UNIVERSITAT POLITÈCNICA DE CATALUNYA

Ph.D. Program:

AUTOMATIC CONTROL, ROBOTICS AND COMPUTER VISION

Ph.D. Thesis:

MOTION PLANNING USING SYNERGIES:
APPLICATION TO ANTHROPOMORPHIC DUAL-ARM ROBOTS

Néstor García Hidalgo

Thesis advisors: Raúl Suárez Feijóo
Jan Rosell Gratacòs

July 2019

MOTION PLANNING USING SYNERGIES: APPLICATION TO ANTHROPOMORPHIC DUAL-ARM ROBOTS

Néstor García Hidalgo

Abstract

Motion planning is a traditional field in robotics, but new problems are nevertheless incessantly appearing due to continuous advances in the robot developments. In order to solve these new problems, as well as to improve the existing solutions to classical problems, new approaches are being proposed. A paradigmatic case is the humanoid robotics, since the advances done in this field require motion planners not only to look efficiently for an optimal solution in the classical way, i.e. optimizing consumed energy or time in the plan execution, but also looking for human-like solutions, i.e. requiring the robot movements to be similar to those of the human beings. This anthropomorphism in the robot motion is desired not only for aesthetical reasons, but it is also needed to allow a better and safer human-robot collaboration. Humans can predict more easily anthropomorphic robot motions, which makes them feel more comfortable and helps in avoiding collision, enhancing therefore the human-robot collaboration. Nevertheless, obtaining a satisfactory performance of these anthropomorphic robotic systems is still an arduous and non-evident task since the complexity of the planning problem increases exponentially with the number of degrees of freedom of the robotic system.

This doctoral thesis tackles the problem of planning the motions of robots with a dual-arm anthropomorphic torso and, optionally, with mobile base. The main objective is twofold: obtaining robot motions both in an efficient and in a human-like fashion at the same time. Trying to mimic the human movements while simultaneously reducing the complexity of the search space for planning purposes leads to the concept of synergies. The synergies can be conceptually defined as correlations between the degrees of freedom of the system and, in this work, synergies in the joint configuration space as well as in the joint velocity space are considered. This work provides a practical use of the synergies existing in the human dual-arm movements for the motion planning of human-like dual-arm robotic systems, integrating the dimensionality reduction inherent to the synergies into classical planning techniques. The proposed sampling-based motion-planning procedures exploit the concept of synergies both in the configuration and velocity space, and are able to tackle the planning of the movements of the arms, the hands and the mobile base of mobile anthropomorphic dual-arm robots. The proposed approach is tested in simulated and real scenarios as well as benchmarked against other state-of-the-art procedures, obtaining good results.

Keywords: *Motion planning, Synergies, Dimensionality reduction, Human-Like movements, Dual-Arm manipulation, Mobile manipulators.*

MOTION PLANNING USING SYNERGIES:
APPLICATION TO ANTHROPOMORPHIC DUAL-ARM ROBOTS

Néstor García Hidalgo

Resumen

La planificación de movimientos es un campo tradicional de la robótica, sin embargo aparecen incesantemente nuevos problemas debido a los continuos avances en el desarrollo de los robots. Para resolver esos nuevos problemas, así como para mejorar las soluciones existentes a los problemas clásicos, se están proponiendo nuevos enfoques. Un caso paradigmático es la robótica humanoide, ya que los avances realizados en este campo requieren que los algoritmos planificadores de movimientos no sólo encuentren eficientemente una solución óptima en el sentido clásico, es decir, optimizar el consumo de energía o el tiempo de ejecución de la trayectoria; sino que también busquen soluciones con apariencia humana, es decir, que el movimiento del robot sea similar al del ser humano. Este antropomorfismo en el movimiento del robot se busca no sólo por razones estéticas, sino porque también es necesario para permitir una colaboración mejor y más segura entre el robot y el operario. Los seres humanos pueden predecir con mayor facilidad los movimientos del robot si éstos son antropomórficos, lo que les hace sentir más cómodos, y facilita la colaboración humano-robot. Sin embargo, obtener un desempeño satisfactorio de estos sistemas robóticos antropomórficos sigue siendo una tarea ardua y poco evidente, ya que la complejidad del problema aumenta exponencialmente con el número de grados de libertad del sistema robótico.

