motion roadmap (SMR) [5] and incremental Markov decision process (iMDP) [33]. These methods have shown to be effective and provide optimality guarantees in terms of the probability of reaching the desired goal; however, they assume perfect knowledge about the environment. Works such as [47] have extended these techniques to partially unknown environments. Nonetheless, their large computational times remain the main hurdle in applications with fully unknown environments or requiring online planning.

Another approach to deal with uncertainties in planning is by means of feedback controllers and sampling-based planners. Van Den Berg *et al.* [75] proposed the linear quadratic Gaussian (LQG) motion planning method, which finds the best path simulating the performance of LQG on all extensions of an RRT [75]. This idea was later applied in roadmaps to propose the feedback-based information roadmap (FIRM) [1]. This method, though, relies on full *a priori* awareness of the environment to explore the belief space offline and then to quickly perform queries online. Consequently, this strategy

is not suitable for planning applications where the *a priori* information about the environment, if available, is not fully informative. A similar strategy is used in [2] for simultaneous localization and planning. Alternatively, Sun *et al.* [73] presented the high-frequency replanning (HFR) architecture, a strategy that leverages from an LQG and a multithread RRT, allowing to continuously replan in the face of alterations in the robot or environment space, while accounting for uncertainties. However, the asymptotic optimality guarantees of such a method can only be assured in multithreaded implementations.

An alternative approach to dealing with uncertainty is the chance-constraint strategy. In these methods, instead of maximizing the probability of success, the objective is to find a path that satisfies a minimum safety probability constraint. The challenge in incorporating this method in planners lies in the computation of the safety probability over plans. In [9], linear chance constraints are combined with disjunctive linear programming to perform probabilistic convex obstacle avoidance. This concept was extended and integrated into a sampling-based planner, leading to the chance constrained RRT (CC-RRT) [45] and the CC-RRT* [46]. These approaches evolve the system's dynamics in an open-loop fashion, hence growing the uncertainty unboundedly forward in time. To improve accuracy, linear chance constraints were applied after propagation of the system's state conditioned on the precedent states being collision-free [59]. Such a strategy is commonly referred to as truncating the distribution estimating the system's state, and its usage in planning led to the CC-RRT*-D planner [43]. The advantage of chance-constraint-based methods is that satisfying plans can be computed quickly, making them desirable for online applications. They are, however, built on strong assumptions that result in overly conservative calculations and rely on the prior knowledge of a convex environment. Nonetheless, chance-constraint methods are still widely used in the planning community to deal with localization, motion, and environmental uncertainties, e.g., [12], [71].

In recent years, planners based on various discretization methods have been developed to deal with limited computational power or online planning requirements in face of uncertainty. Majumdar and Tedrake [48] proposed a precomputed library of funnels to efficiently estimate the system's kinodynamic and uncertainty propagation in 3-D environments [48]. However, library-based approaches consider a reduced set of the real system's capabilities that can endanger the efficacy of the planner. Another approach in favor of performance consists of approximating the computation of the probability of collision to a discrete support [55], [71]. This strategy truncates the infinite expansion of the belief in a bounded patch considered to contain a large portion of the belief's probability mass. In our previous work [55], all uncertainties were projected onto discrete support, referred to as kernel, whose resolution resembled the optimal one for online mapping applications. Although considering discrete support for the computation of the probability of collision allows for quick calculations, none of the works using such technique actually normalizes the calculations for the probability mass laying outside the patch,

i.e., tail events. Therefore, they cannot offer guarantees on the compliance of the probabilistic safety constraints.

C. Frameworks for Online Mapping–Planning

Limited attention has been devoted to the navigation problem as a whole, especially in the face of uncertainties. Note that the navigation requirements differ from those of coverage path planning, for which there is a perpendicular literature thread, e.g., [23], [37]. Current navigation frameworks in the robotics literature are built on strong assumptions, which could endanger (or completely neglect) some of the essential requirements for safe navigation in undiscovered environments. Some of the prerequisites are the ability to create an uncertainty-aware representation of the environment such that uncertainties about the environment can be considered at the planning stage. It is also crucial to ensure completeness guarantees, i.e., the ability of finding a solution if one exists, and among many others, being capable of guaranteeing the vehicle's safety at any time during the mission. Ideally, an online mapping–planning framework should be able to find paths quickly while offering asymptotic optimality guarantees.

A common strategy for online navigation is to continuously replan in the face of changes in the robot's pose or the environment awareness. Scherer *et al.* [69] endowed an UAV with the capability to map online with an occupancy probabilistic grid and then to guide itself toward the goal with a combination of global and local potential field-based planners [69]. Along this line, navigation in 3-D environments by mapping from stereo vision and planning with the RRT was considered in [6]. The resulting paths of these approaches do not account for kinodynamic constraints or safety guarantees. Alternatively, in [42], the local planner of the RRT approximated an UAV capabilities by 3-D Dubins paths. Nevertheless, none of these approaches considers any of the multiple sources of uncertainty in the mapping nor the planning stage, thus not providing any theoretical performance or safety guarantees.

More recently, Ho *et al.* [29] proposed an online framework to build an uncertainty-aware map and plan over it using the RRT. However, the resulting paths do not meet kinodynamic nor safety constraints. Instead, proposals in [27] and [28] presented an online framework to plan paths under motion constraints for AUVs, but their approach assumes zero uncertainty. While their framework succeeded in solving start-to-goal queries in unexplored real-world environments, their planner used *ad hoc* heuristics to estimate the solution's associated risk and approximated the system's dynamics with Dubins curves. Frameworks can employ multilayered schemes to scope the complexity of online constrained planning in a subregion of the entire planning space, e.g., [18], [76]. Youakim *et al.* [80] presented a multirepresentation, multiheuristic A* planner capable of jointly dealing with mobile-base and manipulation planning in unknown environments while accounting for localization uncertainty via heuristics. Despite all methods have been tested in real-world environments, the underlying frameworks lack of theoretical analysis and do not provide a measure of robustness or quantified safety guarantees.

D. Closely Related Contributions

An early version of the work presented in this manuscript has appeared before [55]. This consisted of a simpler framework that proved to be suitable for real-world motion planning problems, but its applicability was strictly limited to underwater robots operating at constant depth, i.e., 2-D workspaces. This motivated the development of this follow-up work to extend the framework's capabilities to suit the requirements of a larger group of robotic systems and environments. Overall, given the precedent efforts by the authors, this article provides the following contributions.

1) An online mapping–planning framework that probabilistically guarantees the robot safety during navigation tasks in unknown environments (see Section IV).

2) A mapping strategy using local submaps that builds an uncertainty-aware map (Section V). These calculations now, in contrast to [55], include efficient retrieval of environmental uncertainties and consider probabilistic map fusion to deal with overlapping local submaps.

3) A multilayered planner (MLP) that guides the search in the high-dimensional belief space (Section VI), in contrast to the uniform search of the single-layered planner in [55]. Our planner satisfies kinodynamic constraints and probabilistic safety guarantees while providing probabilistic completeness and asymptotic optimality guarantees.

4) A rapid probabilistic collision checking routine (Section VI-C). In contrast to [55], the calculations now include a controllable confidence level α that allows to trade the tightness of the safety bound for computational efficiency while correcting for the tail events (i.e., the probability mass excluded by the confidence level).

5) A thorough evaluation of the whole framework and its key constituent components (see Section VII). Besides robot deployments on challenging real-world environments, this assessment, in contrast to [55], now includes rigorous analysis on different scenarios and dynamical systems.

Our contributed advancements allow for faster online motion planning and more efficient evaluation of uncertainties. Consequently, the framework now can compute probabilistically safe navigation actions online for high-dimensional systems and more challenging unknown environments.

III. Problem Formulation

In this work, the focus is on the challenging problem of safe autonomous navigation in unexplored environments. To start with, the robotic system must be capable of perceiving and creating a consistent representation of the surroundings despite its potentially uncertain localization. The perceived surroundings must be encoded efficiently such that the robot can exploit them online for planning purposes. Besides the mapping requirements, the process of planning navigation actions toward the desired goal is challenging by itself. The robot must not only account for its limited and uncertain maneuverability but also for the evolving awareness and uncertainty of the

TABLE I

Summary of the Nomenclature in This Article

Symbol	Description
Generic definitions	
$\mathcal{W}$	workspace
$\mathcal{X}$	state space
$\mathcal{B}$	belief space
$\mathcal{U}$	control space
$b_k = \mathcal{N}(\hat{\mathbf{x}}_k, \Sigma_{\mathbf{x}_k})$	state belief at time k
$b_A^B = \mathcal{N}(\hat{\mathbf{x}}_A^B, \Sigma_A^B)$	state belief of A as seen from B
Framework	
W	global frame
R	robot base frame
R'	planning frame
Mapping	
$\mathcal{M}$	probabilistic map awareness
$\mathcal{LM}$	local submap
$F_O(\mathbf{x})$	occupancy probability at $\mathbf{x}$
$F_{\mathcal{X}}$	cumulative map over $\mathcal{X}$
Planning	
b_{start}	estimated system state at start
$\mathcal{X}_{\text{goal}}$	goal region in state space
$\mathcal{B}_{\text{goal}}$	goal region in belief space
p_{goal}	minimum probability of goal
p_{safe}	minimum probability of safety
ΔT_{MP}	overall planning budget time
ΔT_L	lead planner budget time
ΔT_C	constrained planner budget time
ξ'	lead geometric path
$\mathcal{X}'$	lead region in state space
ξ	feasible and safe trajectory

surroundings as the robot moves. This section provides formal definitions for these uncertainties and the problem of safe autonomous navigation in unexplored environments. Table I summarizes the nomenclature used through this article.

A. Motion Uncertainty and Constraints

Consider a mobile robot that operates in a workspace $\mathcal{W} \subset \mathbb{R}^{n_w}$, where $n_w \in \{2, 3\}$, under motion uncertainty. The uncertainty in the robot's motion can be due to many reasons, e.g., unmodelled dynamics or noise in actuation, and can be described in several ways. In this work, inspired by [8], [26], [40], and [52], the evolution of the uncertain robotic system is assumed to follow a Gaussian process. That is, the robot state $\mathbf{x}_k$ at every time step k is defined by a Gaussian distribution

$$\mathbf{x}_k \sim b_k = \mathcal{N}(\hat{\mathbf{x}}_k, \Sigma_{\mathbf{x}_k}) \tag{1}$$

where b_k is referred to as the *belief* of $\mathbf{x}_k$ and is fully defined by mean $\hat{\mathbf{x}}_k$ and covariance $\Sigma_{\mathbf{x}_k}$. The set of all beliefs is called the belief space and denoted by $\mathcal{B}$. Intuitively, $\mathcal{B}$ is an uncertain representation of the state space $\mathcal{X}$. Mean $\hat{\mathbf{x}} \in \mathcal{X} \subseteq \mathbb{R}^{n_x}$ is the nominal state of the robot and evolves according to

$$\hat{\mathbf{x}}_{k+1} = f(\hat{\mathbf{x}}_k, \mathbf{u}_k) \tag{2}$$

where $f : \mathcal{X} \times \mathcal{U} \rightarrow \mathcal{X}$ captures the nominal (known) dynamics of the robot, and $\mathbf{u}_k \in \mathcal{U} \subset \mathbb{R}^{n_u}$ is the system's control input. Covariance $\Sigma_{\mathbf{x}_k} \in \mathbb{R}_{>0}^{n_x \times n_x}$ describes the uncertainty

around the nominal robot state and evolves according to

$$\Sigma_{\mathbf{x}_{k+1}} = g\left(\Sigma_{\mathbf{x}_k},\ \mathbf{u}_k\right) \tag{3}$$

where $g : \mathbb{R}^{n_x \times n_x} \times \mathcal{U} \to \mathbb{R}^{n_x \times n_x}_{>0}$ is the covariance function. Examples of Gaussian processes for robots with unicycle and fixed-wing dynamics are provided in Appendix A. Methods for modeling robots with (partially) unknown dynamics as Gaussian processes are discussed in [34] and [52].

B. Environment Uncertainty

Some applications in robotics lack complete awareness of the environment, either because there is no information about the surroundings or due to the presence of dynamic elements in the workspace. This work scopes the mapping requirements to undiscovered static environments. In order to reveal the obstacles in the environment, the robot is equipped with exteroceptive sensors such that it can autonomously explore the surroundings as it moves, i.e., to integrate into the map the obstacles when they are inside the sensor's detection range. Importantly, most sensors uniquely detect points on the boundary of a nearby obstacle.

This work assumes no uncertainty in the robot local observations denoted by $\mathbf{h}_k$. To transform this local observation from the robot frame to the global frame, let $\mathbf{h}_k \sim \mathcal{N}(\hat{\mathbf{h}}_k,\ 0)$. Bearing in mind that the robot's location might be uncertain with respect to the global frame $b_k \sim \mathcal{N}(\hat{\mathbf{x}}_k,\ \Sigma_{\mathbf{x}_k})$, the observed point is represented in the global frame as

$$b_O = b_k \oplus \mathbf{h}_k \tag{4}$$
$$= \mathcal{N}\left(\hat{\mathbf{x}}_k \oplus \hat{\mathbf{h}}_k,\ \mathbf{J}_{1\oplus}\Sigma_{\mathbf{x}_k}\mathbf{J}_{1\oplus}^T\right) \tag{5}$$

where $b_O \sim \mathcal{N}(\hat{\mathbf{x}}_O,\ \Sigma_{\mathbf{x}_O})$ is the result of the Gaussian relationships via a compounding operator $\oplus$ explained in Appendix B. From these uncertain points $\mathbf{x}_O$, the robot constructs a probabilistic map $\mathcal{M}$. Then, the obstacle occupancy probability for point $\mathbf{x} \in \mathcal{X}$ denoted by $F_\mathcal{X}(\mathbf{x})$ is the sum of the normally distributed densities in $\mathcal{M}$. The cumulative sum over all space $\mathcal{X}$ is called cumulative map and denoted by $F_\mathcal{X}$.

C. Probabilistic Safety Guarantees

The system's and the environment's uncertainty are jointly considered to guarantee the vehicle's safety. More specifically, the probability of the system being in collision with an obstacle in the environment at time k is characterized by

$$p_{\text{collision}}(b_k,\ \mathcal{M}) = \int_\mathcal{X} b_k(\mathbf{x})\,F_\mathcal{X}(\mathbf{x})\,d\mathbf{x}$$
$$= \int_\mathcal{X} \mathcal{N}\left(\mathbf{x}\,|\,\hat{\mathbf{x}}_k,\ \Sigma_{\mathbf{x}_k}\right) F_\mathcal{X}(\mathbf{x})\,d\mathbf{x} \tag{6}$$

where $F_\mathcal{X}(\mathbf{x})$ is the cumulative obstacle occupancy probability, as introduced in Section III-B. Then, given a minimum probability of safety p_{safe}, we require $1 - p_{\text{collision}}(b,\ \mathcal{M}) \geq p_{\text{safe}}$ for every belief b on the trajectory in order to probabilistically guarantee the robot's safety.
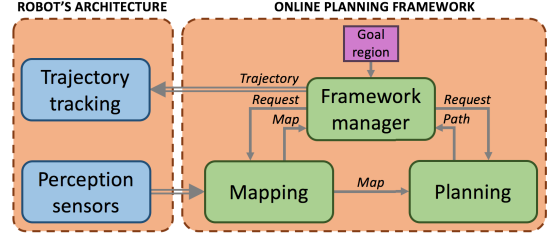

Fig. 1. Framework for online mapping and motion planning under kinematic and uncertainty constraints.

D. Planning Problem

Therefore, the planning problem considered in this work seeks a dynamically feasible trajectory in the belief space $\mathcal{B}$, which is probabilistically safe. Formally, let $\mathcal{B}_{\text{goal}} \subset \mathcal{B}$ denote the set of all belief states that correspond to the desired goal region $\mathcal{X}_{\text{goal}}$ in the environment as

$$\mathcal{B}_{\text{goal}} = \left\{ b \in \mathcal{B} \,\middle|\, \int_{\mathcal{X}_{\text{goal}}} b(\mathbf{x})\,d\mathbf{x} \geq p_{\text{goal}} \right\} \tag{7}$$

where p_{goal} is the minimum probability that a belief must satisfy for being considered to be in the goal region. Then, the constrained planning problem is to compute a sequence of controls $\mathbf{u}_0, \mathbf{u}_1, \ldots, \mathbf{u}_{T-1} \in \mathcal{U}$ that result in a dynamically feasible trajectory $\xi : [0,\ T] \to \mathcal{B}$ for the robotic system described by (1), (2), and (3) such that $\xi(0) = b_{\text{start}} \in \mathcal{B}$, i.e., the system state at the beginning of the mission, $\xi(T) \in \mathcal{B}_{\text{goal}}$, and $1 - p_{\text{collision}}(\xi(t),\ \mathcal{M}) \geq p_{\text{safe}}$ for all $t \in [0,\ T]$.

IV. FRAMEWORK FOR ONLINE NAVIGATION

This article presents a framework that endows a robotic system with the capability of safely navigating through unknown environments. This is achieved by means of online mapping and online motion planning of trajectories that meet motion and probabilistic constraints. The framework, depicted in Fig. 1, is threefold: 1) a mapping module that incrementally builds an uncertainty-aware map; 2) a planning module that continuously computes a safe and feasible trajectory toward the goal; and 3) a framework manager that coordinates the overall framework's execution. The remainder of this section describes the manager's strategy to control the interaction between the two core modules of the framework, i.e., the mapping (see Section V) and the planning (see Section VI). Note that, although the framework's description focuses on the online navigation challenge, the proposed online scheduling intrinsically solves the off-line motion planning problem.

The framework manager coordinates the mapping (MAPPER) and planning (PLANNER) modules according to the pipeline presented in Algorithm 1. This is, given the desired goal region $\mathcal{B}_{\text{goal}}$ and the required probabilistic safety guarantees p_{safe}, the manager conducts an iterative process until the system reaches the predefined goal region (line 7). An iteration consists of solving an updated version of the underlying motion planning problem that accounts for any alteration to the system's state and environment awareness.

Each iteration starts by predicting a planning frame R' that corresponds to the robot state at the time the current iteration solution will be available (line 8). Calculating a suitable planning frame is essential to guarantee the continuity and feasibility between consecutive plans. As the framework iterates each ΔT_{MP} and it has full knowledge of the plan in execution *ongoing_traj*, calculating a planning frame that is suitable after ΔT_{MP} is formulated as a state prediction problem. This is, given the current robot state $\mathbf{x}_R^W$ and the set of subsequent controls $\mathbf{u}$ involved in the execution of *ongoing_traj*, $\mathbf{x}_{R'}^W$ is computed by integrating (2) and (3) for the time-horizon ΔT_{MP}. Then, the manager retrieves, from the MAPPER, the current environment awareness as a cumulative map $F_{\mathcal{X}}^{R'}$ relative to R' (line 9). Both the predicted planning frame R' and the updated cumulative map $F_{\mathcal{X}}^{R'}$ are provided to the PLANNER (lines 10 and 11).

Before proceeding to solve the updated planning problem, the current plan in execution *ongoing_traj*, if any, is probabilistically checked for collision according to the current uncertainty-aware map $F_{\mathcal{X}}^{R'}$. In the event of *ongoing_traj* not being any longer valid, the framework manager dispatches to the robot the segment $\overline{\textit{ongoing\_traj}}$ of *ongoing_traj* that is still safe (lines 12 and 13). This approach prevents the vehicle from stopping every time that a trajectory gets partially invalidated while ensuring its safety.

Finally, the PLANNER attempts to solve the planning problem by growing a new tree in $\mathcal{B}$ for a specific amount of time ΔT_{MP} (line 14). The PLANNER tries to find a near-optimal trajectory that meets kinematic and probabilistic constraints within the allocated time budget ΔT_{MP} and returns a *new_traj* if one is found (line 15). The newly found *new_traj* is uniquely dispatched to the robot when it fulfills the selection criteria defined in satisfiesCriteria() (sline 16–18). This work bases the selection criteria satisfiesCriteria() on the length of the trajectory; *new_traj* is dispatched if length(*new_traj*) ≤ length(*ongoing_traj*), where length(*ongoing_traj*) = ∞ if *ongoing_traj* is partially invalidated, i.e., it does not reach the goal region $\mathcal{B}_{\text{goal}}$.

Note that the computations in lines 8 and 9 are low demanding, and they can be scheduled in parallel to the main execution of the framework's pipeline. Therefore, the overall iteration rate of the framework is $1/\Delta T_{\text{MP}}$, as solving the planning problem (line 14) is the unique process of the framework that requires a nonnegligible amount of time.

Given the nature of the problem of navigation in unknown environments, it may be possible that a feasible and probabilistically safe trajectory toward the goal region does not exist. Therefore, the framework is endowed with a contingency plan that attempts to return the vehicle nearby the deployment location b_{start}. This contingency plan gets activated when the planner has not been able to find a solution in the last n_{cp} consecutive iterations, where n_{cp} is a user-defined safety value. In the event of the contingency plan getting activated, the MANAGER is reinitialized with the new planning problem. Note that, if the environmental awareness is highly uncertain, there might not exist a trajectory toward the new goal region. In this situation, not considering the previous map information for planning would allow the vehicle to move safely toward

---

Algorithm 1 MANAGER($\mathcal{B}_{\text{goal}}$, p_{safe})

1 **Input:**
2 $\mathcal{B}_{\text{goal}}$: Goal region
3 p_{safe}: Required probabilistic safety guarantees
4 **begin**
5 $\quad$ *ongoing_traj* $\leftarrow \varnothing$
6 $\quad$ PLANNER.loadProblem($\mathcal{B}_{\text{goal}}$, p_{safe})
7 $\quad$ **while not** isGoalAchieved() **do**
$\quad\quad$ /* Predict planning frame   */
8 $\quad\quad$ $R' \leftarrow$ pedictFrame(*ongoing_traj*)

$\quad\quad$ /* Retrieve cumulative map   */
9 $\quad\quad$ $F_{\mathcal{X}}^{R'} \leftarrow$ MAPPER.getMap(R')

$\quad\quad$ /* Update planning problem   */
10 $\quad\quad$ PLANNER.setNewFrame(R')
11 $\quad\quad$ PLANNER.updateMap($F_{\mathcal{X}}^{R'}$)

$\quad\quad$ /* Check ongoing plan   */
12 $\quad\quad$ **if not** PLANNER.isValid(*ongoing_traj*) **then**
13 $\quad\quad\quad$ dispatchPath($\overline{\textit{ongoing\_traj}}$)

$\quad\quad$ /* Solve planning problem   */
14 $\quad\quad$ PLANNER.solve(ΔT_{MP})

$\quad\quad$ /* Dispatch best valid plan   */
15 $\quad\quad$ *new_traj* $\leftarrow$ planner.getSolution()
16 $\quad\quad$ **if** satisfiesCriteria(*new_traj*) **then**
17 $\quad\quad\quad$ *ongoing_traj* $\leftarrow$ *new_traj*
18 $\quad\quad\quad$ dispatchPath(*ongoing_traj*)

---

the deployment location. In case no feasible motion plan is found to return to the deployment location, an emergency maneuver should be performed, e.g., coming to a complete stop for ground vehicles, going to the water surface for AUVs, and immediate landing for UAVs. Alternatively, one might consider ensuring the existence of a contingency plan at any time as in [22], but doing so under uncertainties is not trivial.

