ICINCO 2024

21st International Conference on Informatics in Control, Automation and Robotics

PROCEEDINGS

Volume 2

Porto, Portugal

18 - 20 November, 2024

EDITORS

Giuseppina Gini
Radu-Emil Precup
Dimitar Filev

https://icinco.scitevents.org

SPONSORED BY
INSTICC

PAPERS AVAILABLE AT
SCITEPRESS DIGITAL LIBRARY

ICINCO 2024

Proceedings of the
21st International Conference on
Informatics in Control, Automation and Robotics

Volume 2

Porto - Portugal

November 18 - 20, 2024

Sponsored by
INSTICC - Institute for Systems and Technologies of Information, Control and Communication

IEEE Technically co-sponsored by
eIS - IEEE SMC - TC on Evolving Intelligent Systems

Technically Co-sponsored by
IFAC - International Federation of Automatic Control

ACM In Cooperation
SIGAI - ACM Special Interest Group on Artificial Intelligence

In Cooperation with
AAAI - Association for the Advancement of Artificial Intelligence
INNS - International Neural Network Society
APPIA - Portuguese Association for Artificial Intelligence

Copyright © 2024 by SCITEPRESS – Science and Technology Publications, Lda.

Edited by Giuseppina Gini, Radu-Emil Precup and Dimitar Filev

Printed in Portugal
ISSN: 2184-2809
ISBN: 978-989-758-717-7
DOI: 10.5220/0000193700003822
Depósito Legal: 536890/24

https://icinco.scitevents.org
icinco.secretariat@insticc.org

CONTENTS

INVITED SPEAKERS

KEYNOTE SPEAKERS

Control of Road Traffic Systems: A Multi-Scale Perspective
Antonella Ferrara 5

Personal Data Privacy: Especially Location
John C. Krumm 7

Youla-Kučera Parameterization: Theory and Applications
Vladimír Kučera 9

ROBOTICS AND AUTOMATION

FULL PAPERS

Expanded Applicability: Multi-Agent Reinforcement Learning-Based Traffic Signal Control in a Variable-Sized Environment
István Gellért Knáb, Bálint Pelenczei, Bálint Kõvári, Tamás Bécsi and László Palkovics 15

Intuitive Human-Robot Interface: A 3-Dimensional Action Recognition and UAV Collaboration Framework
Akash Chaudhary, Tiago Nascimento and Martin Saska 26

Automated Detection of Defects on Metal Surfaces Using Vision Transformers
Toqa Alaa, Mostafa Kotb, Arwa Zakaria, Mariam Diab and Walid Gomaa 36

EMG-Based Shared Control Framework for Human-Robot Co-Manipulation Tasks
Francesca Patriarca, Paolo Di Lillo and Filippo Arrichiello 46

A Vision Based System for Assisting Blind People at Indoor and Outdoor Exploration
Raluca Didona Brehar and Sand Elena-Andreea 54

Low-Cost Robot Construction Focused on Educational Environments
Douglas Favaretto, Vitor de Assis, Dieisson Martinelli, Andre Schneider De Oliveira and Vivian Cremer Kalempa 66

Cooperation and Synchronization of Robotic Tasks Using a Digital Twin
Alexandre Parant, Laurent Arcese, Sinuhé Martinez-Martinez and Arthur Marguery 73

Uncertainty-Aware DNN for Multi-Modal Camera Localization
M. Vaghi, A. L. Ballardini, S. Fontana and D. G. Sorrenti 80

SHORT PAPERS

Analysis of Drivers' Path Follow Behaviour
Gergõ Ferenc Ignéczi, Ernõ Horváth and Attila Borsos 93

CRANEBot: Teleoperated Crane-Suspended Robotic System for Inspection and Manipulation in Harsh Environments
Giancarlo D'Ago, Sergio Di Giovannantonio, Luca Rosario Buonocore and Mario Di Castro 101

HERA Centroiding Image Processing Algorithm Based on the Normalised Correlation with a Lambertian Sphere
Stancu Florin Adrian, Marcos Avilés Rodrigálvarez, Andrea Pellacani, Ángel Palomino Aguado, Aída Alcalde Barahona, Francesco Pace, Paul Băjănaru, Víctor Manuel Moreno Villa and Jesús Gil-Fernández 109

Adaptive Highway Traffic Management: A Reinforcement Learning Approach for Variable Speed Limit Control with Random Anomalies
Bálint Pelenczei, István Gellért Knáb, Bálint Kõvári, Tamás Bécsi and László Palkovics 117

Triplet Neural Networks for the Visual Localization of Mobile Robots
Marcos Alfaro, Juan José Cabrera, Luis Miguel Jiménez, Óscar Reinoso and Luis Payá 125

Evaluation of Open-Source OCR Libraries for Scene Text Recognition in the Presence of Fisheye Distortion
María Flores, David Valiente, Marcos Alfaro, Marc Fabregat-Jaén and Luis Payá 133

Video Summarization Techniques: A Comprehensive Review
Toqa Alaa, Ahmad Mongy, Assem Bakr, Mariam Diab and Walid Gomaa 141

RoboMorph: In-Context Meta-Learning for Robot Dynamics Modeling
Manuel Bianchi Bazzi, Asad Ali Shahid, Christopher Agia, John Alora, Marco Forgione, Dario Piga, Francesco Braghin, Marco Pavone and Loris Roveda 149

BVE + EKF: A Viewpoint Estimator for the Estimation of the Object's Position in the 3D Task Space Using Extended Kalman Filters
Sandro Costa Magalhães, António Paulo Moreira, Filipe Neves dos Santos and Jorge Dias 157

Augmented Feasibility Maps: A Simultaneous Approach to Redundancy Resolution and Path Planning
Marc Fabregat-Jaén, Adrián Peidró, Esther González-Amorós, María Flores and Óscar Reinoso 166

Automated Mission Management of Small Unmanned Aircraft Systems for Critical Events in Urban Air Traffic
Robin Müller and Maximilian Bauer 174

Unscented Transform-Based Pure Pursuit Path-Tracking Algorithm Under Uncertainty
Chinnawut Nantabut 182

A Vision Based Warning System for Safe Distance Driving with Respect to Cyclists
Raluca Brehar, Moldovan Flavia, Attila Füzes and Radu Dănescu 190

LiDAR-Based Object Recognition for Robotic Inspection of Power Lines
José Mário Nishihara de Albuquerque and Ronnier Frates Rohrich 197

NODE and Contraction Methods for Dynamics Learning from Human Expert Demonstrations
Tufail Ahmed, Sangmoon Lee and Ju H. Park 205

Human-Robot Cooperation in Disassembly: A Rapid Review 212
Sara Jacob, Nathalie Klement, Richard Bearee and Marie-Pierre Pacaux-Lemoine

Local Motion Planning for Overtaking Maneuvers in a Rural Road Environment 220
Dániel Losonczi, Árpád Fehér, Szilárd Aradi and László Palkovics

A Case Study in Building 2D Maps with Robots 228
Theodor-Radu Grumeza, Thomas-Andrei Lazăr, Isabela Drămnesc, Gabor Kusper, Konstantinos Papadopoulos, Nikolaos Fachantidis and Ioannis Lefkos