Esta tesis doctoral aborda el problema de la planificación de movimientos en robots con un torso antropomorfo bibrazo y, opcionalmente, con base móvil. El objetivo aquí es doble: obtener movimientos robóticos de forma eficiente y, a la vez, que tengan apariencia humana. Intentar imitar los movimientos humanos mientras a la vez se reduce la complejidad del espacio de búsqueda conduce al concepto de sinergias. Las sinergias pueden definirse conceptualmente como correlaciones entre los distintos grados de libertad del sistema y, en este trabajo, se consideran sinergias tanto en el espacio de configuraciones como en el espacio de velocidades de las articulaciones. Este trabajo propone nuevos procedimientos de planificación de movimientos que explotan el concepto de sinergias, tanto en el espacio de configuraciones como en el espacio de velocidades. Este trabajo proporciona un uso práctico de las sinergias existentes en los movimientos humanos bimanuales para la planificación de movimientos de sistemas robóticos antropomórficos bibrazo, integrando reducción inherente de la dimensionalidad de las sinergias en las técnicas de planificación clásicas. Los procedimientos aquí propuestos para la planifiación de movimientos explotan el concepto de las sinergias, tanto en el espacio de configuraciones como en el espacio de velocidades, y son capaces de abordar la planificación de los movimientos de los brazos, las manos y la base móvil de los robots móviles bibrazo antropomórficos. El enfoque propuesto ha sido probado en escenarios simulados y reales así como comparado con otros procedimientos, obteniendo buenos resultados.

Palabras clave: *Planificación de movimientos, Sinergias, Reducción de dimensión, Movimientos con apariencia humana, Manipulación bibrazo, Manipuladores móviles.*

Acknowledgements

I would like to thank everyone who has been with me while this work was being developed. Without them, this would not have been possible.

Foremost, I would like to show my appreciation and gratitude to my two supervisors for participating in this project and for all the time they have lost with me. Their patience, continuous support and dedication have been my motivation during my doctorate.

I would like to thank Leo for all his advice, constant feedback, criticism and for being there every time I needed his help. Thank you also for sharing your experience in our never-ending conversations.

I extend my gratitude to all my colleagues and fellow PhD students in the Institute of Industrial and Control Engineering for their friendly laughs and great discussions.

To my long-life friends, for their support all these years. Thank you for suffering my melancholies and celebrating my joys, and definitely making my life easier.

Besides, I owe my sincere appreciation to my family, my parents and my beloved little brother, for their generosity and patience. Thanks for always believing in me and encouraging me to go further. I dedicate this work to them.

I would not want to forget to mention my partner. I know you have come to witness the end of this long adventure, but not for that reason the easiest part or with fewer adversities. Your caring support and the conversations with you have been the last push I needed to make this happen. Thank you, sweetheart.

Thanks to the Department of Mechanical Engineering for allowing me to use the Optitrack motion-capture system.

Finally, I thank the Generalitat de Catalunya for the scholarship FI-DGR I was given and thanks to the Spanish Government for partially supporting this work trough the CICYT projects DPI2011-22471, DPI2013 40882-P, DPI 2014-57757-R and DPI2016-80077-R.

Néstor García Hidalgo
Barcelona, Spain
July 17, 2019

Contents

List of Figures

List of Tables

Notation and Acronyms

Notation

$\forall$	For all
$\Rightarrow$	Implies that
$\rightarrow$	Tends to
$\propto$	Proportional to
$\in$	Belongs to
$\subset$	Subset of
$\cup$	Set union
$\cap$	Set intersection
$\setminus$	Set difference
$\sim$	Distributed as
$\|\cdot\|$	Euclidean norm of a vector
$\|\cdot\|_F$	Frobenius norm of a matrix
$A^\intercal$	Transpose of a given matrix A
$\mathrm{erf}(\cdot)$	Error function
$\mathrm{E}[\cdot]$	Expected-value function
$\mathbb{R}^n$	n-dimensional Euclidean space
$SO(n)$	Group of positive rotations in $\mathbb{R}^n$
$\boldsymbol{x} \in \mathbb{R}^n$	Vector, i.e. $\boldsymbol{x} = [x_1, \dots, x_n]^\intercal$
$\hat{\boldsymbol{u}}$	Unit vector
$\boldsymbol{p} \in \mathbb{R}^3$	3D point, expressed in a given world reference system
$R \in SO(3)$	Rotation matrix in $\mathbb{R}^3$
$I \in \mathbb{R}^{n\times n}$	n-dimensional Identity matrix