V. INCREMENTALLY MAPPING UNKNOWN ENVIRONMENTS VIA LOCAL MAPS

Incrementally exploring the environment with a location-uncertain system leads to an uncertain representation of the surroundings. Under these conditions, obtaining a consistent and reliable representation of the entire environment is a challenging task commonly addressed with probabilistic inference approaches. These algorithms rely on gathering data from which distinctive features (landmarks) can be extracted and used to bound the uncertainty of the environment representation and system localization. Nonetheless, even for scenarios rich in features, there are always some residual uncertainties. Moreover, onboard perception sensors usually suffer from noises, which compromises the accuracy of the environment representation. All these issues motivate the need for an environment representation that jointly explains captures the
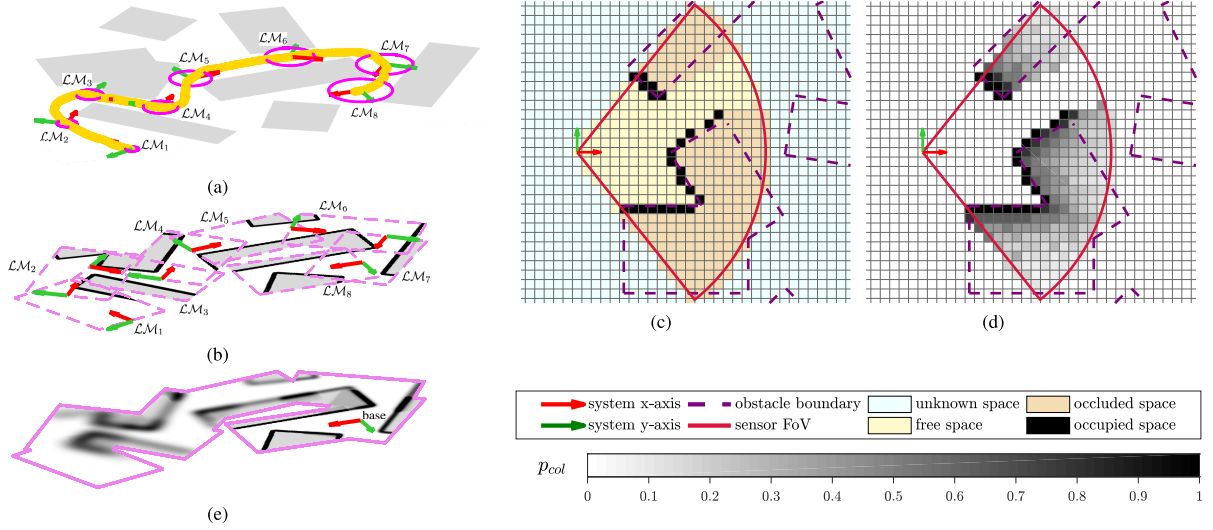
Fig. 2. Online mapping of the environment suitable for motion planning under uncertainty. (a) As the robot navigates through an unknown environment, (b) it builds a probabilistic map $\mathcal{M}$, which represents the surroundings as a set of local maps. Each local map is uncertain [magenta ellipses in (a)] with respect to the global frame. (c) and (d) Each local map is encoded as an occupancy grid map, which is built from a set of sequential sensor scans over a finite horizon time. (e) All local maps are fused into a cumulative map representation $F_{\mathcal{X}}^{R'}$ taking into account the relative uncertainty of each local map with respect to the predicted planning frame R'. (a) Environment. (b) Probabilistic map $\mathcal{M}$. (c) Semantic scene understanding. (d) Probabilistic scene understanding. (e) Cumulative map $F_{\mathcal{X}}^{R'}$. Note that our mapping suits 3-D scenes too and illustrations in 2-D for visualization purposes only.

uncertainty on the true obstacle's localization and the detection confidence according to the sensor model while being suitable for motion planning. In this work, such a representation is referred to as a probabilistic map.

This section details the undertaken mapping approach, which builds a set of local occupancy submaps whose base poses are uncertain with respect to a global frame (see Section V-A). Each submap is an occupancy grid map, which provides an efficient strategy to encode the incremental environment awareness (see Section V-B) and retrieve information about the environment occupancy (see Section V-C and Section V-D). This overall mapping strategy has proven to be suitable for real-time robotic mapping and planning applications in our previous work [55] and, despite being out of the scope of this manuscript, has also shown to be effective for online mapping and localization applications [29].

A. Global Map as a Set of Local Submaps

There are different alternatives to represent the incremental knowledge of an environment, e.g., [3], [53], [66], [77]. The framework presented in this manuscript encodes the environment $\mathcal{M}$ via a set of n local stochastic submaps [60], [61] due to its demonstrated efficiency on dealing with applications requiring real-time robotic localization, mapping, and planning [29], [55]. Formally, the local submaps method is defined as

$$\mathcal{M} = \{\mathcal{LM}_1, \ldots, \mathcal{LM}_n\} \tag{8}$$

$$\mathcal{LM}_i = \left\{ \{\mathbf{v}_1, \ldots, \mathbf{v}_m\}, \ \hat{\mathbf{x}}_{\mathcal{LM}_i}^W, \ \Sigma_{\mathcal{LM}_i}^W \right\} \tag{9}$$

where each local submap $\mathcal{LM}_i$ contains a set of sequential sensor scans over a finite horizon time $\Delta T_{\mathcal{LM}}$. Within this

time period, all point coordinates $\mathbf{v}$ of the sensed environment are registered into the active submap $\mathcal{LM}_n$. The coordinate frame of $\mathcal{LM}_n$ is defined in a global frame W by its estimated state $\mathbf{x}_{\mathcal{LM}_n}^W \sim \mathcal{N}(\hat{\mathbf{x}}_{\mathcal{LM}_n}^W, \Sigma_{\mathcal{LM}_n}^W)$. Importantly, such local registration assumes null uncertainty on observations, i.e., $\Sigma_{\mathbf{v}}^{\mathcal{LM}_n} = \mathbf{0} \ \forall \ \mathbf{v} \in \mathcal{LM}_n$. A new local submap $\mathcal{LM}_{n+1}$ is initiated every $\Delta T_{\mathcal{LM}}$ such that the accumulated localization error within the active local submap $\mathcal{LM}_n$ is low. In other words, the local mapping time horizon $\Delta T_{\mathcal{LM}}$ must be defined such that it always maintains the robot pose uncertainty $\Sigma_R^{\mathcal{LM}_n}$ within the active local map $\mathcal{LM}_n$ negligible.

The coordinate system of a new local submap $\mathcal{LM}_{n+1}$ is defined at the robot state estimate when $\mathcal{LM}_{n+1}$ is initiated, i.e., $\mathbf{x}_{\mathcal{LM}_{n+1}}^W = \mathbf{x}_R^W$. It is assumed that the robot starts building $\mathcal{LM}_{n+1}$ as soon as it finishes the $\mathcal{LM}_n$. Therefore, the robot state at the end of $\mathcal{LM}_n$ (defined as the last global robot state when building $\mathcal{LM}_n$) is the same as the global robot start state of $\mathcal{LM}_{n+1}$. For simplicity, the origin of the global map W is chosen to be the same as the coordinate frame of the first local submap $\mathcal{LM}_1$, i.e., the robot's initial state.

Fig. 2 illustrates the concept of using local submaps to map the incremental knowledge about the environment. Particularly, the figure depicts a robot that has been navigating in an unknown environment, while, in the meantime, it has been encoding the perceived surrounding environment in a total of eight local submaps. Noteworthy, the example assumes open-loop navigation, i.e., without localization updates. Therefore, the first defined submaps are less uncertain with respect to the global frame W than those built at a later stage. This fact corresponds to unbounded growth of the uncertainty on the system localization estimate.

B. Local Submap as Occupancy Grid Map

The assumption of null uncertainty on the robot pose within each local submap, also referred to as known robot poses, enables the representation of each local submap as an occupancy grid map. The chosen alternative to efficiently encode an occupancy grip map is via Octomaps [30]. Octomaps permits fusing range-based data into a probabilistic voxel representation, which generates an occupancy grid map with adjustable resolution. Octomaps store the information in an octree data structure, which provides fast access time while, at the same time, optimizing memory usage. All these desirable features make the undertaken mapping strategy ideal for online mapping and planning.

The probabilistic sensor fusion within an occupancy grid map is performed as an Octomap [30], [50]. This is, the probability $P(\mathbf{v}|\mathbf{h}_{1:k})$ of a cell $\mathbf{v}$ to be occupied given a set of sensor measurements $\mathbf{h}_{1:k}$ is estimated as

$$P(\mathbf{v}|\mathbf{h}_{1:k}) = \left[1 + \frac{1 - P(\mathbf{v}|\mathbf{h}_k)}{P(\mathbf{v}|\mathbf{h}_k)} \frac{1 - P(\mathbf{v}|\mathbf{h}_{1:k-1})}{P(\mathbf{v}|\mathbf{h}_{1:k-1})} \frac{1 - P(\mathbf{v})}{P(\mathbf{v})}\right]^{-1} \tag{10}$$

where $P(\mathbf{v}|\mathbf{h}_k)$ is the inverse sensor model characterizing the sensor used for mapping and $P(\mathbf{v}|\mathbf{h}_{1:k-1})$ is the preceding estimate given all historical measurements. Using log-odds notation

$$L(\cdot) = \log\left[\frac{P(\cdot)}{1 - P(\cdot)}\right] \tag{11}$$

and under the common assumption of a uniform (noninformative) prior, i.e., $P(\mathbf{v}) = 0.5$, (10) is simplified to

$$L(\mathbf{v}|\mathbf{h}_{1:k}) = L(\mathbf{v}|\mathbf{h}_{1:k-1}) + L(\mathbf{v}|\mathbf{h}_k). \tag{12}$$

To change the state of a node $\mathbf{v}$, (12) requires as many observations as the ones used to define its current state. This overconfidence in the map is addressed as in [79] by using a clamping policy to ensure that the confidence in the map remains bounded

$$\begin{aligned} L(\mathbf{v}|\mathbf{h}_{1:k}) &= [L(\mathbf{v}|\mathbf{h}_{1:k})]_{l_{\min}}^{l_{\max}} \\ &= \max(\min(L(\mathbf{v}|\mathbf{h}_{1:k}), l_{\max}), l_{\min}) \end{aligned} \tag{13}$$

where $l_{\min}$ and $l_{\max}$ denote lower and upper bounds on log-odds values. As a consequence, the model of the environment remains updatable [30].

The measurement update rules in (12) and (13) can be used with any kind of distance sensor, as long as the inverse sensor model is available. Our framework employs the extended beam-based inverse sensor model depicted in Fig. 2(c). This model assumes that: 1) the line of sight between the sensor origin and the endpoint of measurement does not contain any obstacle (free space); 2) endpoints correspond to obstacle surfaces (occupied space); and 3) the line continuing beyond the endpoint until the maximum sensor range is likely to be occupied by the observed obstacle (occluded space). Then, the extended ray-casting operation to update each voxel $\mathbf{v}$ from the sensor origin to the maximum sensor range is performed

using the following log-odds inverse sensor model:

$$L(\mathbf{v}|\mathbf{h}_t) = \begin{cases} l_{\text{free}}, & \text{if } \mathbf{v} \text{ is traversed by the beam} \\ l_{\text{occ}}, & \text{if } \mathbf{v} \text{ is hit by the beam} \\ l_{\text{ocl}}, & \text{if } \mathbf{v} \text{ is between the hit and sensor range} \end{cases} \tag{14}$$

where l_{free} and l_{occ} are constants determined according to the sensor model, and l_{ocl} penalizes occluded zones according to the decaying function

$$l_{\text{ocl}} = \gamma^d l_{\text{occ}} \tag{15}$$

where, for a decay rate $\gamma \in [0, 1]$, l_{ocl} decreases γ times for each unit of d, which is the distance from the measurement endpoint. This corresponds to $l_{\text{ocl}} = l_{\text{occ}}$ for $d = 0$, i.e., in the hit point, and to $l_{\text{ocl}} \to 0$, i.e., to a noninformative $P(\mathbf{v}) = 0.5$, as $d \to \infty$. The maximum expand of the occluded region is as far as the sensor range.

C. Map Fusion and Single Point Query

An occupancy query to the current probabilistic map $\mathcal{M}$ is done by converting the given query into multiple local queries. The occupancy probability values at each local submap can be fused together by means of the log-odds update rule in (12) with the corresponding clamping operation in (13). These operations apply because combining measurements from multiple local submaps is a similar operation as combining multiple measurement updates in a single global map [29].

Without loss of generality, assume that an occupancy query at position $\hat{\mathbf{x}}^Y$ is performed from an uncertain coordinate frame Y with known pose estimate $\mathbf{x}_Y^W \sim \mathcal{N}(\hat{\mathbf{x}}_Y^W, \Sigma_Y^W)$. This global query corresponds to the multiple local log-odds occupancy queries

$$L(\hat{\mathbf{x}}^Y) = \sum_{i=1}^{n} \left[L_{1:i-1}(\hat{\mathbf{x}}^Y) + L_i(\mathbf{x}^{\mathcal{LM}_i})\right]_{l_{\min}}^{l_{\max}} \tag{16}$$

where $L_{1:i-1}(\hat{\mathbf{x}}^Y)$ is the accumulative log-odd estimate from the precedent $i - 1$ local submaps with $L_{1:i-1}(\hat{\mathbf{x}}^Y) = 0$ for $i = 1$, $L_i(\cdot)$ implies that the log-odds lookup is done in the local submap $\mathcal{LM}_i$, and $\mathbf{x}^{\mathcal{LM}_i} \sim \mathcal{N}(\hat{\mathbf{x}}^{\mathcal{LM}_i}, \Sigma_Y^{\mathcal{LM}_i})$ corresponds to $\hat{\mathbf{x}}^Y$ in local coordinates. $\mathbf{x}^{\mathcal{LM}_i}$ is calculated via the linear estimation of known spatial relationships

$$\mathbf{x}^{\mathcal{LM}_i} = \ominus\mathbf{x}_{\mathcal{LM}_i}^W \oplus \left(\mathbf{x}_Y^W \oplus \mathbf{x}^Y\right) \tag{17}$$

where $\oplus$ denotes the compounding operation and $\ominus$ corresponds to its inverse relation, as commonly used to simplify notation when calculating spatial transformations (see Appendix B for a brief introduction and [70] for a full review).

Given that $\hat{\mathbf{x}}^Y$ in local coordinates follows a probabilistic distribution, the local occupancy query $L_i(\mathbf{x}^{\mathcal{LM}_i})$ is

$$P_i(\mathbf{x}^{\mathcal{LM}_i}) = \sum_{\mathbf{v} \in \mathcal{LM}_i} P(\mathbf{v}) \, \mathcal{N}\left(\mathbf{v} \mid \hat{\mathbf{x}}^{\mathcal{LM}_i}, \Sigma_Y^{\mathcal{LM}_i}\right) \tag{18}$$

where $\mathbf{v}$ represents the set of voxels in submap $\mathcal{LM}_i$ and $P_i(\mathbf{x}^{\mathcal{LM}_i})$ can be described in log-odds $L_i(\mathbf{x}^{\mathcal{LM}_i})$ notation via the log-odds transform.

D. Computation of the Cumulative Map $F_{\mathcal{X}}^{R'}$

Section V-C provides a strategy to query the occupancy probability $P(\mathbf{x})$ of a single point coordinate $\mathbf{x} \in \mathcal{X}$. Our previous work [55] demonstrated that this approach is suitable for the requirements of an online planner under probabilistic constraints. However, bearing in mind that each planning cycle requires numerous queries of $P(\mathbf{x})$ involving different $\mathbf{x}$, the overall planner performance can be enhanced by computing the map fusion before the planning time budget starts.

The probabilistic map fusion over all state space $\mathcal{X}$ is described by the cumulative distribution $F_{\mathcal{X}}$ over the local density distributions of the sensed environment.[3] In particular, for the online planning problem, it is of interest to fuse the map information with respect to the predicted planning frame R' such that the cumulative map $F_{\mathcal{X}}^{R'}$ reflects the relative uncertainty between the current environmental awareness and the planning frame R'. Fig. 2(e) illustrates the extraction of $F_{\mathcal{X}}^{R'}$ from a set of local maps. Computing $F_{\mathcal{X}}^{R'}$ implies that the computational requirements of retrieving $P(\mathbf{x}^{R'})$ during planning time are reduced to those of a lookup table in the cumulative map $F_{\mathcal{X}}^{R'}$.

Subject to the log-odds transformation, $F_{\mathcal{X}}^{R'}$ is computed by rewriting (16) and (18) as

$$L(\hat{\mathcal{X}}^{R'}) = \sum_{i=1}^{n} \Big[L_{1:i-1}(\hat{\mathcal{X}}^{R'}) + L_i(\mathcal{X}^{\mathcal{LM}_i}) \Big]_{l_{\min}}^{l_{\max}} \quad (19)$$

where $L_{1:i-1}(\hat{\mathcal{X}}^{R'})$ is the accumulative log-odd estimate from the precedent $i-1$ local submaps with $L_{1:i-1}(\hat{\mathcal{X}}^{R'}) = 0$ for $i = 1$, $L_i(\cdot)$ is the log-odds lookup done in the local occupancy submap $\mathcal{LM}_i$, and $\mathcal{X}^{\mathcal{LM}_i} \sim \mathcal{N}(\hat{\mathcal{X}}^{\mathcal{LM}_i}, \Sigma_{R'}^{\mathcal{LM}_i})$ corresponds to the state space $\hat{\mathcal{X}}^{R'}$ in local coordinates defined as

$$\mathcal{X}^{\mathcal{LM}_i} = \ominus \mathcal{X}_{\mathcal{LM}_i}^{W} \oplus \left(\mathcal{X}_{R'}^{W} \oplus \mathcal{X}^{R'} \right). \quad (20)$$

Then, the occupancy probability $L_i(\mathcal{X}^{\mathcal{LM}_i})$ at $\mathcal{LM}_i$ for all $\mathbf{x} \in X^{\mathcal{LM}_i}$ is computed as

$$P_i(\mathcal{X}^{\mathcal{LM}_i}) = \sum_{\mathbf{v} \in \mathcal{LM}_i} P(\mathbf{v}) \, \mathcal{N}\left(\mathbf{v} \mid \hat{\mathbf{x}}, \Sigma_{R'}^{\mathcal{LM}_i} \right) \quad \forall \mathbf{x} \in \mathcal{X}^{\mathcal{LM}_i}$$

$$= \mathcal{LM}_i \otimes \mathcal{K}_\alpha\left(\Sigma_{R'}^{\mathcal{LM}_i} \right) \quad (21)$$

where $\mathbf{v}$ represents the set of voxels in submap $\mathcal{LM}_i$ and $\mathcal{K}_\alpha(\cdot)$ with confidence level $\alpha = 1$ is a kernel representing the discrete version of a Gaussian distribution over the entire span of the local submap $\mathcal{LM}_i$ (see Appendix C). $\otimes$ is the correlation operator, i.e., a sliding inner product, and $P_i(\mathcal{X}^{\mathcal{LM}_i})$ can be described in log-odds $L_i(\mathcal{X}^{\mathcal{LM}_i})$ via the log-odds transform.

Interestingly, the underlying computation of $F_{\mathcal{X}}^{R'}$ is the correlation operator $\otimes$, a common technique for which there exist efficient implementations. On top of that, the independence between local submaps allows parallelizing (21) for each $\mathcal{LM}_i$ in different threads. Ideally, this process could be scheduled

[3]Only those voxels describing the known environment, i.e., free, occluded, and occupied space, are considered in the computation of $F_{\mathcal{X}}$. Considering the unknown space with its $P(\mathbf{v}) = 0.5$ in the computations would lead to a cumulative map with misleadingly overestimating occupancy probabilities.

such that $F_{\mathcal{X}}^{R'}$ is ready before the planning time budget starts.

VI. Multilayered Motion Planning Under Environment and Motion Uncertainty

The planning problem defined in Section III-D has three main requirements: 1) to consider the vehicle's motion constraints; 2) to validate probabilistic constraints in face of uncertainties; and 3) to meet online computation limitations. Our previous approach successfully addressed all these requirements formulating a single-layered sampling-based planning strategy in the belief space [55]. The planner in question: 1) samples feasible states in the system's state space and 2) extends and validates the tree of motions in the belief space. This approach proved to be suitable for solving online motion planning problems in challenging real-world scenarios, but its applicability was limited to low-dimensional planning problems given the huge search space and the computational burden of all considered constraints.

As discussed in Section II, multilayered planning strategies enable online planning in high-dimensional spaces. This motivates the use of such an idea to extend our framework's capabilities to suit the planning requirements of a larger group of robotic systems and environments. Principally, the extended planning strategy employs a multilayered planning scheme (see Section VI-A) to overcome the aforementioned scalability issues. Such a strategy allows deferring the computation of kinematic constraints (see Section VI-B) and probabilistic constraints (see Section VI-C) after identifying some subregions of the system's state space that potentially contain a solution to the planning problem.

A. Multilayered Motion Planning

The capabilities of our previous planner (hereinafter referred to as the constrained planner) are extended to deal with problems of higher dimensionality by means of a multilayered planning strategy. As schematized in Fig. 3, the proposed strategy adopts a sequential two-layered planning scheme consisting of a lead planner and the constrained planner. The lead planner seeks to determine a subregion $\mathcal{X}' \subset \mathcal{X}$ of the entire state space that eases and, consequently, speeds up the search of the final trajectory ξ, which accounts for all considered constraints (see Section III-D). To this aim, the multilayered scheme is designed as follows.

1) *Lead Planner:* It employs the RRT* to rapidly find a path in the workspace $\mathcal{W}$. The computed lead path is a nearly optimal geometric solution $\xi' \in \mathcal{W}$ used to determine $\mathcal{X}'$ via the lifting operator lift : $\mathcal{W} \to \mathcal{X}$ detailed in the following.

2) *Constrained Planner:* It leverages the delimited search space $\mathcal{X}'$ and the SST in [55] to rapidly compute the final solution ξ, which meets kinodynamic (see Section VI-B) and probabilistic safety constraints (see Section VI-C).