Nonlinear Control and State Estimation for the Hand Axes of a Pneumatic Robot 236
Seyed Houman Mirafzal

Applying a Systematic Approach to Design Human-Robot Cooperation in Dynamic Environments 248
Sridath Tula, Marie-Pierre Pacaux-Lemoine, Emmanuelle Grislin-Le Strugeon, Anna Ma-Wyatt and Jean-Philippe Diguet

Model-Based Digital Twin for Collaborative Robots 256
Jeshwitha Jesus Raja, Shaza Elbishbishy, Yanire Gutierrez, Ibrahim Mohamed, Philipp Kranz and Marian Daun

Advanced Techniques for Corners, Edges, and Stacked Gaps Detection and Pose Estimation of Cardboard Packages in Automated Dual-Arm Depalletising Systems 264
Santheep Yesudasu and Jean-François Brethé

Does Path Tracking Benefit from Sequential or Simultaneous RL Speed Controls? 274
Jason Chemin, Eric Lucet and Aurélien Mayoue

Multi-Modal Deep Learning Architecture Based on Edge-Featured Graph Attention Network for Lane Change Prediction 282
Petrit Rama and Naim Bajcinca

Uncertainty Hypervolume in Point Feature-Based Visual Odometry 290
InJun Mun and Sukhan Lee

Spray Quality Assessment on Water-Sensitive Paper Comparing AI and Classical Computer Vision Methods 300
Inês Simões, André Baltazar, Armando Sousa and Filipe Neves dos Santos

A Comparative Analysis of Methods for Hand Pose Detection in 3D Environments 308
Jorge G. Iglesias, Luis Montesinos and David Balderas

Effects of Cognitive Load Variation on Anthropomorphism During a Cooperative Human-Robot Pick-and-Place Task 314
Mohamed Cherif Rais, Barbara Kühnlenz and Kolja Kühnlenz

Control of Fixed-Wing Tethered Aircraft in Circular Take-Off and Landing Maneuvers 319
Sérgio Vinha, Gabriel M. Fernandes, Manuel C. R. M. Fernandes, Huu Thien Nguyen and Fernando A. C. C. Fontes

Autonomous Forklift Navigation Inside a Cluttered Logistics Factory 327
Eric Lucet, Antoine Lucazeau and Jason Chemin

A Modular Multimodal Multi-Object Tracking-by-Detection Approach, with Applications in Outdoor and Indoor Environments 336
Eduardo Borges, Luís Garrote and Urbano J. Nunes

Multimodal 6D Detection of Industrial Pallets, in Real and Virtual Environments, with Applications in Industrial AMRs 345
José Lourenço, Gonçalo Arsénio, Luís Garrote and Urbano Nunes

INDUSTRIAL INFORMATICS

FULL PAPERS

Drone Technology for Efficient Warehouse Product Localization 357
Assia Belbachir, Antonio M. Ortiz, Erik T. Hauge, Ahmed Nabil Belbachir, Giusy Bonanno, Emanuele Ciccia and Giorgio Felline

Knowledge Graph Extraction from Retrieval-Augmented Generator: An Application in Aluminium Die Casting 365
Florian Rötzer, Kai Göbel, Maximilian Liebetreu and Stephan Strommer

SHORT PAPERS

Software Toolchain for Offline-Programming a Jig-Less Fiber Placement Process Using Cooperating Robots 379
Michael Vistein, Lars Brandt, Gabriel Côté, Julien-Mathieu Audet and Bruno Monsarrat

In-Depth Analysis of Recall Initiators of Medical Devices with a Machine Learning-Natural Language Processing Tool 387
Yang Hu and Pezhman Ghadimi

Asset Administration Shell Digital Twin of 5G Communication System 395
Salvatore Cavalieri, Raffaele Di Natale and Salvatore Gambadoro

Streamlining Data Integration and Decision Support in Refinery Operations 403
Ocan Şahin, Aslı Yasmal, Mustafa Oktay Samur and Gizem Kuşoğlu Kaya

Setting up a Digital Twin for Real-Time Remote Monitoring of a Cyber-Physical System 410
Adrien Vinel, Séverine Durieux, Laurent Piétrac, Glênio Simião Ramalho and Nicolas Blanchard

Development of a Lithium-Ion Battery Lifetime Prediction Model Using Deep Learning for Short-Term Learning 418
Yu Fujitaki and Hiroyuki Kobayashi

AUTHOR INDEX 423

Augmented Feasibility Maps: A Simultaneous Approach to Redundancy Resolution and Path Planning

Marc Fabregat-Jaén[1][a], Adrián Peidró[1][b], Esther González-Amorós[1], María Flores[1][c]
and Óscar Reinoso[1,2][d]
[1]*Instituto de Investigación en Ingeniería de Elche, Universidad Miguel Hernández, Avda. Universidad s/n, Elche, Spain*
[2]*ValgrAI: Valencian Graduate School and Research Network of Artificial Intelligence, Camí de Vera s/n, Valencia, Spain*
{*mfabregat, apeidro, esther.gonzaleza, m.flores, o.reinoso*}@*umh.es*

Keywords: Redundancy Resolution, Path Planning, Feasibility Maps, Redundant Manipulators, Obstacle Avoidance.

Abstract: Redundant robotic manipulators are capable of performing complex tasks with an unprecedented level of dexterity and precision. However, their redundancy also introduces significant computational challenges, particularly in the realms of redundancy resolution and path planning. This paper introduces a novel approach to simultaneously address these challenges through the concept of Augmented Feasibility Maps, by integrating task coordinates as decision variables into the traditional feasibility maps. We validate the AFM concept by using Rapidly-Exploring Random Trees to explore the maps, demonstrating its efficacy in simulations of various dimensionalities. The method is capable of incorporating kinematic constraints, such as obstacle avoidance while adhering to joint limits.

1 INTRODUCTION

Kinematically redundant robots have more degrees of freedom n (DoF) than the dimension m of the *primary or main* task that they must perform. For example, a 3-DoF planar manipulator that must control the position coordinates p_x and p_y of its gripper in the plane has one degree of redundancy, as it would suffice to have 2 DoF to complete this task. Kinematic redundancy increases the versatility of robotic manipulators because the excess of $r = n - m$ degrees of freedom can be leveraged to satisfy *secondary* goals or constraints in addition to the main task, e.g., minimization of torques or energy consumption, or avoidance of singularities or obstacles, etc. (Peidro and Haug, 2023). However, this comes at the expense of a higher computational complexity, because their inverse kinematic problem (IKP), i.e., the problem of computing the joint displacements that allow the manipulator to place its gripper at the desired position, admits infinitely many different solutions. Deciding which one of these infinite solutions should be picked to complete its task is the problem of *redundancy resolution*.

Typically, the problem of redundancy resolution is solved for a given desired trajectory of the kinematic variables that define the task of the robot, which we denote by $\mathbf{x}$. For a specified $\mathbf{x}(t)$, with $0 < t < t_g$, a redundancy resolution algorithm should provide the time history $\mathbf{q}(t)$ of all n joint displacements (or coordinates) of the manipulator, considering the infinitude of solutions that exist for each t, as well as any existing constraints or secondary goals. Methods to solve this problem can be classified into velocity-based methods and position-based.