$\mathcal{C} \in \mathbb{R}^n$	Configuration space of a robot with n DOFs
$\boldsymbol{q} \in \mathcal{C}$	Robot configuration
Q	Set of configurations $\boldsymbol{q}$
$\mathcal{C}_{\text{free}} \subseteq \mathcal{C}$	Subspace of the configurations $\boldsymbol{q} \in \mathcal{C}$ in which the robot is not in collision (neither with itself nor with the environment)
$\mathcal{C}_{\text{obs}} \subseteq \mathcal{C}$	Subspace of the configurations $\boldsymbol{q} \in \mathcal{C}$ in which the robot is in collision
$\boldsymbol{\chi} \in \mathbb{R}^2$	Position (x, y) of the robot base, expressed in a given world reference system
G	Object grasping, defined by hand configuration $\boldsymbol{q}_{\text{hand}}$ and pose ${}^{\text{hand}}T_{\text{object}}$ of the object with respect to hand frame at grasping, i.e. $G = (\boldsymbol{q}_{\text{hand}}, {}^{\text{hand}}T_{\text{object}})$
$\mathcal{P}$	Geometric robot path
$\mathcal{T}$	Configuration tree
$\mathcal{G}$	Grid
$c_{\mathcal{G}}$	Grid cell
S	Sample
$V(\boldsymbol{q})$	Potential field value at a given configuration $\boldsymbol{q}$
$\boldsymbol{f}(\boldsymbol{q})$	Vector field value at a given configuration $\boldsymbol{q}$
$\boldsymbol{\mu}$	Barycenter
Σ	Covariance matrix
$\boldsymbol{\sigma}$	Singular values of the covariance matrix Σ
U	Matrix of eigenvectors of the covariance matrix Σ, i.e. $U = [\boldsymbol{u_1}, \dots, \boldsymbol{u_n}]$
$\mathcal{N}(\boldsymbol{\mu}, \Sigma)$	Multivariate normal distribution with barycenter $\boldsymbol{\mu}$ and covariance matrix Σ
${}^0\mathcal{S}$	Basis of zero-order synergies
${}^1\mathcal{S}$	Basis of first-order synergies
α	Sample variance not covered by reduced synergy basis
β	Sample variance not covered by reduced synergy box
B	Synergy box enclosing a given synergy basis
B_k	Lower-dimensional box enclosing a given synergy basis
$\mathcal{T}_{KD}$	k-d tree of synergies

$c_{\mathcal{P}}$	Path cost
θ	Joint value of a rotational joint
$\mathcal{L}$	Task likeness
$\mathcal{H}$	Human-likeness
$\mathcal{U}$	Upstream criterion
$E_i(\boldsymbol{q}) \geq 0$	Wrist tracking error of the i-th robotic arm, evaluated at the configuration $\boldsymbol{q}$
$A_i(\boldsymbol{q}) \geq 0$	Anthropomorphic dissimilarity of the i-th robotic arm, evaluated at the configuration $\boldsymbol{q}$
E	Set of edges in a graph
V	Set of vertices in a graph

Latin abbreviations

e.g.	-exempli gratia-	for instance
et al.	-et alii-	and others
etc.	-et caetera-	and the rest
i.e.	-id est-	that is
Q.E.D.	-quod erat demonstrandum-	which was to be demonstrated
vs.	-versus-	against

English abbreviations

s.t.	such that
w.r.t.	with respect to

Acronyms

API	Application programming interface
DOF	Degree of freedom
EMG	Electromyogram
FK	Forwards kinematics
FOS	First-Order synergies
FOS-BKPIECE	FOS-Based bidirectional KPIECE
GUI	Graphical user interface
HD-RRT*	Human-Demonstrated RRT*
HMM	Hidden Markov Model
IK	Inverse kinematics
KPIECE	Kinodynamic planning by interior-exterior cell exploration
MADAR	Mobile anthropomorphic dual-arm robot
PCA	Principal component analysis
PMD	Principal motion direction
PID	Proportional-Integral-Derivative
PRM	Probabilistic roadmap
ROS	Robot operating system
RGB-D	RGB color (i.e. red, green and blue) and depth information
RRT	Rapidly-Exploring random trees
SBP	Sampling-Based planner
SyCLoP	Synergistic Combination of Layers of Planning
T-RRT	Transition-Based RRT
VF-RRT	Vector-Field RRT
ZOS	Zero-Order synergies