Although the planners within the multilayered planning scheme could be different, the selection above suits the online requirements of our framework. This is, the framework's overall planning time ΔT_{MP} is divided as $\Delta T_{\mathrm{MP}} = \Delta T_L + \Delta T_C$,
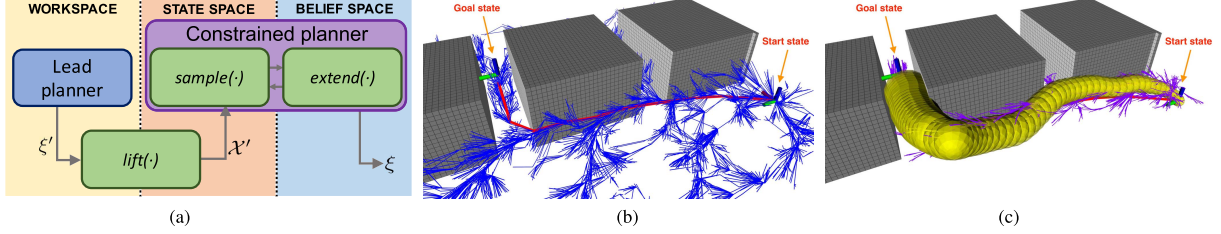
Fig. 3.   (a) Multilayered motion planning framework: (b) lead planner—RRT* shown in blue computes a geometric lead ξ' (red path) to guide the search space $\mathcal{X}'$ of (c) constrained planner—SST shown in magenta. The resulting trajectory ξ with its uncertainty (yellow funnel) satisfies kinodynamic and probabilistic safety constraints.

where ΔT_L and ΔT_C are the time budgets allocated to the lead and constrained planners, respectively. Then, given our selection of planners, the assignment of time budgets allows $\Delta T_L \ll \Delta T_C$ as the lead planner is adept at providing quickly a suitable lead path such that the constrained planner has at its disposal most of the time budget $\Delta T_L \approx \Delta T_{\mathrm{MP}}$ to refine the final trajectory, which accounts for all the considered constraints.

Given our selection of planners and their operational space, the designed multilayered planning scheme requires the lifting lift : $\mathcal{W} \to \mathcal{X}$. A common lift$(\cdot)$ strategy is to define $\mathcal{X}'$ as a tube around ξ' with radius d for the geometric components of the state space, whereas the nongeometric components are left unbounded, e.g., [56], [57], [76]. The performance of this approach, however, is susceptible to the parametrization of d; tight search spaces, i.e., small radius d, promote final solutions with lower cost than those obtained with bigger radius d. On top of that, relying on a fixed d requires hand-tuning such parameter to ensure that the final solution lies within $\mathcal{X}'$; if $\mathcal{X}'$ does not contain the final solution, the planner will lack probabilistic completeness. Adjusting d to ensure probabilistic completeness would prove to be a cumbersome task since the type of environment and planning constraints, among many other factors, should be taken into account.

Different from other multilayered planning schemes in the literature, ours uses a method of information interchange between planners that maintains the completeness and asymptotic optimality properties of the constrained planner when used in a standalone fashion [55]. This work builds on the idea of sampling around a lead path to present alternative definitions of $\mathcal{X}'$ via the lift$(\cdot)$ operator. In particular, the designed MLP exploits a mixture of samplers to trade off the low-cost trajectories found when sampling around a lead path and the probabilistic completeness of uniform sampling. This article proposes two mixtures of sampling techniques.

1) *Bias to Rigid $\mathcal{X}'$:* Given a fixed radius d, the planner samples uniformly in $\mathcal{X}'$ with probability p and uniformly over the space with probability $1 - p$.
2) *Adaptive $\mathcal{X}'$:* The planner adjusts d within the range of a strictly guided sampling to a uniform search. Adjusting d can be conducted via some heuristics or as an optimization problem subject to a cost function.

The performance of both approaches in comparison to a rigid $\mathcal{X}'$ strategy is discussed in Section VII-B. Noteworthy,
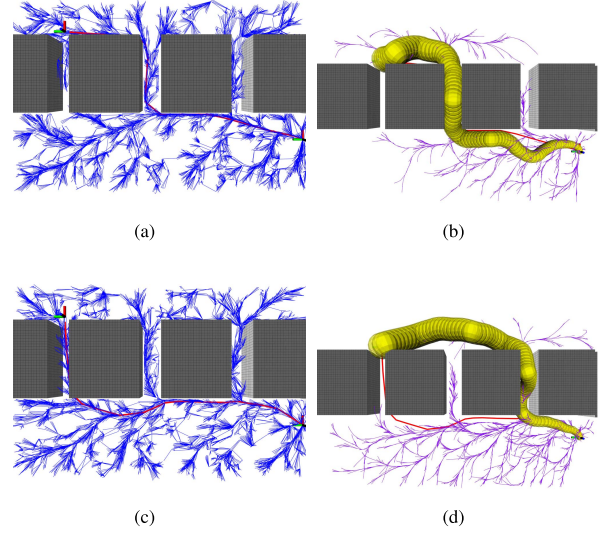
Fig. 4.   Probabilistic completeness of the proposed multilayered planning scheme with adaptive lead, which (a) and (b) promotes finding the final solution in the neighborhood of the asymptotically optimal lead ξ' (red), while it (c) and (d) preserves completeness guarantees even when the lead ξ' transverses a corridor, which does not offer a probabilistic safe passage. (a) Lead planner—RRT*. (b) Constrained planner–SST. (c) Lead planner—RRT*. (d) Constrained planner–SST.

any of the two presented mixtures of sampling strategies ensures probabilistic completeness of the overall multilayered scheme. As an extreme example, let us consider the scenario depicted in Fig. 4(c) and (d), where the lead planner finds an asymptotically optimal solution through the farthest (most left) corridor. However, according to the probabilistic safety constraints defined in Section VI-C, such a corridor does not offer any safe passage. Despite the initial bias toward this unsuitable $\mathcal{X}'$, a mixture of sampling strategies, as the ones introduced in this section, permits finding a solution if one exists provided enough time, thus ensuring probabilistic completeness guarantees.

B. Planning Under Motion Constraints

The system's motion capabilities are considered in the constrained planner by expanding a tree with the system's motion model (2) and (3). In particular, the constrained planner employs the SST algorithm [41] to build a tree in the

Fig. 5. Tree expansion under motion and probabilistic constraints. The state beliefs (nodes) of the tree are obtained by considering the motion capabilities. The ellipses surrounding the states represent their uncertainty, where green corresponds to those states satisfying the probabilistic safety constraints and red corresponds to those that do not.
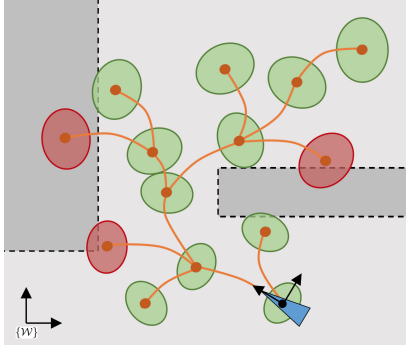
belief space with state beliefs $x \sim b = \mathcal{N}(\hat{x}, \Sigma_x)$ as nodes. The tree expansion is based on two procedures: *sample*($\cdot$) and *extend*($\cdot$), which are conducted in the state and belief space, respectively [see Fig. 3(a)]. That is, *sample*($\cdot$) draws a random state $x_{\text{rand}} \in \mathcal{X}'$, where $\mathcal{X}'$ is a subregion of $\mathcal{X}$ as defined in Section VI-A. The planner then selects a node from the tree to attempt connecting to the randomly sampled state x_{rand}. Such a selection is conducted via nearest neighbor in the state space using the Euclidean metric. The selected node x_{near} has a probabilistic representation in the belief space, i.e., x_{near} is better described as $x_{\text{near}} \sim b_{\text{near}} = \mathcal{N}(\hat{x}_{\text{near}}, \Sigma_{x_{\text{near}}})$. Then, from this belief, the *extend*($\cdot$) procedure expands the tree in the belief space by evolving the system's motion model (2) and (3) with a randomly sampled control input $\mathbf{u} \in \mathcal{U}$. This expansion is done for a random period of time T_{prop}. Since the considered motion model includes the system's uncertainty, each obtained belief (tree node) corresponds to a vehicle's state with its associated uncertainty (see Fig. 5).

C. Planning Under Probabilistic Constraints

In our approach, the environmental awareness relative to the current robot state is represented as a cumulative distribution on discrete support $F_{\mathcal{X}}^{R'}$ (see Section V). As discussed previously, this representation of the environment favors efficiency for online mapping and planning applications. In fact, we leverage such encoding to guarantee $1 - p_{\text{collision}}(b, \mathcal{M}) \geq p_{\text{safe}}$ for each belief b of the tree as

$$1 - \left(p_{\text{collision},\alpha}\left(b, F_{\mathcal{X}}^{R'}\right) + (1-\alpha) \right) \geq p_{\text{safe}} \quad (22)$$

$$\alpha - p_{\text{collision},\alpha}\left(b, F_{\mathcal{X}}^{R'}\right) \geq p_{\text{safe}} \quad (23)$$

where α is the confidence level on the computation of $p_{\text{collision},\alpha}(\cdot) \in [0, \alpha]$. In other words, $p_{\text{collision},\alpha}(\cdot)$ does not cover a $(1 - \alpha)$ span of the belief b over the state space. Therefore, it is assumed that the remaining $(1 - \alpha)$ is in collision to ensure probabilistic guarantees on the collision checking decision. All in all, this method can be exploited to trade a constant conservatism α in favor of performance.

Then, the probability of collision of a robot centered belief $b^{R'} \sim \mathcal{N}(\hat{b}^{R'}, \Sigma_b^{R'})$ with the environment is calculated as

$$p_{\text{collision},\alpha}\left(b^{R'}, F_{\mathcal{X}}^{R'}\right) = \left\langle \mathcal{K}_\alpha\left(\Sigma_b^{R'}\right), F_{\mathcal{X}}^{R'} \right\rangle_F$$
$$= \text{vec}\left(\mathcal{K}_\alpha\left(\Sigma_b^{R'}\right)\right)^T \text{vec}\left(F_{\mathcal{X}}^{R'}\right) \quad (24)$$

where $\langle \cdot, \cdot \rangle_F$ is the Frobenius inner product of the overlapping region between the $\hat{b}^{R'}$-centered discrete state belief $\mathcal{K}_\alpha(\Sigma_{b_k}^{R'})$ (see Appendix C) and the cumulative environment awareness $F_{\mathcal{X}}^{R'}$. The Frobenius inner product is an efficient operation via matrix vectorisation.

The overall proposed MLP leads to the exploration tree depicted in Fig. 5, whose edges account for the vehicle's kinodynamic capabilities and whose nodes are probabilistically safe subject to the system's localization, motion, and environment uncertainties. In addition, the expansion of the tree is also subject to states not leading to an inevitable collision, i.e., a state must allow for the vehicle to make a full stop before colliding.

VII. EXPERIMENTAL EVALUATION

The proposed framework has been implemented in ROS and uses the facilities provided by Octomap [30] and the OMPL [72] as the core building block of the proposed mapping and planning strategies, respectively. This implementation is used to evaluate thoroughly the different proposed features and the framework as a whole. This section reports the results of such analysis in an incremental fashion. First, Section VII-A presents a discussion on the capabilities of our framework's precedent version in simulated and real-world scenarios. Then, Section VII-B and Section VII-C report the performance of the key components of the newly proposed framework, i.e., the multilayered scheme and the probabilistic collision checking. The potential of these components is individually evaluated against closely related state-of-the-art approaches. Finally, in Section VII-D, the capabilities of the new framework are demonstrated in a challenging scenario.

A. Navigation in Unknown 2-D Environments

An early version of the work presented in this article has appeared before [55]. This consisted of a simpler framework that evaluated all uncertainties on the fly (in contrast to the proposed cumulative map encoding) while exploring the belief space via a single-layered planner (in contrast to the proposed multilayered guided exploration). As reported next, our precedent work proved to be suitable for safe robot navigation, but its computational requirements limited its applicability to low-dimensional spaces.

The precedent framework has been deployed on the Sparus II autonomous underwater vehicle (AUV) (see Fig. 6), a nonholonomic torpedo-shaped vehicle with hovering capabilities [10]. To meet the limitations of our precedent work, the AUV is limited to operate at a constant depth, i.e., in SE(2). Under these conditions, the motion model of the Sparus II can be approximated by a unicycle system, as detailed in Appendix A-A. The AUV is equipped with a mechanical scanned imaging sonar (MSIS) to perceive the

Fig. 6. Sparus II AUV, a nonholonomic vehicle.

surroundings and incrementally map the environment. We use the default parameters in [30] of $l_{\text{free}} = -0.4$ ($P(\mathbf{v}) = 0.4$), $l_{\text{occ}} = 0.85$ ($P(\mathbf{v}) = 0.7$), $l_{\text{min}} = -2$ ($P(\mathbf{v}) = 0.12$), and $l_{\text{max}} = 3.5$ ($P(\mathbf{v}) = 0.97$). The decay rate in (15) is set to $\gamma = 0.8$. The framework's planning time is set to $\Delta T_{\text{MP}} = 1.5$ s.

The test bed to evaluate the precedent framework consists of two environments located in Sant Feliu de Guíxols (Spain): 1) breakwater structure that is composed of a series of concrete blocks (14.5-m long $\times$ 12-m width), which are separated by 4-m gaps [see Fig. 7(a) and (b)] and 2) rocky formations that create an underwater canyon of 28-m long [see Fig. 8(a) and (b)]. Using these environments, two experiments are reported: 1) evaluation of the overall performance of the framework in the underwater simulator (UWSim) [65] and 2) validation of the framework in real in-water trials. Experiment 1) is conducted in both environments, while experiment 2) is uniquely tested in the real breakwater structure scenario.

1) Simulated Trials: The framework is exhaustively tested in the simulated breakwater structure and canyon scenarios with 20 attempts per scenario (total of 40 trials).

In the breakwater structure, 19 start-to-goal queries are successfully solved. Among those successful experiments, the robot achieved the goal region $\mathcal{B}_{\text{goal}}$ by crossing through the first 1-m gap in 17 occasions, while, in the remaining two trials, the planner found a less optimal trajectory through the second 4-m passage. Fig. 7 depicts the mission execution in one of those trials. In the initial part of the mission, as the environment is completely undiscovered, the computed trajectory goes straight to the goal [see Fig. 7(c)]. As soon as the trajectory gets invalidated, a new collision-free trajectory is computed [see Fig. 7(d)]. After some mapping–planning iterations, the robot gets out of the 4-m gap between two blocks [see Fig. 7(e)]. On average, the computed trajectories toward the goal have a length of approximately 45.2 m and are completed in 2′21″.

All 20 start-to-goal queries in the simulated canyon scenario are successfully solved. The higher success rate with respect to the previous experiment is given by the nature of the environment; this scenario involves less abrupt maneuvers, and the passage is wider, more than twice larger though. Fig. 8(d) depicts one of the trajectories calculated through the narrow passage of the canyon. On average, the calculated trajectories toward the goal have a length of approximately 58.4 m and are completed in 2′59″.

2) Real-World Trials: In-water experiments are conducted in the real breakwater structure located in Sant Feliu de Guíxols (Spain). Similar to the simulated trials, the robot is required to solve a start-to-goal-query to reach a goal region $\mathcal{B}_{\text{goal}}$ located on the opposite side of the structure, which can only be achieved by navigating through any of the narrow 4-m gaps. A total of five start-to-goal queries are attempted. During those autonomous missions, the vehicle is connected to a wireless access point buoy for monitoring purposes; all components of the framework run on the robot to prove the framework's suitability for real-world robots with limited onboard computation power.

In all five trials, the framework was successful in finding and driving the Sparus II AUV toward the desired goal region $\mathcal{B}_{\text{goal}}$ through one of the narrow gaps in the breakwater structure.[4] In four trials, the trajectory was found through the first corridor, while, in the other trial, the robot went through the second gap. Fig. 9 depicts Sparus II in one of those in-water trials and the trajectory calculated toward the goal, which has a length of 57.9 m and took 3′07″.

B. Multilayered Planning Scheme

The multilayered planning scheme presented in Section VI-A is one of the key features allowing us to overcome the scalability issues of our previous single-layered planner [55]. Nonetheless, different from current multilayered approaches that rely on rigid definitions of the search space $\mathcal{X}'$ (rigid-$\mathcal{X}'$), this article explores two alternative definitions of $\mathcal{X}'$ (biased-$\mathcal{X}'$ and adaptive-$\mathcal{X}'$) based on a mixture of sampling experts. This section reports the performance of these four strategies in the following belief space planning problem: reaching the state between the blocks in Fig. 10 while satisfying kinodynamic and probabilistic safety constraints subject to a $p_{\text{safe}} = 0.99$ minimum safety probability bound. In this evaluation, the entire 3-D environment is considered to be known in advance, and the system dynamics are approximated as described in Appendix A-B.

The four methods [single-layered planner (SLP) and MLP with rigid-$\mathcal{X}'$, biased-$\mathcal{X}'$, and adaptive-$\mathcal{X}'$] are evaluated for their ability to quickly find a solution and for the cost of the resulting trajectory. The given total planning time budget is set at $\Delta T_{\text{MP}} = 1.5$ s to emulate online planning requirements, which is distributed as $\Delta T_L = 0.3$ s and $\Delta T_C = 1.2$ s for the three multilayered schemes. With this setup, each planner attempts to solve the defined planning problem a total of 2000 times.

Fig. 11 depicts the number of successfully solved trials and the resulting trajectory length when considering a rigid-$\mathcal{X}'$ lead with radius parameterisations $d \in [0, 40]$ m. While $d = 0$ m strictly limits the search space to those states on the lead path, $d = 40$ m spans the search over all state space of the defined planning problem, therefore resembling uniform sampling. As it can be observed, small search spaces (small d) endanger the planner's ability to find a solution with limited time. However, when a trajectory is found, the resulting cost is lower than those solutions found with wider $\mathcal{X}'$ leads. Instead, these wide search spaces (big d) make the planner struggle at solving

[4]A complete sea-trial through the real breakwater structure can be seen in https://youtu.be/dTejsNqNC00
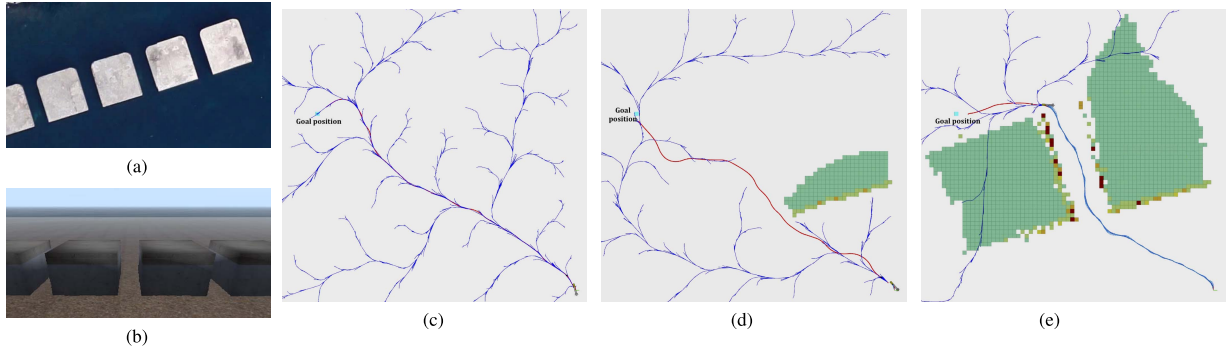
Fig. 7. Incrementally mapping and planning in the undiscovered breakwater structure scenario. (c) Initial state of the Sparus II AUV in the unknown environment with the initially found trajectory (red). (d) Anytime the trajectory is invalidated, a new collision-free trajectory is computed. (e) After some iterations, the robot gets out of the 4-m gap between blocks. (a) Real breakwater. (b) Simulated breakwater. (c) Initial empty map. (d) Replanned trajectory. (e) Final part of the survey.
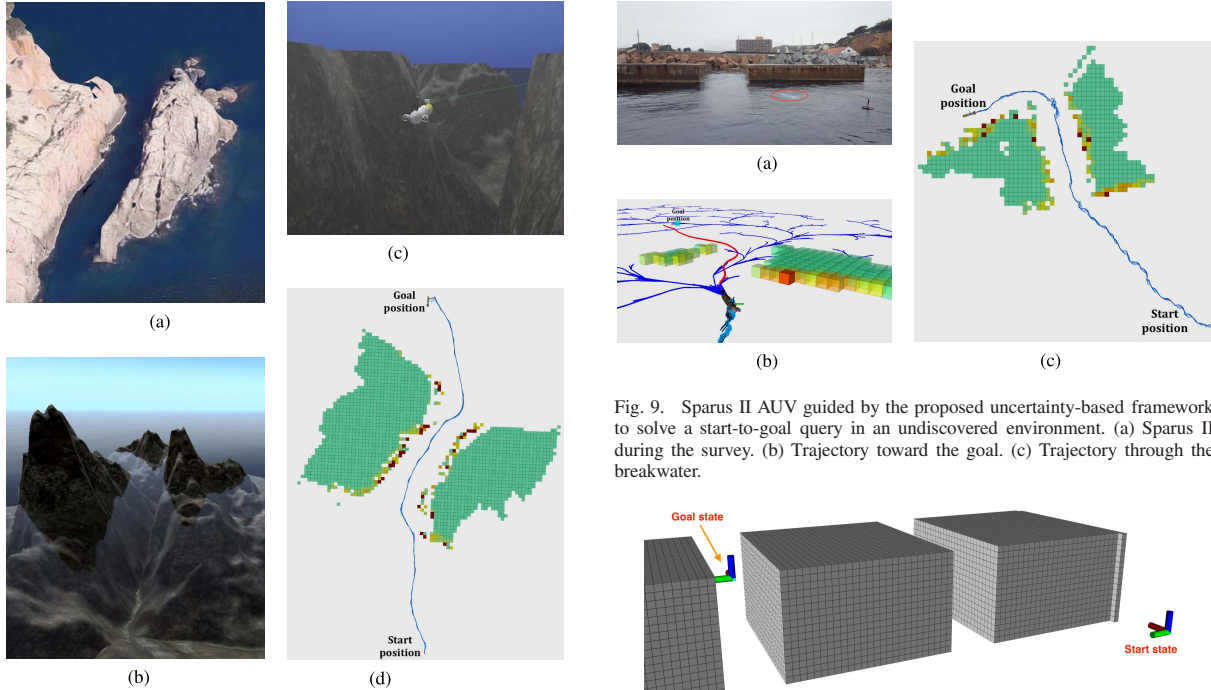

Fig. 8. Incrementally mapping and planning in the undiscovered canyon scenario. (a) Real canyon. (b) Simulated canyon. (c) Sparus II in the UWSim. (d) Trajectory through the canyon.

Fig. 9. Sparus II AUV guided by the proposed uncertainty-based framework to solve a start-to-goal query in an undiscovered environment. (a) Sparus II during the survey. (b) Trajectory toward the goal. (c) Trajectory through the breakwater.
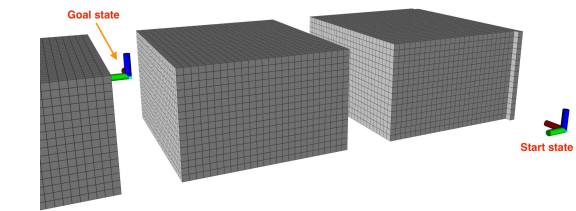
Fig. 10. Planning problem to assess the proposed multilayered scheme with adaptive $\mathcal{X}'$ in comparison to other state-of-the-art approaches. The problem is defined in the belief space for a SE(3) system operating in a 3-D workspace. The minimum safety bound is set to $p_{\text{safe}} = 0.99$.

most of the planning problems due to the search space extent. In between these two extremes, a suitable parameterization with $d = 12$ m (dashed lines) enables solving most of the trials to the planning problem while providing a trajectory with low length cost. Nevertheless, there are no efficient means of defining the optimal d in advance since it is dependant on the planning problem and environment characteristics. Therefore, a rigid-$\mathcal{X}'$ strategy is not suitable for applications that lack a fully prior informative representation of the environment. Moreover, too restrictive guided searches can endanger the completeness guarantees of the planner.