Velocity-based methods operate with the velocity relationship between task and joint coordinates: $\dot{\mathbf{x}} = \mathbf{J}\dot{\mathbf{q}}$, where $\mathbf{J}$ is the non-square $n \times m$ Jacobian matrix. This velocity equation is the basis of pseudoinverse and optimization approaches to redundancy resolution, which obtain $\dot{\mathbf{q}}$ and integrate it. Pseudoinverse methods (Whitney, 1969) solve joint velocities as $\dot{\mathbf{q}} = \mathbf{J}^\dagger\dot{\mathbf{x}}$ plus an additional nullspace term that optimizes secondary goals without affecting the main task (Kazemipour et al., 2022), where $(\cdot)^\dagger$ denotes the Moore-Penrose pseudoinverse. Optimization methods minimize programs to find the optimal value of $\dot{\mathbf{q}}$, where the velocity equation $\dot{\mathbf{x}} = \mathbf{J}\dot{\mathbf{q}}$ is included as an equality constraint, avoiding inversion (Zanchettin and Rocco, 2017). The main advantage of velocity-based methods is their simplicity and compatibility with real time, but they usually provide only locally

---

[a] https://orcid.org/0009-0002-4327-0900
[b] https://orcid.org/0000-0002-4565-496X
[c] https://orcid.org/0000-0003-1117-0868
[d] https://orcid.org/0000-0002-1065-8944

Fabregat-Jaén, M., Peidró, A., González-Amorós, E., Flores, M. and Reinoso, Ó.
Augmented Feasibility Maps: A Simultaneous Approach to Redundancy Resolution and Path Planning.
Paper published under CC license (CC BY-NC-ND 4.0)
In *Proceedings of the 21st International Conference on Informatics in Control, Automation and Robotics (ICINCO 2024) - Volume 2*, pages 166-173
ISBN: 978-989-758-717-7; ISSN: 2184-2809
Proceedings Copyright © 2024 by SCITEPRESS – Science and Technology Publications, Lda.

optimal trajectories, may suffer from non-holonomy and singularities (when pseudoinverting), and may fail to accurately satisfy constraints that, like collisions, are defined better at position level than at velocity level (Peidro and Haug, 2023).

These limitations are alleviated by position-based methods, which operate directly at configuration level. Solving redundancy at position level means obtaining all the solution sets of the space of joint coordinates $\mathbf{q}$ that yield a desired task value $\mathbf{x}$. Generically, these solution sets are manifolds called *self-motion manifolds* (Burdick, 1989). By scanning all self-motion manifolds for a given $\mathbf{x}$, it is possible to find the $\mathbf{q}^*$ that globally optimizes secondary criteria, allowing for globally-optimal solutions to the redundancy problem. This, however, can be a costly approach, as computing the self-motion manifolds can be rather computationally expensive (Albu-Schäffer and Sachtler, 2023). A similar approach is offered by the Feasibility Maps (FMs) (Wenger et al., 1993; Reveles et al., 2016), which represent all feasible values of a set of r redundant variables for each instant t of the trajectory, and allow for planning trajectories in these maps avoiding obstacles and singularities while globally optimizing other criteria.

Independently of the taken approach, it may occur that the desired task trajectory $\mathbf{x}(t)$ is unfeasible. This happens when the solution set of the IKP becomes empty due to singularities, joint limits or collisions, which materialize as external boundaries or internal barriers of the workspace of the manipulator (Peidro et al., 2018). When this happens, the manipulator cannot complete its task regardless of the method used.

To avoid this, in this paper we propose a method to *simultaneously* solve the redundancy problem and the path planning problem, where the latter consists in finding a feasible task trajectory $\mathbf{x}(t)$ that allows the manipulator to reach a specified goal $\mathbf{x}_g$ respecting the considered constraints. To achieve this, we begin with the idea of FMs as in (Fabregat-Jaen et al., 2023), which represent all feasible values of the redundant coordinates at each instant of the trajectory, but, as a novelty, we *augment* these maps by considering also the task coordinates as decision variables whose time evolution must also be determined.

This paper is organized as follows. First, Section 2 defines the Augmented Feasibility Maps (AFM). Next, Section 3 presents a method based on Rapidly-Exploring Random Trees (RRT) to explore the AFM and connect the initial state of the robot to the desired goal $\mathbf{x}_g$ through a path that is asymptotically optimal. Section 4 illustrates the method with one and two degrees of redundancy. Finally, conclusions and future work are discussed in Section 5.

2 AUGMENTED FEASIBILITY MAPS

Consider a manipulator whose joint configuration is given by $\mathbf{q} = [q_1, q_2, \ldots, q_n]^T$, where n represents DoF. Its forward kinematics can be expressed as a function that maps the n-dimensional joint space to the m-dimensional task space:

$$\mathbf{x} = \mathbf{f}(\mathbf{q}), \tag{1}$$

where $\mathbf{x} = [x_1, x_2, \ldots, x_m]^T$ denotes the m task coordinates.

In order to determine the IK, (1) should ideally be inverted to obtain an expression for $\mathbf{q}$ as a function of $\mathbf{x}$ (i.e., $\mathbf{q} = \mathbf{f}^{-1}(\mathbf{x})$). Nevertheless, deriving a universal IK function is often unfeasible, even for non-redundant manipulators (where $n = m$), due to the forward kinematics function generally being non-bijective, rendering the IK mapping multi-valued.

The concept of *aspects* helps interpret this phenomenon. In (Borrel and Liégeois, 1986), an aspect is defined as a connected set of points in the joint space where the Jacobian matrix remains full rank. As such, aspects delineate subsets of the joint space wherein the manipulator can move without encountering singularities. By definition, a transition between aspects is marked by a singularity, at which point the Jacobian matrix loses rank. For manipulators that must pass through singularities to change posture, the IKP yields a unique solution for each aspect. In this case, an IK function $\mathbf{g}(\cdot)$ can be defined for each aspect, mapping each task point to a unique joint position.

Multi-valuedness is further aggravated for redundant manipulators (where $n > m$), as there exist infinitely many joint configurations that can reach a given task space point. The degree of redundancy r, that characterizes a robot and task combination, is defined as $r = n - m$. In this context, the use of an *augmented task space*, which incorporates r additional dimensions to encode the redundancy, has been proposed as a means to simplify the IKP. The augmented task space is defined as:

$$\mathbf{x}_a = \left[\mathbf{x}^{\mathrm{T}}, \mathbf{q}_r^{\mathrm{T}}\right]^{\mathrm{T}} \tag{2}$$

where $\mathbf{q}_r$ corresponds to the additional r-dimensional vector that virtually removes the redundancy. The components of $\mathbf{q}_r$ must be independent of every other component of $\mathbf{x}_a$, and can be chosen arbitrarily, but usually correspond to a subset of joint coordinates or a differentiable function of them.