1

Introduction

The work presented in this thesis deals with the problem of planning the motions of dual-arm anthropomorphic robots in an efficient and human-like fashion, by means of the concept of synergies, which could be conceptually defined as correlations (in the joint configuration space as well as in the joint velocity space) between the degrees of freedom of the system. This chapter presents the treated problem and introduces some basic definitions of concepts used along the document. It also gives a brief description of the experimental setup as well as of the software tools used in the validation of the proposed approach. Furthermore, the thesis objectives are defined and a brief outline of the dissertation is presented.

1.1 Context and motivation

Motion planning is a vital research issue in robotics (Latombe, 1991), particularly since the robots became an essential part in many application fields like, for instance, the medical and the electronic industries, and even the computational biology or the computer animation. The importance of this problem is manifested when the planning of the motions of a robotic system with a high number of degrees of freedom (DOFs) is attempted, like those involving mechanical hands or anthropomorphic systems. Usually, a planned path between two given configurations of a mobile system is only required to be feasible, i.e. satisfying intrinsic constraints of the system (e.g. kinematic or mechanical design constraints) and constraints arising from the environment (e.g. collision avoidance). Nevertheless, it is becoming more and more common to require that the planned paths optimize some path quality metric (e.g. minimizing the execution time of the path or maximizing the path clearance).

In the last years, our society has been looking at robots as future helpers for humans: they could assist elderly people, cooperate with us in the execution of tedious and hard works, and even replace us for the accomplishment of the most dangerous tasks. This long term goal has been pushing the robotics researchers to build more and more complex robots with an increasing number of degrees of freedom and sensors. This has caused that the number of anthropomorphic dual-arm systems available in the market has significantly grown in the recent years, either as a specific product (Agravante et al., 2013; Diftler et al., 2011; Bohren et al., 2011); or as a composition of manipulator arms assembled imitating a human structure (Srinivasa et al., 2012; Maldonado et al., 2012; Albu-Schöffer et al., 2007; Asfour et al., 2006b), that can be installed on a mobile platform (Suárez et al., 2018) or fixed in the environment (Smith et al., 2012). These dual-arm devices typically have around 15 DOFs, which is a significant number when motion planning has to be done, and becomes even more relevant when the dual-arm system is equipped with dexterous mechanical hands or when it is part of a whole humanoid robot (Englsberger et al., 2014; Kaneko et al., 2008; Sakagami et al., 2002). In the latter case we face robot systems that could have up to nearly 60 DOFs. A typical problem with the humanoid robots is that the motion planning should not only focus on the efficient search for an optimal solution in the traditional sense (minimization of the energy or time required in the path execution), but also in the search of robot movements that mimic the natural movements of the human beings.

On the one hand, as robots are increasingly used in environments where humans are also present, the requirement of human-likeness in their movements becomes more relevant. Although robots perform a wide variety of demanding tasks in factories around the world, remote sites and dangerous environments, they are still lacking the ability to coordinate with humans in simple everyday tasks that involve using both hands, e.g. removing a jar lid. It has been long discussed that human-robot collaboration is facilitated if the robot has an anthropomorphic shape and performs human-like movements (Fukuda et al., 2001; Duffy, 2003; Schaal, 2007; Schaal et al., 2007). These characteristics support natural and efficient human-robot interaction since they allow the human user to understand more easily the movements of the robot as goal directed actions (Bicho et al., 2011a,b). On the other hand, looking for human-like movements leads to the search of the proper coordination between the robot joint movements. In the motion planning of mechanical hands, several works used synergies (i.e. correlations between DOFs) with good results to simplify the problem by reducing the dimension of the search space as well as to mimic human postures (Ciocarlie and Allen, 2009; Rosell et al., 2011; Sun et al., 2010). Nevertheless, its application to an anthropomorphic dual-arm systems had not been reported yet. Hence, the results of these pioneering works encourage the author to extend a generalization of the procedures used with the mechanical hands to human-like dual-arm robot systems.