The performance of those approaches that guarantee completeness, i.e., the SLP (green) and MLP with biased-$\mathcal{X}'$ (magenta) and adaptive-$\mathcal{X}'$ (orange) strategies is depicted in Fig. 12. In particular, biased-$\mathcal{X}'$ is parametrized with radius $d = 12$ m (best lead definition according to experimentation in Fig. 11) and analyzed for different $p \in [0, 1]$, whereas adaptive-$\mathcal{X}'$ is defined, as shown in Fig. 13, i.e., with an initial radius $d = 3$ m, which increases at a rate of 20 m/s. Intuitively, adaptive-$\mathcal{X}'$ adjusts d from a strictly guided search to uniform sampling, i.e., as $t \rightarrow \infty$, $d \rightarrow \infty$, i.e., $\mathcal{X}' \rightarrow \mathcal{X}$.
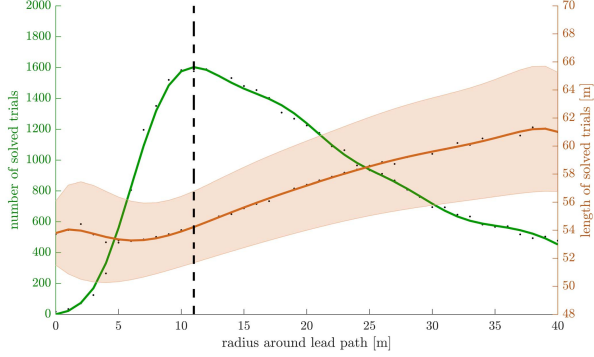
Fig. 11. Performance of the multilayer planning scheme with a rigid-$\mathcal{X}'$, i.e., fixed radius around the geometric lead path.
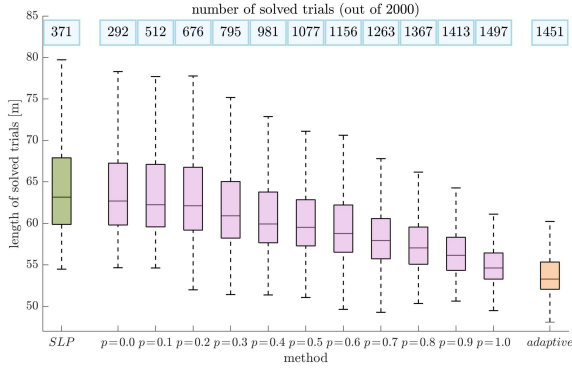
Fig. 12. Performance of 1) our precedent SLP scheme (green) and the newly proposed multilayered scheme when considering 2) a fixed lead with different bias p (magenta, with best radius as found in Fig. 11), or 3) an adaptive lead, as defined in Fig. 13 (orange).

Fig. 13. Two-layered planning scheme proposed in this work. After computing a lead path, the constrained planner leverages an adaptive $\mathcal{X}'$ strategy to initially promote solutions with low cost (small d) before ensuring probabilistic completeness by sampling the entire space ($d \to \infty$). Once a solution is found, $\mathcal{X}'$ is fixed to let the constrained planner refine the found solution until the completion of the planning time ΔT_{MP}.

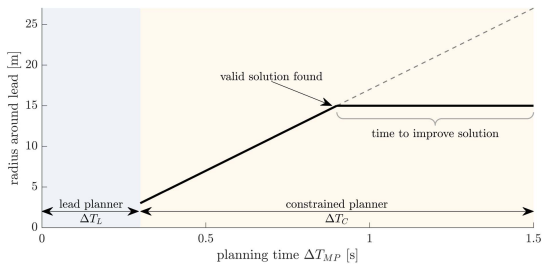As it can be observed in Fig. 12, our precedent single-layered planning scheme struggles at finding a solution on most of the trials; sampling uniformly the entire high-dimensional belief space requires more time to find a

solution than the affordable time budget in online applications. Slightly worse performance is obtained when using a multilayered scheme with biased-$\mathcal{X}'$ and $p = 0$ because it still uses uniform sampling but with a portion of the total planning time budget. However, as $p \to 1$, i.e., the planner is more guided to the lead $\mathcal{X}'$ (whose optimal radius has been determined empirically in Fig. 11), the performance of the planner increases, in both the number of solved trials and the length of the final solution. Interestingly, the proposed adaptive sampling method endows the framework with a competitive success rate and solution length to that obtained when hand-defining the optimal radius.

C. Probabilistic Collision Checking

Sampling-based planners must be able to analyze the validity of a certain state accurately and efficiently. While accuracy is relevant to avoid discarding regions of the state space, which, in fact, are collision-free, efficiency allows for more space exploration given a limited time budget. However, accurate calculations jeopardize the ability to validate a state rapidly, especially when accounting for uncertainty. In this regard, chance constraints formulations [9], [46] offer an interesting accuracy–efficiency tradeoff that has proven to be suitable for many motion planning problems in the last decade (see Section II). In fact, chance constraints formulations are still the most widely used probabilistic collision checking method among those state-of-the-art motion planning applications that account for uncertainty (e.g. [12], [71]). This motivates the use of chance constraints as the baseline reference to assess the proposed probabilistic collision checking algorithm.

The performance analysis comprises two chance constraints formulations [9], [46] and our method with four different parametrizations $\alpha = \{0.90, 0.95, 0.99, 0.999\}$. Each method is assessed by its accuracy and efficiency. Accuracy is computed as the ability to correctly detect that a state is valid

$$\frac{TP}{TP + FN} \in [0, 1] \tag{25}$$

where a true positive (TP) indicates that a method's outcome matches the standard of truth,[5] while a false negative (FN) reflects that the method has mistakenly computed a state as invalid. TP + FN is the total number of valid states according to the standard of truth. Therefore, the higher the value of the metric in (25), the more accurate the method is. For the method's efficiency, the analysis considers the average computation time to process the state validity.[6] These two metrics are analyzed subject to three variables relevant to motion planning problems under uncertainty: 1) number of obstacles n_o in the environment; 2) state uncertainty $\Sigma_{\mathbf{x}}$; and 3) minimum safety probability bound p_{safe}. With this setup, a problem instance is parametrized by the triplet $\langle n_o, \sigma_x, p_{\mathrm{safe}} \rangle$. In total, 847 instances are retrieved according to the parametrization span and discretization defined in Table II.

Each problem instance is set up as follows. An environment $\mathcal{M}$ is defined in $\mathbb{R}^3$ with a total of n_o cubical obstacles.

---

[5]The standard of truth is approximated by numerical integration of (6).

[6]All experiments are performed with an Intel Core i7-7820X CPU at 3.60 GHz × 16 with optimized C++ implementation for all methods.
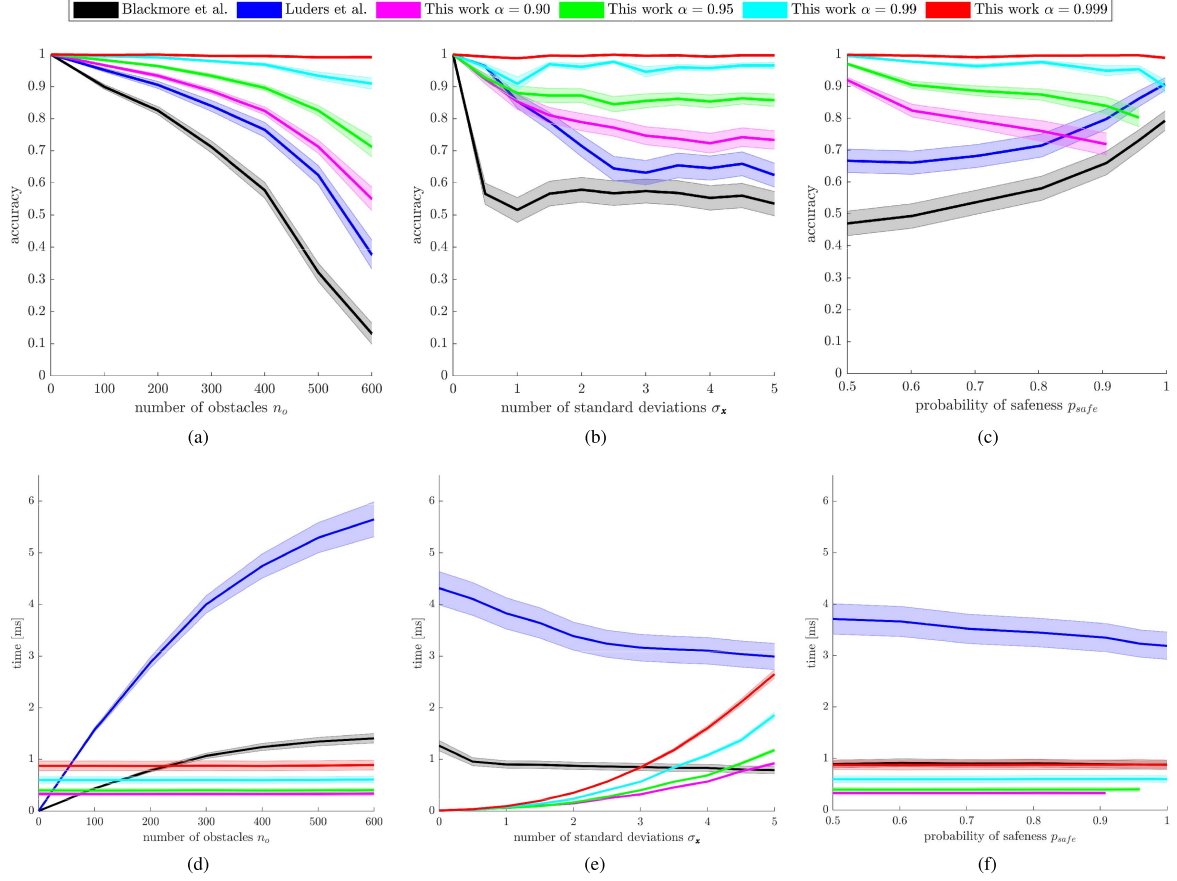
Fig. 14. Evaluation of the accuracy (first row) and performance (second row) of the chance constraints formulations [9], [46] and the proposed probabilistic collision checking method with $\alpha = \{0.90, 0.95, 0.99, 0.999\}$. The accuracy and performance metrics are represented subject to the number of obstacles n_o in the environment (first column), the state uncertainty $\Sigma_\mathbf{x} = \sigma_x^2 \mathbb{I}_{3\times3}$ (second column), and minimum safety probability bound p_{safe} (third column). The shadowed area corresponds to the variance of the metrics. In the interest of clarity, only one tenth of the variance is displayed.

TABLE II
PARAMETRIZATION FOR THE COMPARISON OF PROBABILISTIC
COLLISION CHECKING METHODS

	Minimum value	Maximum value	Discretisation step
n_o	0	600	100
p_{safe}	0.5	1.0	0.05
σ_x	0	5.0	0.5

In order to have a computational representation of the scene suitable for each method, the environment is encoded as: 1) a set of linear constraints, where each cubical obstacle is characterized by six constraints and 2) a global occupancy grid map with 0.5-m resolution. Then, given the known environment $\mathcal{M}$, each probabilistic collision checking method is required to validate, subject to p_{safe}, 10 000 beliefs $b \sim \mathcal{N}(\hat{\mathbf{x}}, \Sigma_\mathbf{x})$. The state estimate $\hat{\mathbf{x}} \in \mathbb{R}^3$ is uniformly sampled over $\mathcal{X}$, and the covariance $\Sigma_\mathbf{x} \in \mathbb{R}^{3\times3}$ is set diagonal, i.e., $\Sigma_\mathbf{x} = \sigma_x^2 \mathbb{I}_{3\times3}$.

The data from the 847 problem instances are depicted in Fig. 14. In the interest of clarity, the corresponding discussion is divided into three parts: accuracy, efficiency, and suitability.

1) Accuracy Discussion: The accuracy analysis (first row in Fig. 14) depicts that the number of obstacles in the environment is the variable penalizing the methods' accuracy the most. This behavior is due to the methods' conservatism, whose relevance increases with the hardness of the motion planning problem. In other words, the more conservative a method is, the more negatively affected it is. On top of that, the conservatism of chance constraints formulations [9], [46] increases with the number of obstacles, whereas our approach accounts for a constant conservatism α. This tighter bound allows our method to outperform both chance constraints formulations, even when choosing the most conservative parametrization $\alpha = 0.9$. Higher values of α favor accuracy at the cost of more computational expenses (see discussion below). Importantly, the confidence level α of our method should
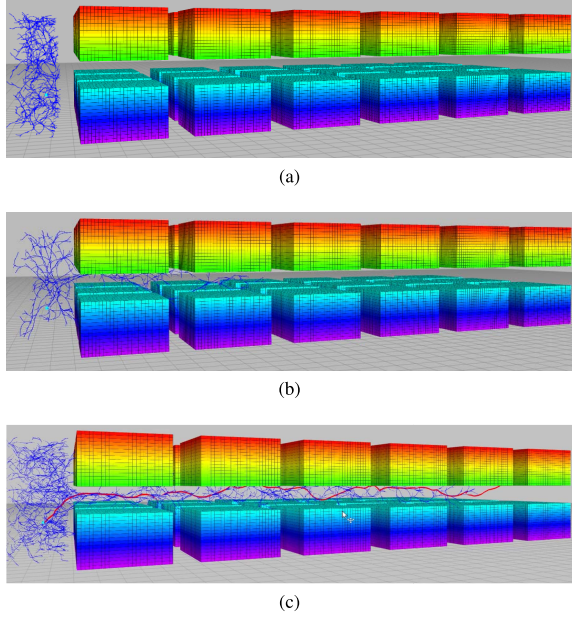
(a)

(b)

(c)

Fig. 15. Effect of conservatism in the workspace. The over conservatism of chance constraints formulations [9], [46] impedes finding a solution. Our probabilistic collision checking with a fixed conservatism α succeeds on finding a trajectory that transverses the environment with a total of 36 obstacles. (a) [9]. (b) [46]. (c) This work.

(a)

(b)   (c)

Fig. 16. Urban Stairwell scenario of the DARPA Subterranean Challenge 2019. (a) Start-to-goal query that requires traversing (perspective view). (b) 40-m-long tunnel and (c) narrow 25-m-long stairwell (entrance to narrow stairwell). Planning through the stairwell is particularly challenging due to the accumulated localization uncertainty.

always be set such that $\alpha \geq p_{\text{safe}}$; otherwise, the constraint in (23) will never be satisfied since the analyzed part of the space is not sufficient to ensure probabilistic safety. This fact is visible in Fig. 14(c), where, for $\alpha < p_{\text{safe}}$, our method with parametrization α is not used.

2) Efficiency Discussion: The efficiency analysis (second row in Fig. 14) reflects the expected computational complexity according to the theoretical grounds of each algorithm. That is, chance constraints strategies are fast for scenarios with few numbers of constraints, but their computational expenses grow linearly as the number of constraints increases. This linear correlation is influenced by the iterative nature of chance constraints, which allows invalidating a state as soon as $1 - p_{\text{collision}}(b, \mathcal{M}) < p_{\text{safe}}$, i.e., without the need to check all constraints. In other words, invalid states involve less time than those which are valid. Consequently, harder planning problems, i.e., those involving more obstacles, higher uncertainties, or more restrictive safety guarantees, show a mild deviation toward lower computational time due to the presence of a high number of invalid states. In contrast, the computational requirements of our method are uniquely influenced by the state uncertainty Σ_z, which determines the number of voxels to include in the calculations (see Section VI-C). This might restrict the suitability of our approach to systems whose state uncertainty is bounded over time (see discussion below).

3) Suitability Discussion: Robotic systems operating in uncrowded environments, i.e., very few obstacles sparsely distributed in the space, might find chance constraints to be a suitable alternative. However, the accuracy and efficiency of such approaches scale poorly as the complexity of the motion planning problem increases, i.e., more crowded
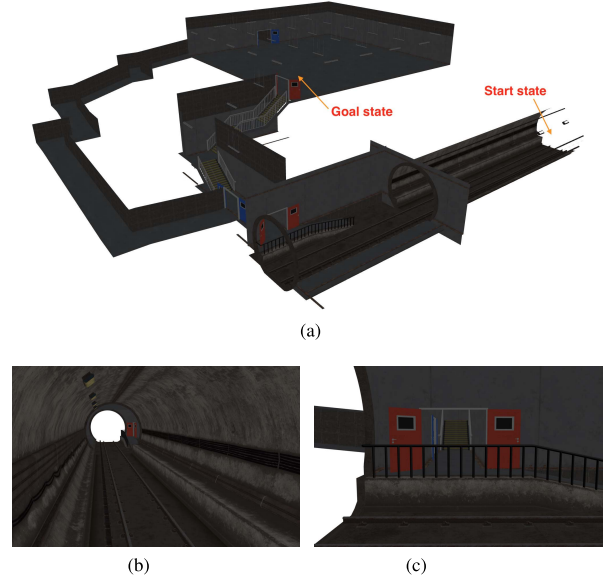
environments or higher uncertainties. As it can be observed in Fig. 15(a) and (b), this behavior endangers the ability of a planner to find a trajectory through tight apertures or narrow passages, even if one exists. If an alternative route toward the goal exists, the resulting solution will be larger than those trajectories found with less conservative approaches. Moreover, chance constraints require the representation of the environment to be a set of linear constraints, which can be prohibitively expensive to compute online, especially in applications where the environment is incrementally discovered.

In contrast, our approach trades a constant conservatism α in favor of accuracy and performance. This allows dealing with crowded environments efficiently while providing higher accuracy than chance constraints methods. Therefore, as depicted in Fig. 15(c), our probabilistic collision checking method enables a planner to find a solution through the tight corridors where chance constraints methods are over conservatist. However, our method involves higher computation times for highly uncertain states. This limitation might be relevant for systems with unbounded uncertainty, but most robotic systems are endowed with state estimation algorithms that keep the state uncertainty bounded over time. Alternatively, the parameter α can be adjusted to reduce the computation time while still guaranteeing safeness.

On the whole, the presented probabilistic collision checking approach proves to be a suitable strategy for a wide range of motion planning problems under uncertainty, even for those where chance constraints struggle at finding a solution. Moreover, our method is suitable for applications building a representation of the environment online, given that

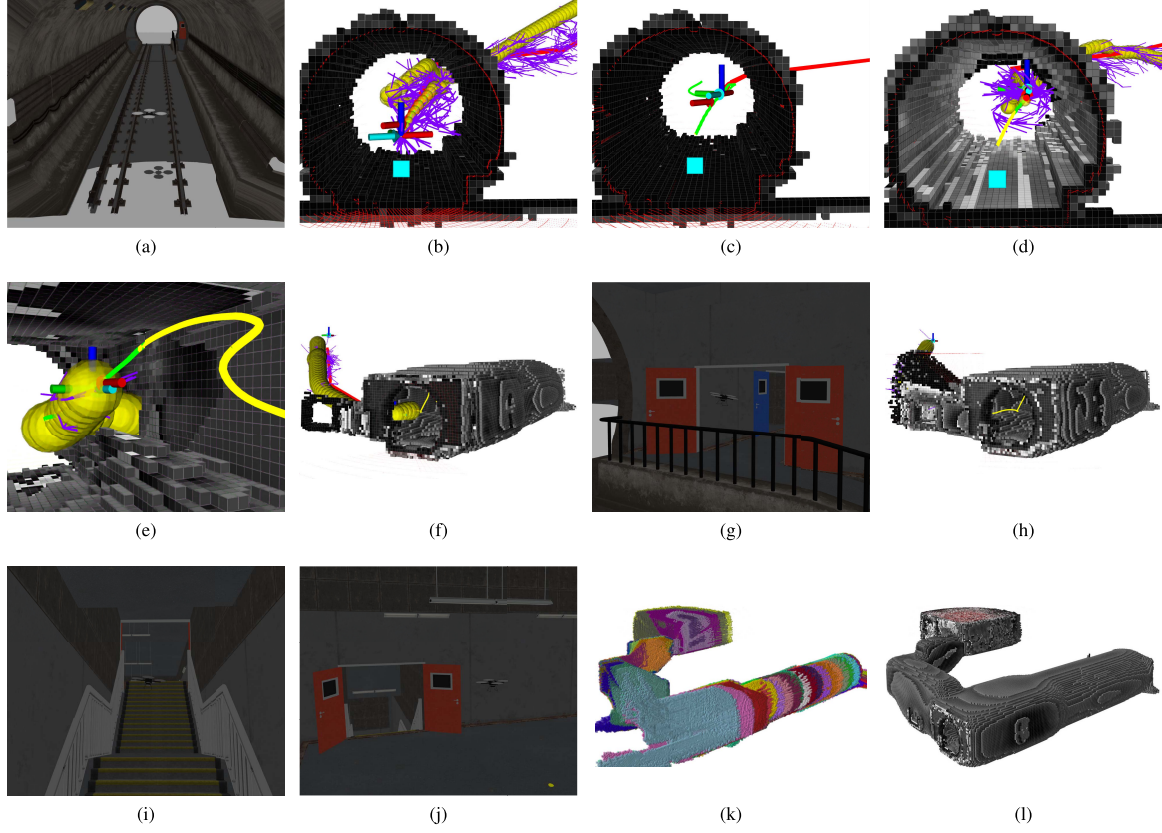Fig. 17. Online mapping and planning through the Urban Stairwell scenario of the DARPA Subterranean Challenge 2019. (a) Initial state of the quadrotor and (b) first mapping and planning iteration: geometric path (red), kinodynamic tree satisfying the probabilistic safety guarantees $p_{\text{safe}} = 0.95$ (magenta), and resulting trajectory (green) with the associated uncertainty propagation (yellow). (c) and (d) When the previous trajectory is partially invalidated due to the incremental knowledge of the surroundings, the framework finds a new trajectory toward the goal. Note that the previously observed patches of the environment become more uncertain (grayish areas) as the robot moves. (e) and (f) Entrance to the narrow stairwell is fully mapped and the framework successfully plans through it despite the considerable accumulated uncertainty. (g), (h), and (i) As the robot moves into the stairwell, the framework continues iterating over the mapping–planning process to ensure safe navigation until (j) reaching the goal region. (k) Incremental set of local maps composing the discovered environment during the mission, and (l) corresponding cumulative map $F_{\mathcal{X}}^{R'}$ (only showing those voxels $P(\mathbf{v}) > 0.4$ for visualization purposes).

those usually exploit the efficient encoding of occupancy grid maps.