In (Pámanes G et al., 2002), the authors studied the effect that the extra dimensions of the augmented task space had on the Jacobian matrix. Given that the rows corresponding to $\mathbf{q}_r$ induced new singularities, a new set of aspects were produced, which were named *extended aspects*. Therefore, depending on the selection of $\mathbf{q}_r$, the number of extended aspects and their domain can vary.

The augmented task space allows for the definition of a unique augmented IK function $\mathbf{g}_a(\cdot)$ that maps every augmented-task-space point to a unique joint configuration that lies in a certain extended aspect:

$$\mathbf{q} = \mathbf{g}_a(\mathbf{x}_a) \tag{3}$$

However, rather than a single fixed task point, the IKP is often posed as the tracking of a task trajectory. Such trajectory $\mathbf{x}(t)$ can be parameterized by an arc-length parameter t (hereafter referred to as time for simplicity). In this situation, the set of joint configurations that successfully track the trajectory can be gathered in an $(r+1)$-dimensional space, where one of the dimensions of this space is time t, whereas the other r dimensions are the $\mathbf{q}_r$ coordinates. This conceptual space is referred to as *feasibility map* (FM), a term introduced in (Wenger et al., 1993). A feasibility map $\mathcal{FM}$ is defined as the set of points in the $\left[t, \mathbf{q}_r^{\mathrm{T}}\right]^{\mathrm{T}}$ space for which the augmented IKP yields a real solution that satisfies additional kinematic constraints (e.g., joint limits or obstacles), i.e.:

$$\mathcal{FM} = \left\{ \begin{bmatrix} t \\ \mathbf{q}_r \end{bmatrix} \;\middle|\; \mathbf{q} = \mathbf{g}_a(\mathbf{x}_a), \mathbf{x}_a = \begin{bmatrix} \mathbf{x}(t) \\ \mathbf{q}_r \end{bmatrix}, \mathbf{q} \in \mathcal{Q} \right\} \tag{4}$$

where $\mathcal{Q}$ denotes the joint-space subset that satisfies the kinematic constraints. Note that, for a specified task trajectory $\mathbf{x}(t)$, and a given choice of $\mathbf{q}_r$, there exist as many FMs as extended aspects, and each FM corresponds to a different augmented IK function $\mathbf{g}_a(\cdot)$ that maps the augmented task space to joint-space points within that extended aspect.

Figure 1(a) shows an example of an FM for a 2-DoF manipulator with a 1-dimensional task, hence redundancy $r = 1$, resulting in a 2D map. Note that the feasible space corresponds to the uncolored region, as the colored regions represent subsets that do not satisfy the kinematic constraints. Specifically, the red regions correspond to configurations that lead to a collision with an obstacle, while the purple regions correspond to configurations that yield complex solutions for the IKP.

In particular, this example corresponds to the manipulator depicted in Figure 2, which has two joints $\mathbf{q} = [q_1, q_2]^{\mathrm{T}}$ and a task $\mathbf{x} = p_y$, with forward kinematic function $p_y = \sin(q_1) + \sin(q_1 + q_2)$. The task trajectory is $p_y(t) = -6.662t^2 + 8.162t - 1.5$ with $t \in [0, 1]$, and the choice of redundant parameters is $\mathbf{q}_r = q_1$, hence $\mathbf{x}_a = [p_y, q_1]^{\mathrm{T}}$. The augmented IK function (3) in this example is:

$$\mathbf{q} = \begin{bmatrix} q_1 \\ \frac{\pi}{2} + \sigma\left(\frac{\pi}{2} - \arcsin(p_y(t) - \sin q_1)\right) - q_1 \end{bmatrix} \tag{5}$$

which has two solutions (i.e., extended aspects) if angles are constrained to the range $[-\pi, \pi]$ rad, corresponding to the two possible values of $\sigma = \pm 1$. This yields two possible feasibility maps; Fig 1(a) shows the one corresponding to $\sigma = -1$. The FM shown in Fig 1(a) is obtained by sweeping plane (t, q_1) between ranges $0 < t < 1$ and $-\pi < q_1 < \pi$ with a given discrete step, and for each point (t, q_1), $\mathbf{q}$ is computed using (5). If the computation of $\mathbf{q}$ yields non-real solutions (i.e., $|p_y - \sin q_1| > 1$), the corresponding point (t, q_1) is marked in purple. If it produces collisions with the ellipse shown in Figure 2, it is marked in red. Otherwise, it is left uncolored, indicating that the solution is feasible, i.e., it belongs to $\mathcal{FM}$.

FMs have been employed in the literature to address the *redundancy resolution problem*, which involves the selection of the redundant parameters $\mathbf{q}_r$ that fulfill a specified task $\mathbf{x}$ while optimizing a certain criterion and satisfying the kinematic constraints. For instance, a modified RRT algorithm that explores FMs for online redundancy resolution is proposed in (Fabregat-Jaen et al., 2023). Another example is (Ferrentino and Chiacchio, 2020), where the authors compute every FM and employ them to globally address the redundancy resolution problem through an exhaustive search across the combined maps. Nonetheless, these methods rely on a predefined task trajectory $\mathbf{x}(t)$, potentially resulting in unfeasible joint-space motions if the task trajectory is not properly designed, accounting for the kinematic constraints, singularities or internal barriers of the robot's workspace (Peidro et al., 2018).

For example, consider the FM shown in Figure 1(a). Assuming that the initial configuration of the manipulator corresponds to the FM point A $= (t = 0, q_1 = -2)$, any valid trajectory that fulfills the desired task $\mathbf{x}(t)$ should connect A with any valid point of the vertical line $t = 1$, e.g., FM point B. The trajectory T shown in Figure 1(a) accomplishes this (note that T wraps around q_1 because $-\pi$ can be identified with $+\pi$). However, if we forbid angular wrapping by setting joint limits, the only way to connect A with any point satisfying t=1 is by traversing the narrow corridor encircled by N. If the desired task trajectory $\mathbf{x}(t)$ was slightly perturbed, it may occur that the red and purple regions near N touch each other, eliminating the corridor and making it impossible to reach $t = 1$ from A unless angular wrapping is permitted, thus rendering the desired task trajectory $\mathbf{x}(t)$ unfeasible. To avoid unfeasible task trajectories, in this paper we will only specify the desired final value for $\mathbf{x}$ instead of specifying its trajectory $\mathbf{x}(t)$, and we will incorporate the task variables into the feasibility maps in order to find their time history at the same time