D. Navigation in Unknown 3-D Environments

The proposed framework as a whole has been deployed on a simulated quadrotor unmanned aerial vehicle (UAV) [49] equipped with a 3-D Light Detection and Ranging (LIDAR). The considered environment is the Urban Stairwell scenario of the DARPA Subterranean Challenge 2019. This scenario is challenging due to its extensive workspace of approximately $40 \times 50 \times 15$ m and all narrow passages that must be traversed to accomplish the requested start-to-goal motion planning query. Fig. 16 illustrates the Urban Stairwell scenario altogether with the defined start-to-goal query. In these experiments, the quadrotor's dynamics are approximated to those of a fixed-wing plane, as described in Appendix A-B, and the surroundings are mapped online from the sensor's data at a resolution of 0.2 m. The remaining parameters of the mapping module are as those in the experiments reported in Section VII-A. During the mission, no localization updates

are considered to test the planner in the most adversarial conditions, i.e., large environmental and localization uncertainties. The required probabilistic safety guarantees are $p_{\text{safe}} = 0.95$, and the planning time is $\Delta T_{\text{MP}} = 1.5$ s, distributed as $\Delta T_L = 0.3$ s and $\Delta T_C = 1.2$ s.

Fig. 17 depicts some snapshots[7] of the online mapping and planning procedure in the Urban Stairwell scenario of the DARPA Subterranean Challenge 2019. Noteworthy is that the mesh of the Urban Stairwell scenario is composed of a total of 108 512 faces. Although these faces could be potentially approximated online from the sensor's data and used as linear constraints in [9] and [46], it would imply checking for collision against 30 times more linear constraints than those considered in Section VII-C, for which chance constraints methods already showed poor performance due to their over conservatism. Instead, our framework efficiently deals with such complex scenarios online. All in all, the proposed framework demonstrates its suitability for probabilistically

---

[7]A complete trial through the Urban Stairwell scenario of the DARPA Subterranean Challenge 2019 can be seen in https://youtu.be/I5X_QFKDpeI

safe autonomous navigation in hostile and unknown environments.

VIII. Conclusion

This article has presented a novel end-to-end framework that probabilistically guarantees the robot's safety when navigating in unexplored environments. The proposed approach is twofold: 1) incrementally maps the vehicle's surroundings to build an uncertain representation of the environment and 2) plans feasible trajectories (according to the robot's kinodynamic constraints) with probabilistic safety guarantees (according to the uncertainties in the vehicle's localization, motion, and mapping). Our proposed approach includes a multilayered planning strategy that enables for faster exploration of the high-dimensional belief space while preserving asymptotically optimal and completeness guarantees, and an efficient evaluation and tighter bound on the computation of the probability of collision than other uncertainty-aware planners in the literature. Overall, the framework is capable to deal with high-dimensional problems online while being suitable for systems with limited onboard computation power. Experimentation conducted in simulation shows some of the theoretical qualities of this work. In addition, simulated and real-world trials on an AUV and a quadrotor UAV demonstrated the suitability of the framework to guarantee the robot's safety while navigating in unexplored environments and dealing with real-robot constraints.
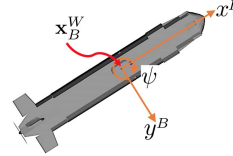
The framework is not restricted to the presented experimental evaluation or a specific platform. Any other mobile robot, either terrestrial, maritime, or aerial system, can benefit from this work. The modularity of the proposed framework allows for multiple extensions and variations. Foremost, although the experimental evaluation of the proposed framework has been conducted considering the worst case scenario of open-loop navigation without uncertainty update, the framework can bear with periodic navigation updates. An interesting possible feature that could be added to the framework is the use of the truncation trick, i.e., to uniquely propagate the posterior of the estimation, which is in no collision. However, truncating the system's belief involves approximating the posterior to a Gaussian distribution. Another possible extension is leveraging the multiresolution encoding of octomaps to check the compliance of the safety guarantee at different resolutions. Formulating this process as a multiresolution kernel checking could speed up computations even further. Finally, the conducted experimentation pointed out that automatically adjusting the replanning period might be beneficial, as well as studying more intelligent methods to leverage from the lead path or even prior solutions.

Appendix A
Kinematic Models

A. Unicycle System

For the particular case of a torpedo-shaped autonomous underwater vehicle (AUV) that operates at a constant depth, i.e., in a 2-D workspace $\mathcal{W} = \mathbb{R}^2$, with configuration space SE(2), the vehicle's motion model can be approximated by a (second-order) unicycle system

$$\dot{x} = v \cos(\psi)$$
$$\dot{y} = v \sin(\psi)$$
$$\dot{\psi} = \omega$$
$$\dot{v} = a$$

where x and y correspond to the Cartesian coordinates of the system with respect to a predefined reference frame, ψ is the system's orientation around the z-axis, v is the vehicle's forward velocity, ω is the vehicle's turning rate, and a is the acceleration. Thus, the system's state is defined as $\mathbf{x} = (x, y, \psi, v)^T$, and the system's control input is defined as $\mathbf{u} = (\omega, a)^T$.

The model above approximates the AUV's behavior, but, in an underwater environment, it is subject to uncertain external forces, e.g., current. To capture this uncertainty in the dynamics, the vehicle motion is modeled as a Gaussian process. The system's motion model is first linearized by using a dynamic feedback linearization controller as presented in [15]. This technique: 1) transforms the state of the closed-loop system to $\mathbf{z} = (x, y, \dot{x}, \dot{y})^T$ and 2) applies a proportional derivative (PD) controller on the model to drive the system toward a desired state $\mathbf{r}$. Then, the differences between the real system and the linearized closed-loop model can be approximated by a Gaussian distribution, and the closed-loop system can be represented as a Gaussian process as in (2) and (3) with states $\mathbf{z} \in \mathcal{X} = \mathbb{R}^4$, and controls $\mathbf{r} \in \mathcal{U} = \mathcal{X}$. Thus, the system state $\mathbf{z}_k$ is best described by its probability distribution in the belief space $\mathcal{B}$, i.e., $b_k = \mathcal{N}(\hat{\mathbf{z}}_k, \Sigma_{\mathbf{z}_k})$. The evolution of the belief is then given by the independent propagation of its mean and covariance as
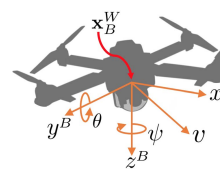
$$\hat{\mathbf{z}}_{k+1} = A\hat{\mathbf{z}}_k + B\mathbf{r}_k \tag{26}$$
$$\Sigma_{\mathbf{z}_{k+1}} = A \Sigma_{\mathbf{z}_k} A^T + \Sigma_{\mathbf{w}} \tag{27}$$

where $A \in \mathbb{R}^{4\times4}$ and $B \in \mathbb{R}^{4\times4}$ define the closed-loop linearized equations of motion with the PD controller as in [15], and $\Sigma_{\mathbf{w}}$ is the covariance of the noise modeling the discrepancies between (26) and the real system behavior.

B. Fixed-Wing System

Although more complex models could be used to represent the motion capabilities of an AUV or a quadrotor unmanned aerial vehicle (UAV) operating in a 3-D workspace $\mathcal{W} = \mathbb{R}^3$ with configuration space SE(3), both vehicle's motion model can be approximated by a fixed-wing system

$$\dot{x} = v \cos(\psi) \cos(\theta)$$
$$\dot{y} = v \sin(\psi) \cos(\theta)$$
$$\dot{z} = v \sin(\theta)$$
$$\dot{\psi} = \omega$$
$$\dot{\theta} = q$$

where x, y, and z correspond to the Cartesian coordinates of the system with respect to a predefined reference frame, ψ and θ, respectively, define the system's orientation around

the z-axis and the y-axis, v is the vehicle's forward velocity, and ω and q are the vehicle's turning rate. Thus, the system's state is defined as $\mathbf{x} = (x,\ y,\ z,\ \psi,\ \theta)^T$, and the system's control input is defined as $\mathbf{u} = (v,\ \omega,\ q)^T$.

Similar to [26], the Gaussian process describing the UAV's motion model is learned from the simulated data. The training data are extracted from the UAV's simulator producing a varied set of stationary excitations via the control input $\mathbf{u}$. Relevant control inputs are selected with the above system's model to maximize information on the output system's state $\mathbf{x}$.

Further discussion on methods for modeling robots with (partially) unknown dynamics as Gaussian processes is available in [34] and [52].

APPENDIX B
GAUSSIAN RELATIONSHIPS

This appendix summarizes the calculation of spatial relationships via the inverse and compound operators. These elemental transformations can be composed to calculate more complex spatial relationships. The interested reader may wish to consult [70] for a more thorough explanation about Gaussian relationships than the brief introduction that follows.

A. Inverse Relationship

The inverse relationship $\ominus$ represents the Gaussian relationship $\mathbf{x}_i^j$ as a function of $\mathbf{x}_j^i$ as

$$\mathbf{x}_i^j := \ominus \mathbf{x}_j^i. \tag{28}$$

The first-order estimates of the mean and the covariance of the compounding operation are

$$\hat{\mathbf{x}}_i^j \approx \ominus \hat{\mathbf{x}}_j^i \tag{29}$$

$$\Sigma_{\mathbf{x}_i^j} \approx \mathbf{J}_\ominus \Sigma_{\mathbf{x}_j^i} \mathbf{J}_\ominus^T \tag{30}$$

where

$$\mathbf{J}_\ominus := \frac{\partial \mathbf{x}_i^j}{\partial \mathbf{x}_j^i}. \tag{31}$$

B. Compound Relationship

The compounding operation $\oplus$ computes the Gaussian relationship $\mathbf{x}_k^i$ from two spatial relationships $\mathbf{x}_j^i$ and $\mathbf{x}_k^j$ that are arranged head-to-tail as

$$\mathbf{x}_k^i := \mathbf{x}_j^i \oplus \mathbf{x}_k^j. \tag{32}$$

The first-order estimates of the mean and the covariance of the compounding operation are

$$\hat{\mathbf{x}}_k^i \approx \hat{\mathbf{x}}_j^i \oplus \hat{\mathbf{x}}_k^j \tag{33}$$

$$\Sigma_{\mathbf{x}_k^i} \approx \mathbf{J}_\oplus \begin{bmatrix} \Sigma_{\mathbf{x}_j^i} & \Sigma_{\left(\mathbf{x}_j^i, \mathbf{x}_k^j\right)} \\ \Sigma_{\left(\mathbf{x}_k^j, \mathbf{x}_j^i\right)} & \Sigma_{\mathbf{x}_k^j} \end{bmatrix} \mathbf{J}_\oplus^T \tag{34}$$

where $\mathbf{J}$ denotes the Jacobian, i.e., the matrix of partial derivatives

$$\mathbf{J}_\oplus := \frac{\partial \mathbf{x}_j^i \oplus \mathbf{x}_k^j}{\partial \left(\mathbf{x}_j^i, \mathbf{x}_k^j\right)} = \frac{\partial \mathbf{x}_k^i}{\partial \left(\mathbf{x}_j^i, \mathbf{x}_k^j\right)} \tag{35}$$

$$= \begin{bmatrix} \mathbf{J}_{1\oplus} & \mathbf{J}_{2\oplus} \end{bmatrix} = \begin{bmatrix} \frac{\partial \mathbf{x}_k^i}{\partial \mathbf{x}_j^i} & \frac{\partial \mathbf{x}_k^i}{\partial \mathbf{x}_k^j} \end{bmatrix}. \tag{36}$$

TABLE III
CRITICAL VALUES t_α COMPUTED WITH (39) FOR DIFFERENT CONFIDENCE LEVELS α AND GAUSSIAN DISTRIBUTION DIMENSIONALITIES

n-D	Confidence level α [%]				
	85.0	90.0	95.0	99.0	99.9
1	1.4395	1.6449	1.9600	2.5758	3.2905
2	1.9479	2.1460	2.4477	3.0349	3.7169
3	2.3059	2.5003	2.7955	3.3682	4.0331

If the relationships $\mathbf{x}_j^i$ and $\mathbf{x}_k^j$ are independent, i.e., $\Sigma_{\left(\mathbf{x}_j^i, \mathbf{x}_k^j\right)} = 0$, (34) can be rewritten as

$$\Sigma_{\mathbf{x}_k^i} \approx \mathbf{J}_{1\oplus} \Sigma_{\mathbf{x}_j^i} \mathbf{J}_{1\oplus}^T + \mathbf{J}_{2\oplus} \Sigma_{\mathbf{x}_k^j} \mathbf{J}_{2\oplus}^T. \tag{37}$$

APPENDIX C
α-KERNEL CONSTRUCTION

A Gaussian distribution $\mathcal{N}(\hat{\mathbf{x}},\ \Sigma_{\mathbf{x}})$ describing a state's belief b is continuous and extends over the entire belief space. For the required computations, b is represented on a discrete support $\mathcal{K}_\alpha(\Sigma_{\mathbf{x}})$, referred to as α-kernel, with resolution h and size m_d at each dimension d as defined by

$$m_d = 2\ \text{ceil}\left(\frac{t_\alpha \sigma_d}{h}\right) + 1 \tag{38}$$

where the kernel size m_d is always odd, σ_d is the standard deviation of $\Sigma_{\mathbf{x}}$ along dimension d, and the critical value t_α is computed from the desired confidence level $\alpha \in [0, 1]$ as

$$t_\alpha = -\phi^{-1}\left(\frac{1}{2}(1 - \alpha)\right) \tag{39}$$

where $\phi^{-1}(\cdot)$ denotes the quantile function of the d-dimensional Gaussian normal distribution describing the system's belief b. Table III shows some critical values t_α according to commonly desirable confidence levels α for 1-D, 2-D, and 3-D Gaussian distributions.

Noteworthy is that the confidence level α involves a tradeoff between computational performance and accuracy. On one hand, α bounds the extend of the resulting $\mathcal{K}_\alpha(\cdot)$ over the belief space, thus determining the total number of voxels in the kernel and, consequently, having an impact on the computational load of the probabilistic collision checking formulated in (23). On the other hand, (23) introduces a constant conservatism α, implying that α must be selected such that $\alpha > p_{\text{safe}}$. Otherwise, the method will not find any valid state. This requirement is implicit in the probabilistic collision checking formulated in (23).

The value of each cell $n \in \mathcal{K}_\alpha(\Sigma_{\mathbf{x}})$ can be drawn from the corresponding Gaussian distribution as $h^D \mathcal{N}(\mathbf{x} \,|\, \hat{\mathbf{x}},\ \Sigma_{\mathbf{x}})$, where h^D is a normalizing constant according to the kernel resolution h and $\mathbf{x}$ is the point coordinate of n referenced at $\hat{\mathbf{x}}$. For the particular case of a multivariate Gaussian $\mathcal{N}(\hat{\mathbf{x}},\ \Sigma_{\mathbf{x}})$ with diagonal covariance matrix $\Sigma_{\mathbf{x}}$, its elements can be written as $\Sigma_{ij} = \sigma_i^2 \mathbb{I}_{ij}$, where $\mathbb{I}_{ij}$ are the matrix elements of the identity matrix (so $\mathbb{I}_{ij} = 0$ if $i \neq j$ and $\mathbb{I}_{ij} = 1$).

Then, the multivariate Gaussian with diagonal $\Sigma_{ij} = \sigma_i{}^2 \mathbb{I}_{ij}$ factorizes into a product of univariate Gaussians as

$$\mathcal{N}(\mathbf{x} \,|\, \hat{\mathbf{x}},\, \Sigma_{\mathbf{x}}) = h^D \prod_{i=1}^{D} \mathcal{N}\big(\mathbf{x}_i \,|\, \hat{\mathbf{x}},\, \sigma_{\mathbf{x}_i}^2\big) \qquad (40)$$

where, for any arbitrary positive definite covariance matrix $\Sigma_{\mathbf{x}}$, the resulting distribution is normalized. The property in (40) provides a computationally efficient strategy to build any d-dimensional kernel $\mathcal{K}(\cdot)$ from 1-D Gaussian signals.

It is worth mentioning that the kernel computation can be conducted and stored offline for different kernel sizes. At the running time, the planner would uniquely need to retrieve in a lookup table fashion the required kernel. Although this is an option to speed up the performance of the presented probabilistic collision checking, the implementation in this work computes the kernels online.

Acknowledgment

References

[1] A.-A. Agha-Mohammadi, S. Chakravorty, and N. M. Amato, "FIRM: Sampling-based feedback motion-planning under motion uncertainty and imperfect measurements," *Int. J. Robot. Res.*, vol. 33, no. 2, pp. 268–304, 2014.

[2] A.-A. Agha-Mohammadi, S. Agarwal, S.-K. Kim, S. Chakravorty, and N. M. Amato, "SLAP: Simultaneous localization and planning under uncertainty via dynamic replanning in belief space," *IEEE Trans. Robot.*, vol. 34, no. 5, pp. 1195–1214, Oct. 2018.

[3] A.-A. Agha-Mohammadi, E. Heiden, K. Hausman, and G. Sukhatme, "Confidence-rich grid mapping," *Int. J. Robot. Res.*, vol. 38, nos. 12–13, pp. 1352–1374, Oct. 2019.

[4] A. Agha *et al.*, "NeBula: Quest for robotic autonomy in challenging environments; TEAM CoSTAR at the DARPA subterranean challenge," 2021, *arXiv:2103.11470*. [Online]. Available: http://arxiv.org/abs/2103.11470

[5] R. Alterovitz, T. Simeon, and K. Goldberg, "The stochastic motion roadmap: A sampling framework for planning with Markov motion uncertainty," in *Proc. Robot., Sci. Syst. III*, Jun. 2007, pp. 233–241.

[6] F. Andert, F. Adolf, L. Goormann, and J. Dittrich, "Mapping and path planning in complex environments: An obstacle avoidance approach for an unmanned helicopter," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2011, pp. 745–750.

[7] H. Balta *et al.*, "Integrated data management for a fleet of search-and-rescue robots," *J. Field Robot.*, vol. 34, no. 3, pp. 539–582, May 2017.

[8] T. Beckers, J. Umlauft, and S. Hirche, "Stable model-based control with Gaussian process regression for robot manipulators," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 3877–3884, Jul. 2017.

[9] L. Blackmore, M. Ono, and B. C. Williams, "Chance-constrained optimal path planning with obstacles," *IEEE Trans. Robot.*, vol. 27, no. 6, pp. 1080–1094, Dec. 2011.

[10] M. Carreras *et al.*, "Testing Sparus II AUV, an open platform for industrial, scientific and academic applications," *Instrum. Viewpoint*, vol. 18, pp. 54–55, Feb. 2021.

[11] M. Chen *et al.*, "FaSTrack: A modular framework for real-time motion planning and guaranteed safe tracking," *IEEE Trans. Autom. Control*, early access, Feb. 16, 2021, doi: 10.1109/TAC.2021.3059838.

[12] M. D. S. Arantes, C. F. M. Toledo, B. C. Williams, and M. Ono, "Collision-free encoding for chance-constrained nonconvex path planning," *IEEE Trans. Robot.*, vol. 35, no. 2, pp. 433–448, Apr. 2019.

[13] N. Dadkhah and B. Mettler, "Survey of motion planning literature in the presence of uncertainty: Considerations for UAV guidance," *J. Intell. Robot. Syst.*, vol. 65, no. 1, pp. 233–246, 2012.

[14] R. De Iaco, S. L. Smith, and K. Czarnecki, "Learning a lattice planner control set for autonomous vehicles," 2019, *arXiv:1903.02044*. [Online]. Available: http://arxiv.org/abs/1903.02044

[15] A. De Luca, G. Oriolo, and M. Vendittelli, "Stabilization of the unicycle via dynamic feedback linearization," in *Proc. 6th IFAC Symp. Robot Control*, 2000, pp. 397–402.

[16] L. E. Dubins, "On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents," *Amer. J. Math.*, vol. 79, no. 3, pp. 497–516, 1957.

[17] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping: Part I," *IEEE Robot. Autom. Mag.*, vol. 13, no. 2, pp. 99–110, Jun. 2006.

[18] D. D. Fan, K. Otsu, Y. Kubo, A. Dixit, J. Burdick, and A.-A. Agha-Mohammadi, "STEP: Stochastic traversability evaluation and planning for safe off-road navigation," in *Robotics: Science and Systems*. RSS Found., 2021, pp. 1–21.

[19] P. Fankhauser, M. Bloesch, and M. Hutter, "Probabilistic terrain mapping for mobile robots with uncertain localization," *IEEE Robot. Autom. Lett.*, vol. 3, no. 4, pp. 3019–3026, Oct. 2018.

[20] E. Frazzoli, M. A. Dahleh, and E. Feron, "Maneuver-based motion planning for nonlinear systems with symmetries," *IEEE Trans. Robot.*, vol. 21, no. 6, pp. 1077–1091, Dec. 2005.

[21] D. Fridovich-Keil, S. L. Herbert, J. F. Fisac, S. Deglurkar, and C. J. Tomlin, "Planning, fast and slow: A framework for adaptive real-time safe trajectory planning," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2018, pp. 387–394.

[22] D. Fridovich-Keil, J. F. Fisac, and C. J. Tomlin, "Safely probabilistically complete real-time planning and exploration in unknown environments," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2019, pp. 7470–7476.

[23] E. Galceran, R. Campos, N. Palomeras, D. Ribas, M. Carreras, and P. Ridao, "Coverage path planning with real-time replanning and surface reconstruction for inspection of three-dimensional underwater structures using autonomous underwater vehicles," *J. Field Robot.*, vol. 32, no. 7, pp. 952–983, Oct. 2015.

[24] S. Ghosh, K. Otsu, and M. Ono, "Probabilistic kinematic state estimation for motion planning of planetary rovers," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2018, pp. 5148–5154.

[25] K. Hauser and Y. Zhou, "Asymptotically optimal planning by feasible kinodynamic planning in a state–cost space," *IEEE Trans. Robot.*, vol. 32, no. 6, pp. 1431–1443, Dec. 2016.

[26] P. Hemakumara and S. Sukkarieh, "Learning UAV stability and control derivatives using Gaussian processes," *IEEE Trans. Robot.*, vol. 29, no. 4, pp. 813–824, Aug. 2013.

[27] J. D. Hernandez, M. Moll, E. Vidal, M. Carreras, and L. E. Kavraki, "Planning feasible and safe paths online for autonomous underwater vehicles in unknown environments," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2016, pp. 1313–1320.

[28] J. D. Hernández, E. Vidal, M. Moll, N. Palomeras, M. Carreras, and L. E. Kavraki, "Online motion planning for unexplored underwater environments using autonomous underwater vehicles," *J. Field Robot.*, vol. 36, no. 2, pp. 370–396, 2019.

[29] B.-J. Ho, P. Sodhi, P. Teixeira, M. Hsiao, T. Kusnur, and M. Kaess, "Virtual occupancy grid map for submap-based pose graph SLAM and planning in 3D environments," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2018, pp. 2175–2182.

[30] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "OctoMap: An efficient probabilistic 3D mapping framework based on octrees," *Auto. Robots*, vol. 34, no. 3, pp. 189–206, Apr. 2013.

[31] F. S. Hover *et al.*, "Advanced perception, navigation and planning for autonomous in-water ship hull inspection," *Int. J. Robot. Res.*, vol. 31, no. 12, pp. 1445–1464, Oct. 2012.

[32] D. Hsu, J.-C. Latombe, and R. Motwani, "Path planning in expansive configuration spaces," in *Proc. Int. Conf. Robot. Automat.*, vol. 3, Apr. 1997, pp. 2719–2726.

[33] V. A. Huynh, S. Karaman, and E. Frazzoli, "An incremental sampling-based algorithm for stochastic optimal control," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2012, pp. 2865–2872.

[34] J. Jackson, L. Laurenti, E. Frew, and M. Lahijanian, "Safety verification of unknown dynamical systems via Gaussian process regression," 2020, *arXiv:2004.01821*. [Online]. Available: http://arxiv.org/abs/2004.01821

[35] L. Janson, E. Schmerling, and M. Pavone, "Monte Carlo motion planning for robot trajectory optimization under uncertainty," in *Robotics Research*. Cham, Switzerland: Springer, 2018, pp. 343–361.

[36] L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. H. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE Trans. Robot. Automat.*, vol. 12, no. 4, pp. 566–580, Aug. 1996.

[37] S.-K. Kim *et al.*, "PLGRIM: Hierarchical value learning for large-scale exploration in unknown environments," in *Proc. Int. Conf. Automated Planning Scheduling*, vol. 31, May 2021, pp. 652–662.

[38] S. M. LaValle and J. J. Kuffner, Jr., "Randomized kinodynamic planning," *Int. J. Robot. Res.*, vol. 20, no. 5, pp. 378–400, 2001.

[39] S. M. LaValle and R. Sharma, "A framework for motion planning in stochastic environments: Modeling and analysis," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 1995, pp. 3057–3062.

[40] J. Le Ny and G. J. Pappas, "On trajectory optimization for active sensing in Gaussian process models," in *Proc. 48h IEEE Conf. Decis. Control (CDC) Held Jointly With 28th Chin. Control Conf.*, Dec. 2009, pp. 6286–6292.

[41] Y. Li, Z. Littlefield, and K. E. Bekris, "Asymptotically optimal sampling-based kinodynamic planning," *Int. J. Robot. Res.*, vol. 35, no. 5, pp. 528–564, Feb. 2016.

[42] Y. Lin and S. Saripalli, "Path planning using 3D Dubins curve for unmanned aerial vehicles," in *Proc. Int. Conf. Unmanned Aircr. Syst. (ICUAS)*, May 2014, pp. 296–304.

[43] W. Liu and M. H. Ang, "Incremental sampling-based algorithm for risk-aware planning under motion uncertainty," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2014, pp. 2051–2058.

[44] I. Lluvia, E. Lazkano, and A. Ansuategi, "Active mapping and robot exploration: A survey," *Sensors*, vol. 21, no. 7, p. 2445, Apr. 2021.

[45] B. Luders, M. Kothari, and J. How, "Chance constrained RRT for probabilistic robustness to environmental uncertainty," in *Proc. AIAA Guid., Navigat., Control Conf.*, 2010, p. 8160.

[46] B. D. Luders, S. Karaman, and J. P. How, "Robust sampling-based motion planning with asymptotic optimality guarantees," in *Proc. AIAA Guid., Navigat., Control (GNC) Conf.*, Aug. 2013, p. 5097.

[47] R. Luna, M. Lahijanian, M. Moll, and L. E. Kavraki, "Optimal and efficient stochastic motion planning in partially-known environments," in *Proc. 28th AAAI Conf. Artif. Intell.*, Jul. 2014, pp. 2549–2555.

[48] A. Majumdar and R. Tedrake, "Funnel libraries for real-time robust feedback motion planning," *Int. J. Robot. Res.*, vol. 36, no. 8, pp. 947–982, Jun. 25, 2017.

[49] J. Meyer, A. Sendobry, S. Kohlbrecher, U. Klingauf, and O. von Stryk, "Comprehensive simulation of quadrotor UAVs using ROS and gazebo," in *Proc. 3rd Int. Conf. Simulation, Modeling Program. Auto. Robots (SIMPAR)*, 2012, pp. 400–411.

[50] H. Moravec and A. Elfes, "High resolution maps from wide angle sonar," in *Proc. IEEE Int. Conf. Robot. Autom.*, vol. 2, Mar. 1985, pp. 116–121.

[51] G. Morgenthal and N. Hallermann, "Quality assessment of unmanned aerial vehicle (UAV) based visual inspection of structures," *Adv. Struct. Eng.*, vol. 17, no. 3, pp. 289–302, Mar. 2014.

[52] D. Nguyen-Tuong, M. Seeger, and J. Peters, "Model learning with local Gaussian process regression," *Adv. Robot.*, vol. 23, no. 15, pp. 2015–2034, 2009.

[53] H. Oleynikova, Z. Taylor, M. Fehr, R. Siegwart, and J. Nieto, "Voxblox: Incremental 3D Euclidean signed distance fields for on-board MAV planning," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2017, pp. 1366–1373.

[54] K. Otsu, G. Matheron, S. Ghosh, O. Toupet, and M. Ono, "Fast approximate clearance evaluation for rovers with articulated suspension systems," *J. Field Robot.*, vol. 37, no. 5, pp. 768–785, Aug. 2020.

[55] E. Pairet, J. D. Hernandez, M. Lahijanian, and M. Carreras, "Uncertainty-based online mapping and motion planning for marine robotics guidance," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2018, pp. 2367–2374.

[56] E. Pairet, C. Chamzas, Y. R. Petillot, and L. Kavraki, "Path planning for manipulation using experience-driven random trees," *IEEE Robot. Autom. Lett.*, vol. 6, no. 2, pp. 3295–3302, Apr. 2021.

[57] L. Palmieri, S. Koenig, and K. O. Arras, "RRT-based nonholonomic motion planning using any-angle path biasing," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2016, pp. 2775–2781.

[58] C. Park, J. Pan, and D. Manocha, "ITOMP: Incremental trajectory optimization for real-time replanning in dynamic environments," in *Proc. 22nd Int. Conf. Automated Planning Scheduling*, 2012, pp. 207–215.

[59] S. Patil, J. van den Berg, and R. Alterovitz, "Estimating probability of collision for safe motion planning under Gaussian motion and sensing uncertainty," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2012, pp. 3238–3244.

[60] P. Pinies and J. D. Tardos, "Scalable SLAM building conditionally independent local maps," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2007, pp. 3466–3471.

[61] P. Pinies and J. D. Tardos, "Large-scale SLAM building conditionally independent local maps: Application to monocular vision," *IEEE Trans. Robot.*, vol. 24, no. 5, pp. 1094–1106, 2008.

[62] M. Pivtoraiko and A. Kelly, "Kinodynamic motion planning with state lattice motion primitives," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Sep. 2011, pp. 2172–2179.

[63] E. Plaku, "Region-guided and sampling-based tree search for motion planning with dynamics," *IEEE Trans. Robot.*, vol. 31, no. 3, pp. 723–735, Jun. 2015.

[64] E. Plaku, L. E. Kavraki, and M. Y. Vardi, "Motion planning with dynamics by a synergistic combination of layers of planning," *IEEE Trans. Robot.*, vol. 26, no. 3, pp. 469–482, 2010.

[65] M. Prats, J. Perez, J. J. Fernandez, and P. J. Sanz, "An open source tool for simulation and supervision of underwater intervention missions," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2012, pp. 2577–2582.

[66] F. Ramos and L. Ott, "Hilbert maps: Scalable continuous occupancy mapping with stochastic gradient descent," *Int. J. Robot. Res.*, vol. 35, no. 14, pp. 1717–1730, Jan. 2016.

[67] J. A. Reeds and L. A. Shepp, "Optimal paths for a car that goes both forwards and backwards," *Pacific J. Math.*, vol. 145, no. 2, pp. 367–393, Oct. 1990.

[68] J. S. Willners, D. Gonzalez-Adell, J. D. Hernández, È. Pairet, and Y. Petillot, "Online 3-dimensional path planning with kinematic constraints in unknown environments using hybrid A* with tree pruning," *Sensors*, vol. 21, no. 4, p. 1152, Feb. 2021.

[69] S. Scherer, S. Singh, L. Chamberlain, and M. Elgersma, "Flying fast and low among obstacles: Methodology and experiments," *Int. J. Robot. Res.*, vol. 27, no. 5, pp. 549–574, May 2008.

[70] R. Smith, M. Self, and P. Cheeseman, "Estimating uncertain spatial relationships in robotics," in *Autonomous Robot Vehicles*. New York, NY, USA: Springer, 1990, pp. 167–193.

[71] D. Strawser and B. Williams, "Approximate branch and bound for fast, risk-bound stochastic path planning," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2018, pp. 7047–7054.

[72] I. A. Şucan, M. Moll, and L. E. Kavraki, "The open motion planning library," *IEEE Robot. Autom. Mag.*, vol. 19, no. 4, pp. 72–82, Dec. 2012.

[73] W. Sun, S. Patil, and R. Alterovitz, "High-frequency replanning under uncertainty using parallel sampling-based motion planning," *IEEE Trans. Robot.*, vol. 31, no. 1, pp. 104–116, Feb. 2015.

[74] S. Suresh, P. Sodhi, J. G. Mangelson, D. Wettergreen, and M. Kaess, "Active SLAM using 3D submap saliency for underwater volumetric exploration," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2020, pp. 3132–3138.

[75] J. van den Berg, P. Abbeel, and K. Goldberg, "LQG-MP: Optimized path planning for robots with motion uncertainty and imperfect state information," *Int. J. Robot. Res.*, vol. 30, no. 7, pp. 895–913, 2011.

[76] E. Vidal, M. Moll, N. Palomeras, J. D. Hernandez, M. Carreras, and L. E. Kavraki, "Online multilayered motion planning with dynamic constraints for autonomous underwater vehicles," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2019, pp. 8936–8942.

[77] J. Wang and B. Englot, "Fast, accurate Gaussian process occupancy maps via test-data octrees and nested Bayesian fusion," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2016, pp. 1003–1010.

[78] D. J. Webb and J. van den Berg, "Kinodynamic RRT*: Optimal motion planning for systems with linear differential constraints," 2012, *arXiv:1205.5088*. [Online]. Available: https://arxiv.org/abs/1205.5088

[79] M. Yguel, O. Aycard, and C. Laugier, "Update policy of dense maps: Efficient algorithms and sparse representation," in *Field and Service Robotics*. Berlin, Germany: Springer, 2008, pp. 23–33.

[80] D. Youakim, P. Cieslak, A. Dornbush, A. Palomer, P. Ridao, and M. Likhachev, "Multirepresentation, multiheuristic A* search-based motion planning for a free-floating underwater vehicle-manipulator system in unknown environment," *J. Field Robot.*, vol. 37, no. 6, pp. 925–950, Sep. 2020.

Èric Pairet (Associate Member, IEEE) received the B.Sc. degree in electronic engineering from the University of Girona, Girona, Spain, in 2015, the Erasmus Mundus M.Sc. degree in computer vision and robotics from the University of Burgundy, Dijon, France, University of Girona, and Heriot-Watt University, Edinburgh, U.K., in 2017, and the M.Sc. degree in robotics and autonomous systems from The University of Edinburgh, Edinburgh, and Heriot-Watt University, Edinburgh Campus, Edinburgh, in 2018, where he is currently pursuing the Ph.D. degree in robotics.

He is currently a Senior Researcher with the Technology Innovation Institute, Abu Dhabi, UAE. His research is focused on techniques capable of extracting and leveraging motion plan abstractions to efficiently define robotics behavior, particularly for object manipulation and robot navigation applications.

Juan David Hernández (Senior Member, IEEE) received the B.Sc. degree in electronic engineering from Pontifical Xavierian University, Bogotá, Colombia, in 2009, the M.Sc. degree in robotics and automation from the Technical University of Madrid, Madrid, Spain, in 2012, and the Ph.D. degree in technology (robotics) from the University of Girona, Girona, Spain, in 2017.

He worked as a Robotics Research Engineer with the Netherlands Organization for Applied Scientific Research (TNO), The Hague, The Netherlands, from 2017 to 2018. He was a Post-Doctoral Research Associate with Rice University, Houston, TX, USA, from 2018 to 2019. He was a Senior Engineer for simulation of autonomous systems with Apple Inc., Sunnyvale, CA, USA, from 2019 to 2020. He is currently a Lecturer (Assistant Professor) with Cardiff University, Cardiff, U.K., where he is part of the Center for AI, Robotics and Human-Machine Systems (IROHMS). His research is focused on motion planning algorithms and human–robot collaboration.

Dr. Hernández is a Senior Member of the IEEE Robotics and Automation Society.

Marc Carreras (Member, IEEE) received the B.Sc. degree in industrial engineering and the Ph.D. degree in computer engineering from the University of Girona, Girona, Spain, in 1998 and 2003, respectively.

From 1999 to 2019, he has participated in 24 research projects (European and national projects). He is currently an Associate Professor with the Computer Vision and Robotics Institute, University of Girona. He is an author of about 150 publications. He has supervised five Ph.D. dissertations and participated in several European autonomous underwater vehicle (AUV) competitions (five times winner). His research activity mainly focuses on underwater robotics in research topics, such as intelligent control architectures, robot learning, path planning, and AUV design, modeling, and identification.

Yvan Petillot (Member, IEEE) received the Telecommunications degree with a specialization in image and signal processing, the M.Sc. degree in optics and signal processing, and the Ph.D. degree in real-time pattern recognition using optical processors from the Université de Bretagne Occidentale, Ecole Nationale Supérieure des Télécommunications de Bretagne (ENSTBr), Brest, France, in 1991, 1992, and 1996, respectively.

He is currently a Professor of robotics and autonomous systems with Heriot-Watt University, Edinburgh, U.K. He is interested in developing robotics solutions in the marine sector, supporting his long-term vision of robot teams operating in hazardous environments, supported by remote human operators. Over the last 20 years, he has made major contributions to robot perception, navigation, and planning in the underwater domain.

Morteza Lahijanian (Member, IEEE) received the B.S. degree in bioengineering from the University of California at Berkeley, Berkeley, CA, USA, in 2005, and the M.S. and Ph.D. degrees in mechanical engineering from Boston University, Boston, MA, USA, in 2009 and 2013, respectively.

He was a Research Scientist with the Department of Computer Science, University of Oxford, Oxford, U.K. He is currently an Assistant Professor with the Department of Aerospace Engineering Sciences and Department of Computer Science (by courtesy) with the University of Colorado Boulder, Boulder, CO, USA, and also the Director of Assured, Robust, and Interactive Autonomous (ARIA) Systems Group. He conducted post-doctoral research at the Department of Computer Science, Rice University.

Critical Review - Closing

In this part of the thesis, we have explored the applicability of behavioural abstractions to address motion synthesis problems that are intractable with traditional methods. In particular, we have framed the challenge of planning in unknown environments under kinodynamic and probabilistic constraints. To that, we formalised a strategy that extracts and leverages behavioural abstractions on-the-fly to solve the ever-changing motion synthesis problem in bounded time. To conclude this part, we provide a summary of the key results and discuss how the findings form a coherent piece of this thesis.

13.1 Results

Our work on behavioural abstractions to tackle ever-changing high-dimensional uncertain planning problems motivated contributing on many fronts. We supported our findings with experimental evaluation, in both synthetic environments and real-world robotic platforms. In particular, we showed the following results:

Chapter 12 (first) We demonstrated the proposed multi-layered to enable the solving of particularly high-dimensional and complex problems. We provided a throughout analysis of the performance of the two proposed strategies for adaptive lead-bias exploitation under varied parametrisation. Results showed that the proposed multi-layered planning strategy enables rapid exploration of the high-dimensional belief space while preserving asymptotic optimality and completeness guarantees.

Chapter 12 (second) We demonstrated our probabilistic state validation routine supports the multi-layered planner in the need of online motion synthesis. A thorough benchmark against state-of-the-art uncertainty-aware state validation methods shows that the proposed approach provides a tighter probability bound (higher accuracy), and does so by several orders of magnitude faster.

Chapter 12   We demonstrated the overall framework as a whole in real-world in-water
(third)   experiments using a non-holonomic torpedo-shaped autonomous underwater
             vehicle (AUV), and simulated trials in the Stairwell scenario of the DARPA
             Subterranean Challenge 2019 on a quadrotor unmanned aerial vehicle (UAV). The
             results demonstrated the efficacy of the method as well as its suitability for
             systems with limited onboard computational power.

13.2   Discussion

This part of the thesis has reported the work on online behavioural abstractions to rapidly
solve planning problems that are ever-changing, and intractable with traditional methods. Our
efforts in that direction have resulted in two notable contributions that enable dealing with high-
dimensional problems online, provide probabilistic safety guarantees in planning problems with
stochastic nature, while being suitable for systems with limited onboard computation power:

Chapter 12   A multi-layered planning strategy that promotes faster exploration of the high-
(first)   dimensional belief space with asymptotically optimal and completeness guarantees.

Chapter 12   An efficient evaluation and tighter bound on the computation of the probability of
(second)   collision than other uncertainty-aware planners in the literature.

The first stage of our multi-layered planning strategy aims at extracting relevant behavioural
navigation hints. Despite this objective is similar to that more generic of Part II, the contributed
technical approach is completely different given the contrasting availability of prior information
for the underlying problem. Then, similarly to the aims in Part III, the second stage pursues
leveraging the computed lead to calculate a trajectory that satisfies all constraints imposed in
the planning problem. Yet, oppositely to the manipulation applications targeted in the previous
parts of the thesis, while navigating both stages occur recursively on-the-fly to counteract the
lack of prior information and satisfy the online computation constraints.

Importantly, the proposed framework is not restricted to the presented experimental evaluation
nor robotic platform; any other mobile robot, either terrestrial, maritime or aerial system can
benefit from this work. In fact, work developed in this part of the thesis has been employed in
other research lines within the ORCA Hub project, e.g., [122, 31, 29, 28, 30, 173, 172], which
include a variety of robotic applications and platforms.

We have discussed some interesting directions for future work in the corresponding manuscript.
We are particularly keen on extending our contributions to leverage distinctive environmental
features to guide the exploration of the space. We elaborate on this thought in Chapter 15.

Part V

FINAL REMARKS

Conclusions

"The more I learn, the more I realize how much I don't know"

— Albert Einstein

In this thesis, we have explored motion synthesis methods via leverage of problem abstractions to make real-world deployment in complex environments practical. To this end, we have contributed several novel approaches to motion planning that account for abstractions of the problem at hand to ease and speed up the solving. Primarily, this thesis has focused on abstractions for robotic manipulation tasks but, the later part of the thesis, investigates the application of our theoretical findings to other problems such as mobile-base navigation in unknown environments.

First, in Part II, we explored approaches to bootstrap robotic behaviour. Here, we first proposed storing experienced motions in a library of motions, to then recall it to bootstrap robotic behaviour (see Chapter 4 [123]). We then introduced an affordance-guided strategy to inform the bootstrapping of the library of motions (see Chapter 5 [5]).

Then, in Part III, building on the capability of bootstrapping robot behaviour from a library of experiences, we focused on the challenge of generalising those experiences to novel task contexts. Our first contribution on this thread has been a hierarchical framework that modulates an ongoing DMP-encoded policy to avoid obstacles (see Chapter 8 [124]). Our second contribution are two new experience-based planners (the uni-directional experience-driven random trees (ERT) and its bi-directional version ERTConnect), which follow a tree sampling-based strategy to iteratively exploit a single prior path experience to ease the capture of connectivity of the space (see Chapter 9 [125]).

Finally, in Part IV, we elaborated on the leverage of a lead to guide a planner's search towards relevant parts of the planning space, to solve efficiently challenging path planning problems from which, mostly, there is no available prior knowledge. We motivated this case with the ever-changing high-dimensional uncertain planning problem posed by a mobile base robot

navigating in an unknown environment. On this thread, we made two novel contributions that make such complex problem tractable: (i) a multi-layered planning strategy that promotes faster exploration of the high-dimensional belief space with asymptotically optimal and completeness guarantees, and (ii) an efficient evaluation and tighter bound on the computation of the probability of collision than other uncertainty-aware planners in the literature (see Chapter 12 [126]).

Then, in a similar vein as our planners in Chapter 9 [125], our scheme's second layer exploits the computed lead to guide the connectivity search towards relevant parts of the belief space.

Throughout this thesis, both theoretical and experimental development has been conducted without losing sight of the connection of motion planning to other tightly linked robotic problems. As such, we tested our methods on a range of settings in both simulation and real-world experiments: from high-dimensional humanoids and robotic arms with a focus on autonomous manipulation in resembling environments, to high-dimensional kinematic motion planning with a focus on autonomous safe navigation in unknown environments. Furthermore, we have been able to apply, diversify and motivate some continuing work from the research in this thesis, such as on the study of affordance-driven grasp and path planning [121, 8, 6, 7, 4], to explore synergies between task and motion planning [31, 29, 28, 30], to investigate planning strategies for navigation, inspection, and digital-twin applications [127, 122, 173, 172], and to research shared-autonomy strategies to keep the human-in-the-loop in the autonomy pipeline [148, 103, 27, 3]. We have made some of our contributions open-source hoping they will be of use to the robotics community at large.

Future Directions

"Imagine, and it shall be. There are no limits"

— Evelyn Skye

The research conducted in this thesis has exposed a range of possible avenues for future work. While we investigated some of these alongside the main research direction of this thesis, e.g., [7, 122, 31, 29, 173, 27], many others remain unaddressed. We detail several of these below.

15.1 Single-, Joint- and Cross-trajectory Behavioural Models

In this thesis, we focused on behavioural models represented as trajectories, and the exploitation of those based on a single best candidate policy. Such an approach has proven to be versatile to tackle a variety of motion planning problems efficiently. There are several lessons learned in the course of our research that point at attractive ideas to investigate further the usage of trajectories for behaviour-driven motion synthesis.

Employing raw trajectories has the benefit of no information loss; all behavioural samples are kept at the cost of a large memory footprint. A remarkable challenge that we have not delved into is to pick the most relevant sample to the underlying planning problem. In Chapter 9 [125], inspired by [17], we estimated the suitability using the Euclidean distance between the start and goal configurations of the current planning problem, with that of the trajectories. We verified the approach to select a unique prior from a library of experiences experimentally; although it led to good results, the topic merits further attention. Potentially, other metrics besides distance and trajectory similarity measures (see relevant reviews [108, 159]) might represent better the suitability of a behavioural sample on a novel task instance. Similar metrics might potentially aid in identifying the most distinctive subset of trajectory samples that promote generalisation in new task instantiations and behavioural sparsity, thus lowering the memory footprint.

Alternatively to employing raw trajectories, these can be encoded jointly to reduce storage load. To this aim, one can use resources from the machine learning community to derive a policy from a set of raw trajectories, or fuse them to compose a roadmap (see seminal work [37]). Both options are promising behavioural supports moving forward, as they enable inferring and recalling trajectory samples and, indirectly, leveraging multiple experiences simultaneously, such that the local exploration is conducted with the most suitable segment in the library. Such a strategy would potentially allow formalising behavioural models that are more robust to changes in the planning context.