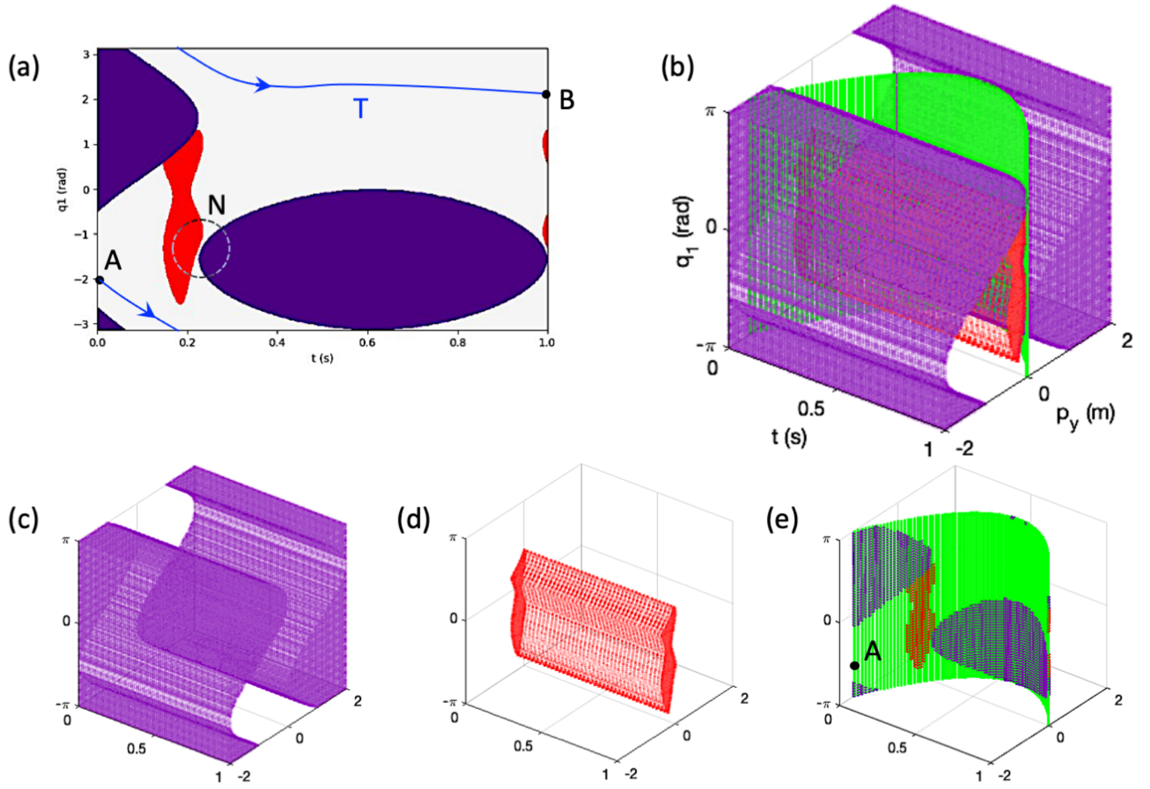

Figure 1: (a) Example of an FM for $r = 1$. (b) AFM that includes the task coordinate (p_y). (c) Region of non-real solutions. (d) Region of collisions. (e) Parabolic sheet corresponding to the task $p_y = -6.662t^2 + 8.162t - 1.5$.

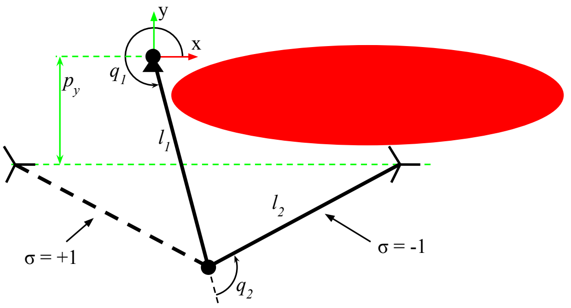

Figure 2: 2-DoF manipulator with an elliptical obstacle.

that the motion of $\mathbf{q}_r$ is determined, simultaneously solving the path planning of $\mathbf{x}(t)$ and the redundancy resolution.

To achieve this, we propose extending the concept of FMs to *augmented feasibility maps* (AFMs), which incorporate the task trajectory into the FM definition. An AFM is defined as:

$$\mathcal{AFM} = \left\{ \begin{bmatrix} t \\ \mathbf{x}_a \end{bmatrix} \,\middle|\, \mathbf{q} = \mathbf{g}_a(\mathbf{x}_a), \mathbf{x}_a = \begin{bmatrix} \mathbf{x} \\ \mathbf{q}_r \end{bmatrix}, \mathbf{q} \in \mathcal{Q} \right\} \quad (6)$$

To illustrate this idea, consider the same example of Figure 2, but instead of specifying the time history of the task $p_y(t)$, we specify only its final value $p_y(1) = 0$. If p_y can be freely varied, it can be included as an additional dimension in the FM, resulting in a 3D map (t, p_y, q_1) of feasible configurations, shown in Figure 1b. This map includes three objects that, for clarity, have been represented separately in Figures 1c-e. Firstly, Figure 1c shows some purple surfaces that enclose three-dimensional volumes made of configurations that, when substituted in (5), yield non-real solutions of $\mathbf{q}$. Secondly, the red surfaces shown in Figure 1d enclose configurations for which the end-effector of the manipulator penetrates the elliptical obstacle. Finally, the green surface shown in Figure 1e is a parabolic sheet defined parametrically as: $\{(t, p_y, q_1) \mid 0 < t < 1, p_y = -6.662t^2 + 8.162t - 1.5, -\pi < q_1 < \pi\}$. This sheet represents the task trajectory that was specified in the previous example, and Figure 1e includes the intersections between this sheet and the purple and red forbidden regions. In fact, the projection of this sheet on plane (t, q_1) yields the conventional FM depicted in Figure 1a.

Note that, if we do not specify the task trajectory $p_y(t)$, the configuration of the robot is not constrained to move along the parabolic sheet, being able to move freely along the three-dimensional feasible space that is free from non-real solutions and collisions. This increases the chances of finding a task trajectory that successfully connects point A to any feasible point of the *goal set*, which is the vertical line defined as: $\{(t, p_y, q_1) \mid t = 1, p_y = 0, -\pi < q_1 < \pi\}$.

Generalizing the previous example, we propose a method to solve the motion of redundant manipulators to reach a goal task point $\mathbf{x}_g$ in time t_g. This defines a goal set as an r-dimensional subset of the AFM where the coordinates t and $\mathbf{x}$ are fixed to their goal values (i.e., $t = t_g$ and $\mathbf{x} = \mathbf{x}_g$), and the remaining dimensions of $\mathbf{q}_r$ are free to vary. The method explores the free space of the augmented space $(t, \mathbf{x}, \mathbf{q}_r)$ to find a feasible trajectory connecting the initial configuration of the manipulator to the goal set, simultaneously solving path planning (i.e., time history of the task $\mathbf{x}$) and redundancy resolution (i.e., $\mathbf{q}_r$). The proposed method, which performs online, is described in the following section.

Other works have proposed concurrently addressing both problems, albeit through different approaches. In (Tassi et al., 2021), a method based on hierarchical quadratic programming is proposed to effectively augment the decision variables of the optimization problem with the task trajectory. The authors of (Zhou et al., 2023) propose a two-stage approach. First, an RRT is built by sampling points in the task space. In the second stage, for each sampled RRT node, self-motion manifolds are calculated, and a joint trajectory is derived based on an exhaustive search that evaluates an averaged performance index (e.g., reciprocal condition number) along each sequence of self-motion manifolds.

3 EXPLORING AFMs VIA RRT

The proposed method is based on the RRT algorithm. Particularly, we employ the RRT* variant (Karaman and Frazzoli, 2011), which is an extension that guarantees asymptotic optimality. In this section, we present the modifications required to adapt the RRT* algorithm to explore AFMs. Algorithm 1 outlines the method. The general structure of the RRT* algorithm is maintained, with the main difference being the specific definition of the procedures.

The first line of Algorithm 1 initializes the tree $\mathcal{T}$ with a root node $\mathbf{n}_0 = \mathbf{s}_0$, where $\mathbf{s}_0$ is the initial state of the robot. We define a state as a point in the AFM space $\mathbf{s} = \left[t, \mathbf{x}^{\mathrm{T}}, \mathbf{q}_r^{\mathrm{T}}\right]^{\mathrm{T}}$, while a node $\mathbf{n}$ is an element of the tree $\mathcal{T}$ that contains a stored state $\mathbf{s}$, and a pointer to its parent node $\mathbf{n}_{\text{parent}}$. The main loop of the algorithm is then entered, where $\mathcal{T}$ is expanded N times.

Algorithm 1: RRT** algorithm.

```
1  Initialize tree $\mathcal{T}$ with root node $\mathbf{n}_0 = \mathbf{s}_0$
2  for $i = 1$ to $N$
3      $\mathbf{s}_{\text{rand}} \leftarrow$ SampleState($\alpha$, $\mathbf{x}_g$)
4      $\mathbf{n}_{\text{near}} \leftarrow$ NearestNode($\mathbf{s}_{\text{rand}}$, $\mathcal{T}$)
5      $\mathbf{n}_{\text{new}} \leftarrow$ Steer($\mathbf{n}_{\text{near}}$, $\mathbf{s}_{\text{rand}}$, $\Delta s$)
6      if FeasibleConnection($\mathbf{n}_{\text{near}}$, $\mathbf{n}_{\text{new}}$)
7          $\mathcal{N} \leftarrow$ Neighbors($\mathbf{n}_{\text{new}}$, $\mathcal{T}$, $r$)
8          $\mathbf{n}_{\text{parent}} \leftarrow$ BestParent($\mathcal{N}$, $\mathbf{n}_{\text{new}}$)
9          Add $\mathbf{n}_{\text{new}}$ to $\mathcal{T}$ with parent $\mathbf{n}_{\text{parent}}$
10         $\mathcal{T} \leftarrow$ Rewire($\mathcal{T}$, $\mathcal{N}$, $\mathbf{n}_{\text{new}}$)
11 $\mathcal{P} \leftarrow$ BestPath($\mathcal{T}$)
```

In each iteration, a random state $\mathbf{s}_{\text{rand}}$ is sampled from the AFM space (line 3). The SampleState function generates a random state in the domain of the AFM. The t dimension of the AFM is constrained to the interval $[0, t_g]$, where t_g is the user-defined duration of the trajectory. The $\mathbf{x}$ domain is defined by a prior workspace analysis of the robot, and $\mathbf{q}_r$ is constraint by the joint limits. To guide the exploration towards the goal $\mathbf{x}_g$, samples are drawn from the goal set (which depends on $\mathbf{x}_g$) with probability α, which normally ranges from 0.05 to 0.2 (i.e., at most, 20% of the samples generated in line 3 of Algorithm 1 will belong to the goal set).

In line 4, NearestNode finds the node $\mathbf{n}_{\text{near}}$ in the tree $\mathcal{T}$ that is closest to $\mathbf{s}_{\text{rand}}$. The distance metric is given by a cost function defined by the user. For instance, the cost function could be the weighted norm of the differences between the states:

$$c(\mathbf{s}_1, \mathbf{s}_2) = \sqrt{(\mathbf{s}_2 - \mathbf{s}_1)^{\mathrm{T}} \mathbf{W} (\mathbf{s}_2 - \mathbf{s}_1)} \tag{7}$$

where $\mathbf{W}$ is a positive-definite diagonal matrix that weights the dimensions of the AFM space, due to the non-homogeneous units of the different coordinates of $\mathbf{s}$. The weights can be used to prioritize the minimization of certain dimensions over others. One important aspect of the NearestNode function is that it must consider the time dimension t when computing the distance between states. Since the time is strictly monotonically increasing, only the states with a time value less than that of $\mathbf{s}_{\text{rand}}$ are considered.

The Steer function (line 5) generates a new state $\mathbf{s}_{\text{new}}$ by moving an incremental distance Δs from $\mathbf{n}_{\text{near}}$ towards $\mathbf{s}_{\text{rand}}$. The Δs parameter is a user-defined constant that determines the step size of the exploration. The function is exemplified in Figure 3(a), where $\mathbf{n}_{\text{new}}$ is generated by moving from $\mathbf{n}_{\text{near}}$ towards $\mathbf{s}_{\text{rand}}$ with a step size of Δs.

Next, the FeasibleConnection function (line 6) determines if the connection between $\mathbf{n}_{\text{near}}$ and $\mathbf{n}_{\text{new}}$ is feasible (red segment in Figure 3(a)). This function evaluates if the segment between the two states satisfies the kinematic constraints, i.e., configurations are real and collision-free. If the connection is feasible, the algorithm proceeds to the two characteristic steps of the RRT* algorithm: the selection of the parent node and the rewiring of the tree.

In line 7, the NEIGHBORS function computes the set of nodes $\mathcal{N}$ in the tree $\mathcal{T}$ that are within a radius r from $\mathbf{n}_{\text{new}}$. In Figure 3, the set of neighbors $\mathcal{N}$ are represented by the nodes within the purple circle, of radius r, centered at $\mathbf{n}_{\text{new}}$. Then, the BESTPARENT function (line 8) selects the best parent node $\mathbf{n}_{\text{parent}}$ from $\mathcal{N}$. The best parent is the one that minimizes the cost function defined in (7) for the entire path from the root node $\mathbf{n}_0$ to $\mathbf{n}_{\text{new}}$, routing through $\mathbf{n}_{\text{parent}}$. The path and its cost can be computed by backtracking from $\mathbf{n}_{\text{new}}$ to its parent node, and so on until the root node is reached, accumulating the cost of each segment during the process. The tree is then updated by adding $\mathbf{n}_{\text{new}}$ as a child of $\mathbf{n}_{\text{parent}}$ (line 9). In Figure 3(b), the node $\mathbf{n}_{\text{new}}$ is connected to its best parent $\mathbf{n}_{\text{parent}}$ (purple segment), instead of the parent $\mathbf{n}_{\text{near}}$ that the original RRT would choose (red segment).

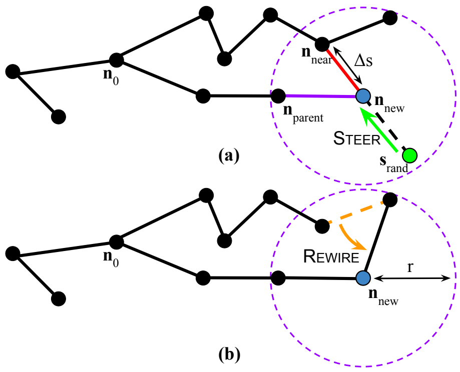
$\mathbf{n}_0$
$\mathbf{n}_{\text{near}}$
Δs
$\mathbf{n}_{\text{new}}$
$\mathbf{n}_{\text{parent}}$
STEER
$\mathbf{s}_{\text{rand}}$
(a)
$\mathbf{n}_0$
REWIRE
r
$\mathbf{n}_{\text{new}}$
(b)

Figure 3: Procedures of the RRT* algorithm.