At this point, we have argued the need for more informative models for each particular behaviour. An interesting thought is that when dealing with a particular task there might be relevant information in non-related behavioural models, such as invariant-constraint robot state sample. An exciting avenue is to encompass all behavioural models in a single dynamic-weighted roadmap, where the weight of the edges changes according to the task. That way, instead of focusing exploitation within a unique categorical behaviour, the dynamic-weighted roadmap would prioritise exploitation of problem-related motions while allowing to exploit invariant constraints seen in other tasks.

15.2   Constraint- and Environment-aware Exploitation

In this thesis, we have mostly focused on the leverage of the motion plans, i.e., trajectories. Nonetheless, there are other elements in the planning problem that can be taken into account to define behavioural abstractions in favour of efficient motion synthesis, e.g., motion constraints, performance preferences and environmental information.

Task constraints and performance preferences, such as exerting a particular force/torque or turning a valve (see relevant survey [87]), optimising motion according to a desired metric, for instance, length or smoothness (see relevant summaries [176, 47]), or producing human-favoured motions (see relevant study [109]), are common in many planning problems. While some of these requisites can be enforced within exploratory routines, they tend to lead to long computation times. A potential alternative could be leveraging behavioural samples that already satisfy, or were obtained, under these constraints. Intuitively, if a solution to the current planning problem lies in the neighbourhood of a behavioural sample, the invariant robotic constraints encoded in the sample itself are more likely to prevail and thus, ease the planner's computations. Applying some of the findings in this thesis to constraint- and preference-aware motion synthesis is an exciting area for further investigation.

Leveraging environmental information to bias search towards interesting regions of the space has recently demonstrated significant progress on speeding up the motion synthesis process, e.g., [67, 98, 34, 114]. Exploiting environmental features alongside trajectory-encoded behaviour is a promising line for future work; environment-derived robot state samples can, for instance, indicate middle waypoints where to bias a behaviour. Likewise, environmental collisions and constraints violations can be useful to determine the most suitable candidate among multiple samples of a behavioural model or, when using a roadmap, to weight the conflicting segments accordingly. Metrics reflecting these violations might aid in adjusting the reliance on a selected behavioural sample, i.e., trade-off exploration and exploitation, by regulating the neighbouring range or probability of exploration around a lead trajectory.

15.3   Multi-interpretation Planning Specifications

As prospects on robotics are towards enhanced autonomy, a robotic agent is expected to deal with less structured planning specifications. Think of a robot in shared-autonomy with a human via a multi-modal interface commanded to the task "place a bottle onto the table". Such specification is open to multiple interpretations at a planning level, as there might be multiple bottles to choose from, multiple feasible grasp configurations on each bottle, as well as multiple possible locations onto the table. Multiple interpretations imply the possibility to establish a trajectory between multiple start and goal pairs. As thus, planning approaches need to cope with planning problems that are not strictly delimited by a single start and goal configuration.

A seminal work addressing such a challenge is the task-constructor in [55], which draws multiple hypotheses on each stage of a given task. Each hypothesis is corroborated with a motion synthesis planner to validate the existence of a solution. The order in which hypotheses are checked is according to multiple user-defined heuristics. Albeit this approach has proven to work for complex task problems, its individual hypothesis checking can lead to long computation times. An exciting avenue on dealing with multi-interpretation planning specifications is to consider all starts and goals pairs when discovering the connectivity of the space, e.g. [59, 72]. Leveraging such approaches altogether with behaviour-driven motion synthesis planners could potentially speed up computations drastically.

Bibliography

[1] Ron Alterovitz, Sven Koenig, and Maxim Likhachev. "Robot Planning in the Real World: Research Challenges and Opportunities". In: *AI Magazine* 37.2 (2016), pp. 76–84.

[2] Paola Ardón, Maria Eugenia Cabrera, **Èric Pairet**, Ronald PA Petrick, Subramanian Ramamoorthy, Katrin S Lohan, and Maya Cakmak. *Affordance-Aware Handovers with Human Arm Mobility Constraints*. Best presentation award. 2021. Workshop on Learning for Caregiving Robots, IEEE International Conference on Robotics and Automation.

[3] Paola Ardón, Maria Eugenia Cabrera, **Èric Pairet**, Ronald PA Petrick, Subramanian Ramamoorthy, Katrin S Lohan, and Maya Cakmak. "Affordance-Aware Handovers with Human Arm Mobility Constraints". In: *IEEE Robotics and Automation Letters* 6.2 (2021), pp. 3136–3143.

[4] Paola Ardón, **Èric Pairet**, Katrin S Lohan, Subramanian Ramamoorthy, and Ronald Petrick. "Building Affordance Relations for Robotic Agents - A Survey". In: *International Joint Conference on Artificial Intelligence* (2021).

[5] Paola Ardón, **Èric Pairet**, Yvan Petillot, Subramanian Ramamoorthy, Ronald PA Petrick, and Katrin S Lohan. "Self-Assessment of Grasp Affordance Transfer". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*. First two authors contributed equally to this work. IEEE. 2020, pp. 9385–9392.

[6] Paola Ardón, **Èric Pairet**, Ron Petrick, Subramanian Ramamoorthy, and Katrin Solveig Lohan. "Reasoning on Grasp-Action Affordances". In: *Annual Conference Towards Autonomous Robotic Systems*. Best paper award finalist. Springer. 2019, pp. 3–15.

[7] Paola Ardón, **Èric Pairet**, Ronald PA Petrick, Subramanian Ramamoorthy, and Katrin S Lohan. "Learning Grasp Affordance Reasoning through Semantic Relations". In: *IEEE Robotics and Automation Letters* 4.4 (2019), pp. 4571–4578.

[8]   Paola Ardón, **Èric Pairet**, Subramanian Ramamoorthy, and Katrin Solveig Lohan. "Towards Robust Grasps: Using the Environment Semantics for Robotic Object Affordances". In: *Proceedings of the AAAI Fall Symposium on Reasoning and Learning in Real-World Systems for Long-Term Autonomy*. AAAI Press. 2018, pp. 5–12.

[9]   Brenna D Argall, Sonia Chernova, Manuela Veloso, and Brett Browning. "A Survey of Robot Learning From Demonstration". In: *Robotics and Autonomous Systems* 57.5 (2009), pp. 469–483.

[10]   Minoru Asada, Koh Hosoda, Yasuo Kuniyoshi, Hiroshi Ishiguro, Toshio Inui, Yuichiro Yoshikawa, Masaki Ogino, and Chisato Yoshida. "Cognitive Developmental Robotics: A Survey". In: *IEEE Transactions on Autonomous Mental Development* 1.1 (2009), pp. 12–34.

[11]   Christopher G Atkeson and Jun Morimoto. "Nonparametric Representation of Policies and Value Functions: A Trajectory-based Approach". In: *Advances in Neural Information Processing Systems*. 2003, pp. 1643–1650.

[12]   Claudine Badue, Rânik Guidolini, Raphael Vivacqua Carneiro, Pedro Azevedo, Vinicius Brito Cardoso, Avelino Forechi, Luan Jesus, Rodrigo Berriel, Thiago Meireles Paixão, Filipe Mutz, et al. "Self-driving Cars: A survey". In: *Expert Systems with Applications* (2020), p. 113816.

[13]   Jérôme Barraquand, Lydia Kavraki, Jean-Claude Latombe, Rajeev Motwani, Tsai-Yen Li, and Prabhakar Raghavan. "A Random Sampling Scheme for Path Planning". In: *The International Journal of Robotics Research* 16.6 (1997), pp. 759–774.

[14]   Jérôme Barraquand and J-C Latombe. "A Monte-Carlo Algorithm for Path Planning with Many Degrees of Freedom". In: *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 1990, pp. 1712–1717.

[15]   Jerome Barraquand and Jean-Claude Latombe. "Robot Motion Planning: A Distributed Representation Approach". In: *The International Journal of Robotics Research* 10.6 (1991), pp. 628–649.

[16]   Stefan Berchtold and Bernhard Glavina. "A Scalable Optimizer for Automatically Generated Manipulator Motions". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Vol. 3. IEEE. 1994, pp. 1796–1802.

[17]   Dmitry Berenson, Pieter Abbeel, and Ken Goldberg. "A Robot Path Planning Framework that Learns from Experience". In: *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2012, pp. 3671–3678.

[18] Subhrajit Bhattacharya. "Search-based Path Planning with Homotopy Class Constraints". In: *AAAI Conference on Artificial Intelligence*. Vol. 24. 1. 2010.

[19] Aude Billard, Sylvain Calinon, Ruediger Dillmann, and Stefan Schaal. "Robot Programming by Demonstration". In: *Springer Handbook of Robotics*. Springer, 2008, pp. 1371–1394.

[20] Robert Bohlin and Lydia E Kavraki. "Path Planning using Lazy PRM". In: *IEEE International Conference on Robotics and Automation (ICRA)*. Vol. 1. IEEE. 2000, pp. 521–528.

[21] Robert Bohlin and Lydia E Kavraki. "A Randomized Approach to Robot Path Planning Based on Lazy Evaluation". In: *Combinatorial Optimisation - Dordecht* 9.1 (2001), pp. 221–249.

[22] Douglas A Bristow, Marina Tharayil, and Andrew G Alleyne. "A Survey of Iterative Learning Control". In: *IEEE Control Systems Magazine* 26.3 (2006), pp. 96–114.

[23] Oliver Brock and Oussama Khatib. "Elastic Strips: A Framework for Motion Generation in Human Environments". In: *The International Journal of Robotics Research* 21.12 (2002), pp. 1031–1052.

[24] Rodney A Brooks. "Solving the Find-path Problem by Good Representation of Free Space". In: *IEEE Transactions on Systems, Man, and Cybernetics* 2 (1983), pp. 190–197.

[25] Herman Bruyninckx. "Open Robot Control Software: the OROCOS Project". In: *IEEE International Conference on Robotics and Automation (ICRA)*. Vol. 3. 2001, pp. 2523–2528.

[26] Michael H Carr. *The Surface of Mars*. Vol. 6. Cambridge University Press, 2007.

[27] Yaniel Carreno, Pierre Le Bras, **Èric Pairet**, Paola Ardón, Mike Chantler, and Ronald PA Petrick. "An Integrated Framework for Remote Planning". In: *Proceedings of the International Conference on Automated Planning and Scheduling - Integrated Planning, Acting, and Execution Workshop*. 2021.

[28] Yaniel Carreno, **Èric Pairet**, Paola Ardón, Yvan Petillot, and Ronald PA Petrick. "Task Allocation and Planning for Offshore Mission Automation". In: *Proceedings of the International Conference on Automated Planning and Scheduling System Demonstration*. 2020.

[29] Yaniel Carreno, **Èric Pairet**, Yvan Petillot, and Ronald PA Petrick. "A Decentralised Strategy for Heterogeneous AUV Missions via Goal Distribution and Temporal Planning". In: *Proceedings of the International Conference on Automated Planning and Scheduling*. Vol. 30. 2020, pp. 431–439.

[30] Yaniel Carreno, **Èric Pairet**, Yvan Petillot, and Ronald PA Petrick. "Decentralised Task Allocation and Planning for Heterogeneous AUVs". In: *Proceedings of the International Conference on Automated Planning and Scheduling System Demonstration*. 2020.

[31] Yaniel Carreno, **Èric Pairet**, Yvan Petillot, and Ronald PA Petrick. "Task Allocation Strategy for Heterogeneous Robot Teams in Offshore Missions". In: *Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems*. 2020, pp. 222–230.

[32] Marc Carreras, Carles Candela, David Ribas, Narcís Palomeras, Lluís Magí, Angelos Mallios, Eduard Vidal, **Èric Vidal**, and Pere Ridao. "Testing SPARUS II AUV, an open platform for industrial, scientific and academic applications". In: *Instrumentation Viewpoint* 18 (2015), pp. 54–55.

[33] Constantinos   Chamzas,   Zachary   Kingston,   Carlos   Quintero-Peña, Anshumali Shrivastava, and Lydia E Kavraki. "Learning Sampling Distributions using Local 3D Workspace Decompositions for Motion Planning in High Dimensions". In: *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2021, pp. 1283–1289.

[34] Constantinos Chamzas, Anshumali Shrivastava, and Lydia E Kavraki. "Using Local Experiences for Global Motion Planning". In: *International Conference on Robotics and Automation (ICRA)*. IEEE. 2019, pp. 8606–8612.

[35] Sanjiban   Choudhury,   Jonathan   D   Gammell,   Timothy   D   Barfoot, Siddhartha S Srinivasa, and Sebastian Scherer. "Regionally Accelerated Batch Informed Trees (RABIT*): A Framework to Integrate Local Information into Optimal Path Planning". In: *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2016, pp. 4207–4214.

[36] William S Cleveland. "Robust Locally Weighted Regression and Smoothing Scatterplots". In: *Journal of the American statistical association* 74.368 (1979), pp. 829–836.

[37] David Coleman, Ioan A Şucan, Mark Moll, Kei Okada, and Nikolaus Correll. "Experience-based Planning with Sparse Roadmap Spanners". In: *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2015, pp. 900–905.

[38] Mathew DeDonato, Velin Dimitrov, Ruixiang Du, Ryan Giovacchini, Kevin Knoedler, Xianchao Long, Felipe Polido, Michael A Gennert, Taşkın Padır, Siyuan Feng, et al. "Human-in-the-loop Control of a Humanoid Robot for Disaster Response: A Report from the DARPA Robotics Challenge Trials". In: *Journal of Field Robotics* 32.2 (2015), pp. 275–292.

[39] Arthur P Dempster, Nan M Laird, and Donald B Rubin. "Maximum Likelihood from Incomplete Data via the EM Algorithm". In: *Journal of the Royal Statistical Society. Series B (Methodological)* (1977), pp. 1–38.

[40] Didier Devaurs, Thierry Siméon, and Juan Cortés. "Optimal Path Planning in Complex Cost Spaces with Sampling-based Algorithms". In: *IEEE Transactions on Automation Science and Engineering* 13.2 (2015), pp. 415–424.

[41] Edsger W Dijkstra. "A Note on Two Problems in Connexion with Graphs". In: *Numerische Mathematik* 1.1 (1959), pp. 269–271.

[42] Anca D Dragan, Nathan D Ratliff, and Siddhartha S Srinivasa. "Manipulation Planning with Goal Sets using Constrained Trajectory Optimization". In: *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2011, pp. 4582–4588.

[43] Emilio Frazzoli, Munther A Dahleh, and Eric Feron. "Maneuver-based Motion Planning for Nonlinear Systems with Symmetries". In: *IEEE Transactions on Robotics* 21.6 (2005), pp. 1077–1091.

[44] Jonathan D Gammell, Timothy D Barfoot, and Siddhartha S Srinivasa. "Batch Informed Trees (BIT*): Informed Asymptotically Optimal Anytime Search". In: *The International Journal of Robotics Research* 39.5 (2020), pp. 543–567.

[45] Jonathan D Gammell, Siddhartha S Srinivasa, and Timothy D Barfoot. "Informed RRT*: Optimal Sampling-based Path Planning Focused via Direct Sampling of an Admissible Ellipsoidal Heuristic". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2014, pp. 2997–3004.

[46] Jonathan D Gammell, Siddhartha S Srinivasa, and Timothy D Barfoot. "Batch Informed Trees (BIT*): Sampling-based Optimal Planning via the Heuristically Guided Search of Implicit Random Geometric Graphs". In: *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2015, pp. 3067–3074.

[47] Jonathan D Gammell and Marlin P Strub. "Asymptotically Optimal Sampling-Based Motion Planning Methods". In: *Annual Review of Control, Robotics, and Autonomous Systems* 4 (2021).

[48] Andrej Gams, Auke J Ijspeert, Stefan Schaal, and Jadran Lenarčič. "On-line Learning and Modulation of Periodic Movements with Nonlinear Dynamical Systems". In: *Autonomous Robots* 27.1 (2009), pp. 3–23.

[49] Andrej Gams, Bojan Nemec, Auke Jan Ijspeert, and Aleš Ude. "Coupling Movement Primitives: Interaction with the Environment and Bimanual Tasks". In: *IEEE Transactions on Robotics* 30.4 (2014), pp. 816–830.

[50] Maxim Garber and Ming C Lin. "Constraint-based Motion Planning using Voronoi Diagrams". In: *Algorithmic Foundations of Robotics V*. Springer, 2004, pp. 541–558.

[51] Roland Geraerts and Mark H Overmars. "Creating High-quality Paths for Motion Planning". In: *The International Journal of Robotics Research* 26.8 (2007), pp. 845–863.

[52] Malik Ghallab, Nick Hawes, Daniele Magazzeni, Brian C Williams, and Andrea Orlandini. "Planning and Robotics (Dagstuhl seminar 17031)". In: 7.1 (2017), pp. 32–73.

[53] Bryant Gipson, Mark Moll, and Lydia E Kavraki. "Resolution Independent Density Estimation for Motion Planning in High-dimensional Spaces". In: *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2013, pp. 2437–2443.

[54] David González, Joshué Pérez, Vicente Milanés, and Fawzi Nashashibi. "A Review of Motion Planning Techniques for Automated Vehicles". In: *IEEE Transactions on Intelligent Transportation Systems* 17.4 (2015), pp. 1135–1145.

[55] Michael Görner, Robert Haschke, Helge Ritter, and Jianwei Zhang. "Moveit! Task Constructor for Task-level Motion Planning". In: *International Conference on Robotics and Automation (ICRA)*. IEEE. 2019, pp. 190–196.

[56] Peter E Hart, Nils J Nilsson, and Bertram Raphael. "A Formal Basis for the Heuristic Determination of Minimum Cost Paths". In: *IEEE Transactions on Systems Science and Cybernetics* 4.2 (1968), pp. 100–107.

[57] Florian Hauer and Panagiotis Tsiotras. "Deformable Rapidly-Exploring Random Trees". In: *Robotics: Science and Systems*. 2017.

[58] Kris Hauser and Victor Ng-Thow-Hing. "Fast Smoothing of Manipulator Trajectories using Optimal Bounded-acceleration Shortcuts". In: *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2010, pp. 2493–2498.

[59] Juan David Hernández, Mark Moll, and Lydia E Kavraki. "Lazy Evaluation of Goal Specifications Guided by Motion Planning". In: *International Conference on Robotics and Automation (ICRA)*. IEEE. 2019, pp. 944–950.

[60] Cecilia M Heyes. "Social learning in animals: categories and mechanisms". In: *Biological Reviews* 69.2 (1994), pp. 207–231.

[61] Heiko Hoffmann, Peter Pastor, Dae-Hyung Park, and Stefan Schaal. "Biologically-inspired Dynamical Systems for Movement Generation: Automatic Real-time Goal Adaptation and Obstacle Avoidance". In: *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2009, pp. 2587–2592.

[62] David Hsu. *Randomized Single-query Motion Planning in Expansive Spaces*. Stanford University USA, 2000.

[63] David Hsu, Robert Kindel, Jean-Claude Latombe, and Stephen Rock. "Randomized Kinodynamic Motion Planning with Moving Obstacles". In: *The International Journal of Robotics Research* 21.3 (2002), pp. 233–255.

[64] David Hsu, J-C Latcombe, and Stephen Sorkin. "Placing a Robot Manipulator Amid Obstacles for Optimized Execution". In: *IEEE International Symposium on Assembly and Task Planning (ISATP)*. IEEE. 1999, pp. 280–285.

[65] David Hsu, Jean-Claude Latombe, and Rajeev Motwani. "Path Planning in Expansive Configuration Spaces". In: *International Journal of Computational Geometry & Applications* 9 (1999), pp. 495–512.

[66] Lukas Huber, Aude Billard, and Jean-Jacques Slotine. "Avoidance of Convex and Concave Obstacles with Convergence ensured through Contraction". In: *IEEE Robotics and Automation Letters* (2019).

[67] Brian Ichter, James Harrison, and Marco Pavone. "Learning Sampling Distributions for Robot Motion Planning". In: *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2018, pp. 7087–7094.

[68] Auke J Ijspeert, Jun Nakanishi, and Stefan Schaal. "Learning Attractor Landscapes for Learning Motor Primitives". In: *Advances in Neural Information Processing Systems*. 2003, pp. 1547–1554.

[69] Auke Jan Ijspeert, Jun Nakanishi, Heiko Hoffmann, Peter Pastor, and Stefan Schaal. "Dynamical Movement Primitives: Learning Attractor Models for Motor Behaviors". In: *Neural Computation* 25.2 (2013), pp. 328–373.

[70] Auke Jan Ijspeert, Jun Nakanishi, and Stefan Schaal. "Movement Imitation with Nonlinear Dynamical Systems in Humanoid Robots". In: *IEEE International Conference on Robotics and Automation (ICRA)*. Vol. 2. IEEE. 2002, pp. 1398–1403.

[71] Léonard Jaillet, Juan Cortés, and Thierry Siméon. "Sampling-based Path Planning on Configuration-space Costmaps". In: *IEEE Transactions on Robotics* 26.4 (2010), pp. 635–646.

[72] Jaroslav Jano, Vojtech Vonasek, and Robert Penicka. "Multi-goal Path Planning Using Multiple Random Trees". In: *IEEE Robotics and Automation Letters* (2021).

[73] Nikolay Jetchev and Marc Toussaint. "Fast Motion Planning from Experience: Trajectory Prediction for Speeding up Movement Generation". In: *Autonomous Robots* 34.1-2 (2013), pp. 111–127.

[74] Leslie Pack Kaelbling, Michael L Littman, and Andrew W Moore. "Reinforcement Learning: A Survey". In: *Journal of Artificial Intelligence Research* 4 (1996), pp. 237–285.

[75] Mrinal Kalakrishnan, Sachin Chitta, Evangelos Theodorou, Peter Pastor, and Stefan Schaal. "STOMP: Stochastic Trajectory Optimization for Motion Planning". In: *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2011, pp. 4569–4574.

[76] Satyam Kalan, Sanket Chauhan, Rafael F Coelho, Marcelo A Orvieto, Ignacio R Camacho, Kenneth J Palmer, and Vipul R Patel. "History of Robotic Surgery". In: *Journal of Robotic Surgery* 4.3 (2010), pp. 141–147.

[77] Ishay Kamon, Ehud Rivlin, and Elon Rimon. "A New Range-sensor based Globally Convergent Navigation Algorithm for Mobile Robots". In: *IEEE International Conference on Robotics and Automation (ICRA)*. Vol. 1. IEEE. 1996, pp. 429–435.

[78] Sertac Karaman and Emilio Frazzoli. "Incremental Sampling-based Algorithms for Optimal Motion Planning". In: *Robotics Science and Systems* 104.2 (2010).

[79] Sertac Karaman and Emilio Frazzoli. "Sampling-based Algorithms for Optimal Motion Planning". In: *The International Journal of Robotics Research* 30.7 (2011), pp. 846–894.

[80] Lydia E Kavraki, Mihail N Kolountzakis, and J-C Latombe. "Analysis of Probabilistic Roadmaps for Path Planning". In: *IEEE International Conference on Robotics and Automation (ICRA)*. Vol. 4. IEEE. 1996, pp. 3020–3025.

[81] Lydia E Kavraki, Petr Svestka, J-C Latombe, and Mark H Overmars. "Probabilistic Roadmaps for Path Planning in High-dimensional Configuration Spaces". In: *IEEE Transactions on Robotics and Automation* 12.4 (1996), pp. 566–580.