Finally, the REWIRE function in line 10 updates the tree $\mathcal{T}$ by performing a local analysis of the neighborhood of $\mathbf{n}_{\text{new}}$. The same set of neighbors $\mathcal{N}$ as in BESTPARENT is employed, but the roles are swapped. In this case, the function evaluates if the path from the root node to each neighbor in $\mathcal{N}$ can be improved by routing through $\mathbf{n}_{\text{new}}$. If the path is improved for any neighbor, the parent of such neighbor is updated to $\mathbf{n}_{\text{new}}$, and the tree is restructured accordingly. This process is illustrated in Figure 3(b).

Once the main loop is completed, the BESTPATH function (line 11) is called to extract the best path $\mathcal{P}$ from the tree $\mathcal{T}$. Calculating the cost in a similar way to the BESTPARENT and REWIRE functions, every path from the root node to each node that lies in the goal set is evaluated, and the best path $\mathcal{P}$ is selected.

Due to the nature of the RRT* algorithm, the returned path presents discontinuous velocities, as well as infinite accelerations at the node points. To address this issue, a post-processing step is required to smooth the path. We propose employing a cubic b-spline interpolation to generate a smooth trajectory that connects the nodes of the path. Similarly to (Fabregat-Jaen et al., 2023), a series of intermediate control points are generated between each pair of consecutive nodes, and the b-spline is fitted to these points (see Figure 4). However, instead of defining a fixed number of control points per segment, we propose employing the value of Δs as the distance between control points. Since the values of Δs are small, the resulting b-spline will not deviate significantly from the original path, i.e., there is little risk that the smoothed path becomes unfeasible due to invading forbidden regions, while ensuring a smooth trajectory.

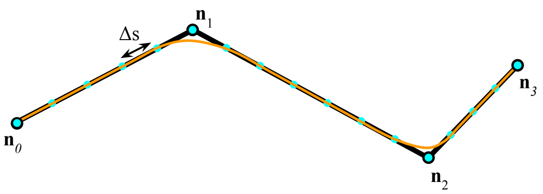

Figure 4: B-Spline interpolation (orange) of the original path (black) along the blue control points.

4 SIMULATIONS

In this section, we present results of the proposed method for different combinations of manipulators and tasks. The simulations are conducted in an Intel Core i7-9700F CPU @ 3.00GHz with 32 GB of RAM, running Ubuntu 24.04. The code is implemented in Python 3.12.

4.1 2-DoF Robot, 1-Dimensional Task

The first example corresponds to the same scenario considered in the first experiment of (Fabregat-Jaen et al., 2023). It consists of a 2-DoF planar manipulator and a 1-dimensional task, shown in Figure 2. However, instead of tasking the manipulator with tracking a predefined trajectory, only the final task point is specified, and corresponds to the p_y coordinate of the end-effector being equal to 0 at $t_g = 1$ (see green line at Figure 5(b)). The kinematic constraints are defined by the joint limits $q_i \in [-\pi, \pi]\,\text{rad}\ \forall i \in \{1,2\}$, and an elliptical obstacle that the end-effector must avoid (red ellipse in Figures 2 and 5(b)), defined by the equation $\frac{(p_x-1.1)^2}{1^2} + \frac{(p_y+0.2)^2}{0.25^2} \leq 1$. Both links of the manipulator l_1 and l_2 have a length of 1m. We selected q_1 as the single redundant parameter in $\mathbf{q}_r$.

For this example, we set the RRT* parameters as

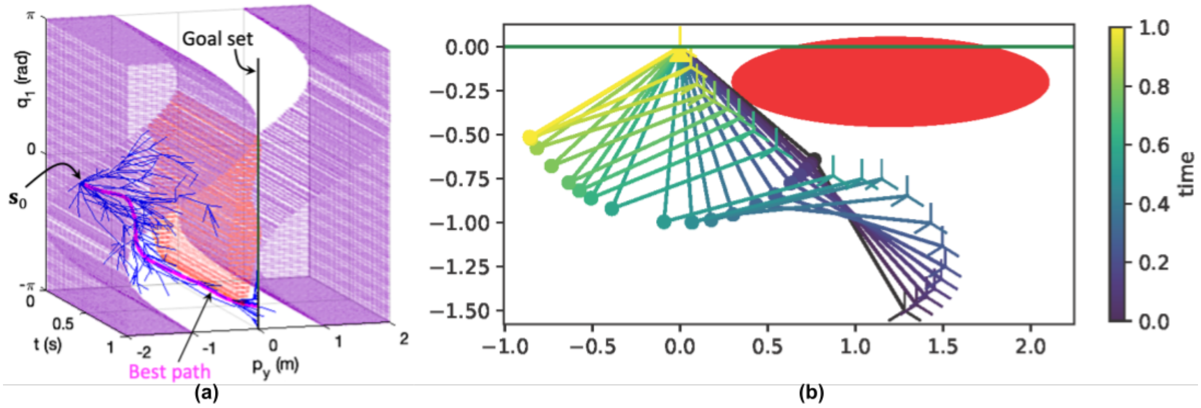

Figure 5: (a) AFM and RRT solution for a 2-DoF manipulator and a 1-dimensional task. (b) Corresponding robot motion.

follows: $\Delta s = 0.2$, $r = 0.4$, $\alpha = 0.2$ and $N = 1000$. The cost function corresponds to the one defined in (7), with $\mathbf{W} = \mathbf{I}$.

Figure 5(a) illustrates the AFM for the imposed constraints and an executed RRT* tree that explores the AFM. The same color code as in the FMs in Section 2 is employed. In addition, the best path computed is plotted in green, and its corresponding robot motion is represented in Figure 5(b) for different time values. It is worth remarking that the AFM is illustrated only for visual purposes, as the method does not require the computation of the AFM.

In addition, in Figure 6, we present a graph that shows the execution times of the RRT* algorithm, as well as the cost of the solution, averaged over 100 runs, for different values of the number of iterations N. As expected, the execution times proportionally increase with the number of iterations. As for the cost of the solution, although some variability is observed, the values seem to follow a decreasing trend as the number of iterations increases.

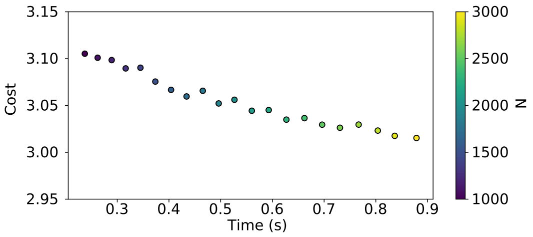

Figure 6: Execution time versus cost of the solution for 21 different values of N, averaged over 1000 runs.

4.2 3-DoF Robot, 1-Dimensional Task

Finally, we consider an RPR (Revolute-Prismatic-Revolute) planar manipulator (shown in Figure 7(a)) and a 1-dimensional task, which produces redundancy $r = 2$ and a 4-dimensional AFM. The same final task value and kinematic constraints as in the previous example are considered. However, in order to increase the complexity of the problem and showcase the method's efficiency, we consider that the entire manipulator must avoid the obstacle, rather than just the end-effector. For this purpose, we have employed the HPP-FCL library (Pan et al., 2024) to evaluate the collisions between the manipulator and the obstacle. The new prismatic joint coordinate q_2 is restricted to $q_2 \in [0,1]\,\mathrm{m}$, and its value extends the length of the first link of the manipulator. The redundant parameter vector is composed of the first revolute and the prismatic coordinates: $\mathbf{q}_r = [q_1, q_2]^{\mathrm{T}}$.

The same RRT* parameters as for the previous experiment are used, which are: $\Delta s = 0.2$, $r = 0.4$, $\alpha = 0.2$, $N = 1000$ and $\mathbf{W} = \mathbf{I}$.

Since the corresponding AFM is 4-dimensional, it is not possible to visualize it directly. Therefore, we only present different time values of the robot motion in Figure 7(b). The average execution time for 100 runs is 97.3ms, demonstrating the method's efficiency in solving the problem, even when considering whole-body collisions and a higher-dimensional AFM.

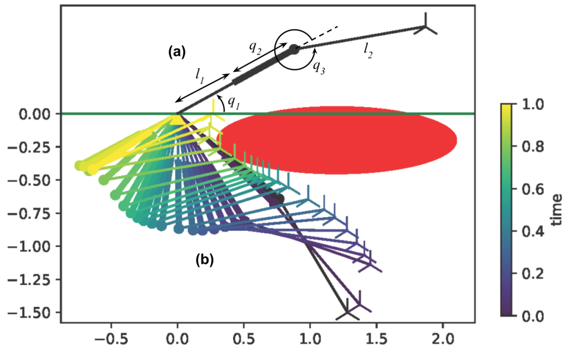

Figure 7: (a) RPR manipulator. (b) Algorithm solution.

5 CONCLUSIONS

In this paper, we have proposed a method to solve the motion of redundant manipulators to reach a goal task point in a given time. The method is based on the RRT* algorithm, which is adapted to explore AFMs, which incorporate the task trajectory into the feasibility analysis. The proposed method is capable of simultaneously solving path planning and redundancy resolution, and it is demonstrated to be efficient in two examples of planar manipulators with different degrees of redundancy. The method overcomes the limitations of previous works that require a predefined task trajectory, which may not be always feasible.

In the future, we will implement direct comparisons with other methods that address the redundancy resolution and path planning problem, such as the hierarchical quadratic programming method proposed in (Tassi et al., 2021) and the two-stage approach proposed in (Zhou et al., 2023). In addition, the inherent flexibility of the RRT algorithm will allow our method to incorporate non-holonomic constraints, as well as dynamic constraints, since the time dimension is taken into account in the AFM space.

Additional future work should focus on extending the method to higher dimensionalities, as well as incorporating dynamic constraints and obstacles that move in time. Moreover, the method could be extended to consider every AFM that corresponds to different extended aspects. Finally, the method should be tested in real robotic systems to evaluate its performance in practical scenarios.

ACKNOWLEDGEMENTS

Work supported by project PID2020-116418RB-I00 and grant PRE2021-099226, funded by MCIN/AEI/10.13039/501100011033 and the ESF+.

REFERENCES

Albu-Schäffer, A. and Sachtler, A. (2023). Redundancy resolution at position level. *IEEE Transactions on Robotics*.

Borrel, P. and Liégeois, A. (1986). A study of multiple manipulator inverse kinematic solutions with applications to trajectory planning and workspace determination. In *Proceedings. 1986 ieee international conference on robotics and automation*, volume 3, pages 1180–1185. IEEE.

Burdick, J. W. (1989). On the inverse kinematics of redundant manipulators: Characterization of the self-motion manifolds. In *Advanced Robotics: 1989: Proceedings of the 4th International Conference on Advanced Robotics Columbus, Ohio, June 13–15, 1989*, pages 25–34. Springer.

Fabregat-Jaen, M., Peidro, A., Gil, A., Valiente, D., and Reinoso, O. (2023). Exploring feasibility maps for trajectory planning of redundant manipulators using rrt. In *2023 IEEE 28th International Conference on Emerging Technologies and Factory Automation (ETFA)*, pages 1–8. IEEE.

Ferrentino, E. and Chiacchio, P. (2020). On the optimal resolution of inverse kinematics for redundant manipulators using a topological analysis. *Journal of Mechanisms and Robotics*, 12(3):031002.

Karaman, S. and Frazzoli, E. (2011). Sampling-based algorithms for optimal motion planning. *The international journal of robotics research*, 30(7):846–894.

Kazemipour, A., Khatib, M., Al Khudir, K., Gaz, C., and De Luca, A. (2022). Kinematic control of redundant robots with online handling of variable generalized hard constraints. *IEEE Robotics and Automation Letters*, 7(4):9279–9286.

Pámanes G, J. A., Wenger, P., and Zapata D, J. L. (2002). Motion planning of redundant manipulators for specified trajectory tasks. *Advances in Robot Kinematics: Theory and Applications*, pages 203–212.

Pan, J., Chitta, S., Pan, J., Manocha, D., Mirabel, J., Carpentier, J., and Montaut, L. (2024). HPP-FCL - An extension of the Flexible Collision Library.

Peidro, A. and Haug, E. J. (2023). Obstacle avoidance in operational configuration space kinematic control of redundant serial manipulators. *Machines*, 12(1):10.

Peidro, A., Reinoso, O., Gil, A., Marín, J. M., and Paya, L. (2018). A method based on the vanishing of self-motion manifolds to determine the collision-free workspace of redundant robots. *Mechanism and Machine Theory*, 128:84–109.

Reveles, D., Wenger, P., et al. (2016). Trajectory planning of kinematically redundant parallel manipulators by using multiple working modes. *Mechanism and Machine Theory*, 98:216–230.

Tassi, F., De Momi, E., and Ajoudani, A. (2021). Augmented hierarchical quadratic programming for adaptive compliance robot control. In *2021 IEEE international conference on robotics and automation (ICRA)*, pages 3568–3574. IEEE.

Wenger, P., Chedmail, P., and Reynier, F. (1993). A global analysis of following trajectories by redundant manipulators in the presence of obstacles. In *[1993] Proceedings IEEE International Conference on Robotics and Automation*, pages 901–906. IEEE.

Whitney, D. E. (1969). Resolved motion rate control of manipulators and human prostheses. *IEEE Transactions on man-machine systems*, 10(2):47–53.

Zanchettin, A. M. and Rocco, P. (2017). Motion planning for robotic manipulators using robust constrained control. *Control Engineering Practice*, 59:127–136.

Zhou, Z., Zhao, J., Zhang, Z., and Li, X. (2023). Motion planning method of redundant dual-chain manipulator with multiple constraints. *Journal of Intelligent & Robotic Systems*, 108(4):69.