[82] Seyed Mohammad Khansari-Zadeh and Aude Billard. "A Dynamical System Approach to Realtime Obstacle Avoidance". In: *Autonomous Robots* 32.4 (2012), pp. 433–454.

[83] Oussama Khatib. "Real-time Obstacle Avoidance for Manipulators and Mobile Robots". In: *IEEE International Conference on Robotics and Automation (ICRA)*. Vol. 2. IEEE. 1985, pp. 500–505.

[84] Oussama Khatib. "Real-time Obstacle Avoidance for Manipulators and Mobile Robots". In: *Autonomous Robot Vehicles*. Springer, 1986, pp. 396–404.

[85] Donghyuk Kim, Mincheul Kang, and Sung-Eui Yoon. "Volumetric Tree*: Adaptive Sparse Graph for Effective Exploration of Homotopy Classes". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2019, pp. 1496–1503.

[86] Donghyuk Kim, Youngsun Kwon, and Sung-Eui Yoon. "Dancing PRM*: Simultaneous Planning of Sampling and Optimization with Configuration Free Space Approximation". In: *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2018, pp. 7071–7078.

[87] Zachary Kingston, Mark Moll, and Lydia E Kavraki. "Sampling-based Methods for Motion Planning with Constraints". In: *Annual Review of Control, Robotics, and Autonomous Systems* 1 (2018), pp. 159–185.

[88] James C Kinsey, Ryan M Eustice, and Louis L Whitcomb. "A Survey of Underwater Vehicle Navigation: Recent Advances and New Challenges". In: *IFAC Conference of Manoeuvering and Control of Marine Craft*. Vol. 88. 2006, pp. 1–12.

[89] Jens Kober, J Andrew Bagnell, and Jan Peters. "Reinforcement Learning in Robotics: A survey". In: *The International Journal of Robotics Research* 32.11 (2013), pp. 1238–1274.

[90] Sven Koenig and Maxim Likhachev. "D* Lite". In: *AAAI* 15 (2002).

[91] James J Kuffner and Steven M LaValle. "RRT-connect: An Efficient Approach to Single-query Path Planning". In: *International Conference on Robotics and Automation (ICRA)*. Vol. 2. IEEE. 2000, pp. 995–1001.

[92] Jinsung Kwon, Taizo Yoshikawa, and Oussama Khatib. "Elastic Strips: Implementation on a Physical Humanoid Robot". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2012, pp. 3369–3376.

[93] Jean-Claude Latombe. *Robot Motion Planning*. Vol. 124. Springer Science & Business Media, 2012.

[94] Steven M LaValle. "Rapidly-exploring Random Trees: A New Tool for Path Planning". In: (1998).

[95] Steven M LaValle. *Planning Algorithms*. Cambridge University Press, 2006.

[96] Steven M LaValle and James J Kuffner Jr. "Rapidly-exploring Random Trees: Progress and Prospects". In: (2000).

[97] Steven M LaValle and James J Kuffner Jr. "Randomized Kinodynamic Planning". In: *The International Journal of Robotics research* 20.5 (2001), pp. 378–400.

[98] Peter Lehner and Alin Albu-Schäffer. "The Repetition Roadmap for Repetitive Constrained Motion Planning". In: *IEEE Robotics and Automation Letters* 3.4 (2018), pp. 3884–3891.

[99] Timothée Lesort, Natalia Díaz-Rodríguez, Jean-Franois Goudou, and David Filliat. "State Representation Learning for Control: An Overview". In: *Neural Networks* 108 (2018), pp. 379–392.

[100]   Maxim Likhachev, David I Ferguson, Geoffrey J Gordon, Anthony Stentz, and Sebastian Thrun. "Anytime Dynamic A*: An Anytime, Replanning Algorithm". In: *International Conference on Automated Planning and Scheduling (ICAPS)*. Vol. 5. 2005, pp. 262–271.

[101]   Maxim Likhachev, Geoffrey J Gordon, and Sebastian Thrun. "ARA*: Anytime A* with Provable Bounds on sub-optimality". In: *Advances in Neural Information Processing Systems*. 2004, pp. 767–774.

[102]   Maxim Likhachev and Anthony Stentz. "R* Search". In: *23rd National Conference on Artificial Intelligence*. Vol. 1. 2008, pp. 344–350.

[103]   Katrin Lohan, Muneeb Imtiaz Ahmad, Christian Dondrup, Paola Ardón, **Èric Pairet**, and Alessandro Vinciarelli. "Adapting Movements and Behaviour to Favour Communication in Human-Robot Interaction". In: *Modelling Human Motion*. Springer, 2020, pp. 271–297.

[104]   Winfried Lohmiller and Jean-Jacques E Slotine. "On Contraction Analysis for Non-linear Systems". In: *Automatica* 34.6 (1998), pp. 683–696.

[105]   Tomas Lozano-Perez. "Spatial Planning: A Configuration Space Approach". In: *Autonomous Robot Vehicles*. Springer, 1990, pp. 259–271.

[106]   Tomás Lozano-Pérez and Michael A Wesley. "An Algorithm for Planning Collision-free Paths among Polyhedral Obstacles". In: *Communications of the ACM* 22.10 (1979), pp. 560–570.

[107]   Vladimir J Lumelsky and Alexander A Stepanov. "Path-planning Strategies for a Point Mobile Automaton Moving Amidst Unknown Obstacles of Arbitrary Shape". In: *Algorithmica* 2.1-4 (1987), pp. 403–430.

[108]   Nehal Magdy, Mahmoud A Sakr, Tamer Mostafa, and Khaled El-Bahnasy. "Review on trajectory similarity measures". In: *IEEE 7th International Conference on Intelligent Computing and Information Systems (ICICIS)*. IEEE. 2015, pp. 613–619.

[109]   Arjun Menon, Pooja Kacker, and Sachin Chitta. "Towards a Data-driven Approach to Human Preferences in Motion Planning". In: *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2015, pp. 920–927.

[110]   Wolfgang Merkt, Vladimir Ivan, and Sethu Vijayakumar. "Leveraging Precomputation with Problem Encoding for Warm-Starting Trajectory Optimization in Complex Environments". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2018, pp. 5877–5884.

[111]   Giorgio Metta, Paul Fitzpatrick, and Lorenzo Natale. "YARP: Yet Another Robot Platform". In: *International Journal of Advanced Robotic Systems* 3.1 (2006), p. 8.

[112] Giorgio Metta, Giulio Sandini, David Vernon, Lorenzo Natale, and Francesco Nori. "The iCub Humanoid Robot: An Open Platform for Research in Embodied Cognition". In: *8th Workshop on Performance Metrics for Intelligent Systems*. 2008, pp. 50–56.

[113] Johannes Meyer, Alexander Sendobry, Stefan Kohlbrecher, Uwe Klingauf, and Oskar Von Stryk. "Comprehensive Simulation of Quadrotor UAVs using ROS and Gazebo". In: *International Conference on Simulation, Modeling, and Programming for Autonomous Robots*. Springer. 2012, pp. 400–411.

[114] Daniel Molina, Kislay Kumar, and Siddharth Srivastava. "Learn and Link: Learning Critical Regions for Efficient Planning". In: *IEEE International Conference on Robotics and Automation (ICRA)*. 2020.

[115] Luis Montesano, Manuel Lopes, Alexandre Bernardino, and José Santos-Victor. "Learning Object Affordances: from Sensory-motor Coordination to Imitation". In: *IEEE Transactions on Robotics* 24.1 (2008), pp. 15–26.

[116] Chrystopher L Nehaniv, Kerstin Dautenhahn, et al. "The Correspondence Problem". In: *Imitation in Animals and Artifacts* 41 (2002).

[117] Alberto Olivares-Alarcos, Daniel Beßler, Alaa Khamis, Paulo Goncalves, Maki K Habib, Julita Bermejo-Alonso, Marcos Barreto, Mohammed Diab, Jan Rosell, João Quintas, et al. "A Review and Comparison of Ontology-based Approaches to Robot Autonomy". In: *The Knowledge Engineering Review* 34 (2019).

[118] Andreas Orthey and Marc Toussaint. "Sparse Multilevel Roadmaps for High-Dimensional Robotic Motion Planning". In: (2020).

[119] Michael Otte. "A Survey of Machine Learning Approaches to Robotic Path-planning". In: *International Journal of Robotics Research* 5.1 (2008), pp. 90–98.

[120] **Èric Pairet**. "Learning and Generalisation of Primitive Skills for Robust Dual-arm Manipulation". Master Thesis. University of Edinburgh and Heriot-Watt University, Aug. 2018.

[121] **Èric Pairet**, Paola Ardón, Frank Broz, Michael Mistry, and Yvan Petillot. "Learning and Generalisation of Primitives Skills Towards Robust Dual-arm Manipulation". In: *Proceedings of the AAAI Fall Symposium on Reasoning and Learning in Real-World Systems for Long-Term Autonomy*. AAAI Press. 2018, pp. 62–69.

[122] **Èric Pairet**, Paola Ardón, Xingkun Liu, José Lopes, Helen Hastie, and Katrin S Lohan. "A Digital Twin for Human-Robot Interaction". In: *14th ACM/IEEE International Conference on Human-Robot Interaction*. First two authors contributed equally to this work. IEEE. 2019, pp. 372–372.

[123]   **Èric Pairet**, Paola Ardón, Michael Mistry, and Yvan Petillot. "Learning and Composing Primitive Skills for Dual-arm Manipulation". In: *Annual Conference Towards Autonomous Robotic Systems.* Advanced Robotics at Queen Mary (ARQ) best paper award. Springer. 2019, pp. 65–77.

[124]   **Èric Pairet**, Paola Ardón, Michael Mistry, and Yvan Petillot. "Learning Generalizable Coupling Terms for Obstacle Avoidance via Low-Dimensional Geometric Descriptors". In: *IEEE Robotics and Automation Letters* 4.4 (2019), pp. 3979–3986.

[125]   **Èric Pairet**, Constantinos Chamzas, Yvan Petillot, and Lydia Kavraki. "Path Planning for Manipulation using Experience-driven Random Trees". In: *IEEE Robotics and Automation Letters* 6.2 (2021), pp. 3295–3302.

[126]   **Èric Pairet**, Juan David Hernández, Marc Carreras, Yvan Petillot, and Morteza Lahijanian. "Online mapping and motion planning under uncertainty for safe navigation in unknown environments". In: *IEEE Transactions on Automation Science and Engineering* (2021), pp. 1–23.

[127]   **Èric Pairet**, Juan David Hernández, Morteza Lahijanian, and Marc Carreras. "Uncertainty-based Online Mapping and Motion Planning for Marine Robotics Guidance". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems.* IEEE. 2018, pp. 2367–2374.

[128]   Luigi Palmieri, Sven Koenig, and Kai O Arras. "RRT-based Nonholonomic Motion Planning using Any-angle Path Biasing". In: *IEEE International Conference on Robotics and Automation (ICRA).* IEEE. 2016, pp. 2775–2781.

[129]   Chonhyon Park, Jia Pan, and Dinesh Manocha. "ITOMP: Incremental Trajectory Optimization for Real-time Replanning in Dynamic Environments". In: *22nd International Conference on Automated Planning and Scheduling (ICAPS).* 2012.

[130]   Dae-Hyung Park, Heiko Hoffmann, Peter Pastor, and Stefan Schaal. "Movement Reproduction and Obstacle Avoidance with Dynamic Movement Primitives and Potential Fields". In: *8th IEEE International Conference on Humanoid Robots (Humanoids).* IEEE. 2008, pp. 91–98.

[131]   Peter Pastor, Heiko Hoffmann, Tamim Asfour, and Stefan Schaal. "Learning and Generalization of Motor Skills by Learning from Demonstration". In: *IEEE International Conference on Robotics and Automation (ICRA).* IEEE. 2009, pp. 763–768.

[132]   Judea Pearl. "Heuristics: Intelligent Search Strategies for Computer Problem Solving". In: *The Addison-Wesley Series in Artificial Intelligence* (1984).

[133] Baris E Perk and Jean-Jacques E Slotine. "Motion Primitives for Robotic Flight Control". In: *arXiv preprint cs/0609140* (2006).

[134] Mike Phillips, Benjamin J Cohen, Sachin Chitta, and Maxim Likhachev. "E-Graphs: Bootstrapping Planning with Experience Graphs". In: *Robotics: Science and Systems*. 2012.

[135] Mihail Pivtoraiko and Alonzo Kelly. "Efficient Constrained Path Planning via Search in State Lattices". In: *International Symposium on Artificial Intelligence, Robotics, and Automation in Space*. 2005, pp. 1–7.

[136] Erion Plaku. "Region-guided and Sampling-based Tree Search for Motion Planning with Dynamics". In: *IEEE Transactions on Robotics* 31.3 (2015), pp. 723–735.

[137] Erion Plaku, Lydia E Kavraki, and Moshe Y Vardi. "Motion Planning with Dynamics by a Synergistic Combination of Layers of Planning". In: *IEEE Transactions on Robotics* 26.3 (2010), p. 469.

[138] Florian T Pokorny, Majd Hawasly, and Subramanian Ramamoorthy. "Topological Trajectory Classification with Filtrations of Simplicial Complexes and Persistent Homology". In: *The International Journal of Robotics Research* 35.1-3 (2016), pp. 204–223.

[139] Morgan Quigley, Brian Gerkey, Ken Conley, Josh Faust, Tully Foote, Jeremy Leibs, Eric Berger, Rob Wheeler, and Andrew Ng. "ROS: An Open-source Robot Operating System". In: *ICRA Workshop on open Source Software*. Vol. 3. 3.2. Kobe, Japan. 2009, p. 5.

[140] Sean Quinlan and Oussama Khatib. "Elastic Bands: Connecting Path Planning and Control". In: *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 1993, pp. 802–807.

[141] Lawrence Rabiner and B Juang. "An Introduction to Hidden Markov Models". In: *IEEE ASSP magazine* 3.1 (1986), pp. 4–16.

[142] Akshara Rai, Franziska Meier, Auke Ijspeert, and Stefan Schaal. "Learning Coupling Terms for Obstacle Avoidance". In: *14th IEEE International Conference on Humanoid Robots (Humanoids)*. IEEE. 2014, pp. 512–518.

[143] Akshara Rai, Giovanni Sutanto, Stefan Schaal, and Franziska Meier. "Learning Feedback Terms for Reactive Planning and Control". In: *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2017, pp. 2184–2191.

[144] Carl Edward Rasmussen and Christopher KI Williams. *Gaussian Processes for Machine Learning*. Vol. 1. MIT press Cambridge, 2006.

[145]   Nathan Ratliff, Marc Toussaint, and Stefan Schaal. "Understanding the Geometry of Workspace Obstacles in Motion Optimization". In: *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2015, pp. 4202–4209.

[146]   Nathan Ratliff, Matt Zucker, J Andrew Bagnell, and Siddhartha Srinivasa. "CHOMP: Gradient Optimization Techniques for Efficient Motion Planning". In: *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2009, pp. 489–494.

[147]   Harish Ravichandar, Athanasios S Polydoros, Sonia Chernova, and Aude Billard. "Recent Advances in Robot Learning from Demonstration". In: *Annual Review of Control, Robotics, and Autonomous Systems* 3 (2020).

[148]   David Robb, Muneeb Ahmad, Carlo Tiseo, Simona Aracri, Alistair C McConnell, Vincent Page, Christian Dondrup, Francisco Garcia, Hai Nguyen, **Èric Pairet**, Paola Ardón, Tushar Semwal, Hazel Taylor, Lindsay Wilson, David Lane, Helen Hastie, and Katrin S Lohan. "Robots in the Danger Zone: Exploring Public Perception through Engagement". In: *15th ACM/IEEE International Conference on Human-Robot Interaction*. IEEE. 2020, pp. 93–102.

[149]   Stefan Schaal and Christopher G Atkeson. "Constructive Incremental Learning from Only Local Information". In: *Neural Computation* 10.8 (1998), pp. 2047–2084.

[150]   Stefan Schaal, Auke Ijspeert, and Aude Billard. "Computational Approaches to Motor Learning by Imitation". In: *Philosophical Transactions of the Royal Society of London B: Biological Sciences* 358.1431 (2003), pp. 537–547.

[151]   John Schulman, Jonathan Ho, Alex X Lee, Ibrahim Awwal, Henry Bradlow, and Pieter Abbeel. "Finding Locally Optimal, Collision-Free Trajectories with Sequential Convex Optimization". In: *Robotics: Science and Systems*. Vol. 9. 1. Citeseer. 2013, pp. 1–10.

[152]   Sepanta Sekhavat, Petr Svestka, Jean-Paul Laumond, and Mark H Overmars. "Multilevel Path Planning for Nonholonomic Robots using Semiholonomic Subsystems". In: *The International Journal of Robotics Research* 17.8 (1998), pp. 840–857.

[153]   Ken Shoemake. "Animating Rotation with Quaternion Curves". In: *Annual Conference on Computer Graphics and Interactive Techniques*. 1985, pp. 245–254.

[154]   Siddhartha S Srinivasa, Dave Ferguson, Casey J Helfrich, Dmitry Berenson, Alvaro Collet, Rosen Diankov, Garratt Gallagher, Geoffrey Hollinger, James Kuffner, and Michael Vande Weghe. "HERB: A Home Exploring Robotic Butler". In: *Autonomous Robots* 28.1 (2010), pp. 5–20.

[155] Anthony Stentz et al. "The Focussed D* Algorithm for Real-time Replanning". In: *International Joint Conference on Artificial Intelligence (IJCAI)*. Vol. 95. 1995, pp. 1652–1659.

[156] Anthony Stentz. "Optimal and Efficient Path Planning for Partially Known Environments". In: *Intelligent Unmanned Ground Vehicles*. Springer, 1997, pp. 203–220.

[157] Martin Stolle and Christopher G Atkeson. "Policies based on Trajectory Libraries". In: *IEEE International Conference on Robotics and Automation (ICRA)*. 2006, pp. 3344–3349.

[158] Marlin P Strub and Jonathan D Gammell. "Adaptively Informed Trees (AIT*): Fast Asymptotically Optimal Path Planning Through Adaptive Heuristics". In: *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2020, pp. 3191–3198.

[159] Han Su, Shuncheng Liu, Bolong Zheng, Xiaofang Zhou, and Kai Zheng. "A Survey of Trajectory Distance Measures and Performance Evaluation". In: *The VLDB Journal* 29.1 (2020), pp. 3–32.

[160] Ioan A Şucan and Lydia E Kavraki. "Kinodynamic Motion Planning by Interior-exterior Cell Exploration". In: *Algorithmic Foundation of Robotics VIII*. Springer, 2009, pp. 449–464.

[161] Ioan A Sucan, Mark Moll, and Lydia E Kavraki. "The Open Motion Planning Library". In: *IEEE Robotics & Automation Magazine* 19.4 (Dec. 2012), pp. 72–82.

[162] Giovanni Sutanto, Zhe Su, Stefan Schaal, and Franziska Meier. "Learning Sensor Feedback Models from Demonstrations via Phase-modulated Neural Networks". In: *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2018, pp. 1142–1149.

[163] Matthew E Taylor and Peter Stone. "Transfer Learning for Reinforcement Learning Domains: A Survey". In: *Journal of Machine Learning Research* 10.Jul (2009), pp. 1633–1685.

[164] Evangelos Theodorou, Jonas Buchli, and Stefan Schaal. "A Generalized Path Integral Control Approach to Reinforcement Learning". In: *Journal of Machine Learning Research* 11.Nov (2010), pp. 3137–3181.

[165] Elin A Topp, Maj Stenmark, Alexander Ganslandt, Andreas Svensson, Mathias Haage, and Jacek Malec. "Ontology-based Knowledge Representation for Increased Skill Reusability in Industrial Robots". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2018, pp. 5672–5678.

[166] Marc Toussaint. "Robot Trajectory Optimization using Approximate Inference". In: *26th Annual International Conference on Machine Learning*. ACM. 2009, pp. 1049–1056.

[167] Marc Toussaint. "A Tutorial on Newton Methods for Constrained Trajectory Optimization and Relations to SLAM, Gaussian Process Smoothing, Optimal Control, and Probabilistic Inference". In: *Geometric and Numerical Foundations of Movements*. Springer, 2017, pp. 361–392.

[168] Eduard Vidal, Mark Moll, Narcís Palomeras, Juan David Hernández, Marc Carreras, and Lydia E Kavraki. "Online Multilayered Motion Planning with Dynamic Constraints for Autonomous Underwater Vehicles". In: *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2019, pp. 8936–8942.

[169] Sethu Vijayakumar and Stefan Schaal. "Locally Weighted Projection Regression: An O(n) Algorithm for Incremental Real Time Learning in High Dimensional Space". In: *7th International Conference on Machine Learning (ICML)*. Vol. 1. 2000, pp. 288–293.

[170] Henry Williams. "Human Inspired Robotic Path Planning and Heterogeneous Robotic Mapping". PhD Thesis. Victoria University of Wellington, 2016.

[171] Jonatan Scharff Willners, Ignacio Carlucho, Sean Katagiri, Chandler Lemoine, Joshua Roe, Dylan Stephens, Tomasz Łuczyński, Shida Xu, Yaniel Carreno, **Èric Pairet**, Sen Wang, Corina Barbalata, and Yvan Petillot. "From market-ready ROVs to low-cost AUVs". In: *OCEANS 2021*. IEEE. 2021.

[172] Jonatan Scharff Willners, Yaniel Carreno, Shida Xu, Tomasz Łuczyński, Sean Katagiri, Joshua Roe, **Èric Pairet**, Yvan Petillot, and Sen Wang. "Robust Underwater SLAM using Autonomous Relocalisation". In: *13th IFAC Conference on Control Applications in Marine Systems, Robotics, and Vehicles*. 2021.

[173] Jonatan Scharff Willners, Daniel Gonzalez-Adell, Juan David Hernández, **Èric Pairet**, and Yvan Petillot. "Online 3-Dimensional Path Planning with Kinematic Constraints in Unknown Environments Using Hybrid A* with Tree Pruning". In: *Sensors* 21.4 (2021).

[174] Melonee Wise, Michael Ferguson, Derek King, Eric Diehr, and David Dymesich. "Fetch and Freight: Standard Platforms for Service Robot Applications". In: *Workshop on Autonomous Mobile Service Robots*. 2016.

[175] Artur Wolek and Craig A Woolsey. "Model-based Path Planning". In: *Sensing and Control for Autonomous Vehicles*. Springer, 2017, pp. 183–206.

[176] Yajue Yang, Jia Pan, and Weiwei Wan. "Survey of Optimal Motion Planning". In: *IET Cyber-systems and Robotics* 1.1 (2019), pp. 13–19.

[177]   Yuandong Yang and Oliver Brock. "Elastic Roadmaps — Motion Generation for Autonomous Mobile Manipulation". In: *Autonomous Robots* 28.1 (2010), p. 113.

[178]   Matt Zucker, James Kuffner, and J Andrew Bagnell. "Adaptive Workspace Biasing for Sampling-based Planners". In: *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2008, pp. 3757–3762.