Therefore, the main motivation behind this study comes from the fact that when humans are able to predict the intent of robot motions, they may adjust their motion to avoid possible injuries or enhance collaborations. Thus, the study of the synergies and its use in dual-arm motion planning to derive anthropomorphic robot motions can be significant not only for aesthetical but also for practical reasons.

1.2 Objectives

The problem to be solved is the coordinated motion planning of an anthropomorphic dual-arm system equipped with dexterous mechanical hands and mounted on a mobile base. For this, the existence of synergies in the human dual-arm system will be investigated and used to reduce the computational cost of the planning process and derive human-like robot movements. This approach will be considered in the motion planning of:

(a) Both arms performing coordinately bimanual tasks.

(b) The arms and the mobile base performing coordinately the approaching motions previous to a bimanual task.

(c) An arm and its hand performing coordinately the grasping motions previous to a manipulation task.

Towards this objective, several more specific objectives are pursued. These objectives are summarized in the following list:

- Development of a general procedure for planning motions exploiting the concept of postural synergies and extending it to the velocity space.
- Development of path planning algorithms that use synergies and that are specifically designed for anthropomorphic dual-arm robot systems.
- Development of a software architecture to implement the above developments, addressing important software design issues such as efficiency, robustness, modularity, scalability, and portability.
- Study of the dual-arm synergies while executing different tasks.
- Classification of dual-arm tasks according to the used synergies and employ them to improve the motion planning of the dual-arm system.
- Execution of simulated and real experiments that demonstrate the applicability of the proposed techniques.

1.3 Hardware

The main hardware involved in this work consists of:

(a) A mobile anthropomorphic dual-arm robot used as a testbed to validate the proposed approach, testing the performance of the developed planners and comparing them to other state-of-the-art motion-planning algorithms.

(b) A set of motion-capture devices used to record the movements of the arms and hands, as well as the translation motions, of different human operators in order to obtain the human synergies.

The following subsections give a brief discussion of the features of each system.

1.3.1 Mobile Anthropomorphic Dual-Arm Robot

The motion-planning algorithms proposed in this doctoral thesis have been tested in different real cases using a general-purpose dexterous dual-arm mobile manipulator named MADAR (from Mobile Anthropomorphic Dual-Arm Robot), shown in Fig. 1.1 (Suárez et al., 2018). This mobile manipulator is composed of a dual-arm torso with a human-like structure assembled on an omnidirectional platform.

The omnidirectional platform is circular with a diameter of 72 cm that allows the mobile robot to pass through standard doors and elevators. The platform has three traction groups, composed each one of a wheel and a motor, located under the base and disposed in radial directions at 120 degrees of each other, see Fig. 1.2. The wheels have an original special design that ensures a permanent contact between each wheel and the floor without needing any suspension systems or articulated frames. Moreover, the three degrees of freedom of the platform can be controlled by means of the three independent wheel rotations, simplifying the platform control and allowing omnidirectional displacements of the platform.

The mobile robot is equipped with two industrial robotic arms UR5 from Universal Robots[1], see Fig. 1.3. These arms have 6 DOFs, can hold up to 5 kg, and have collision-detection capabilities that allow a safe human-robot interaction. The arms are attached to a central metal structure with T-shape, on top of the mobile base. This design gives to the robot a human-like height with a larger wingspan, allowing the robot to work in human environments, for instance, operating on a table or picking items up from a typical bookshelf and move them around. The heavy controller boxes of the arms have been mounted trying to keep the center of gravity of the robot as low as possible for stability reasons. The assembling of the arms emulates the human-arm configuration and, at the same time, it allows the arms to fit within the circular area of the base, when they are folded. The mounting orientation of the arms and the inter-arm distance have been chosen to obtain a suitable overlapping of the two arm-workspaces, as well as a good manipulability value (Kurazume and Hasegawa, 2006) in front of the robot, where it is expected to perform most bimanual manipulations.

---

[1] www.universal-robots.com/products/ur5-robot

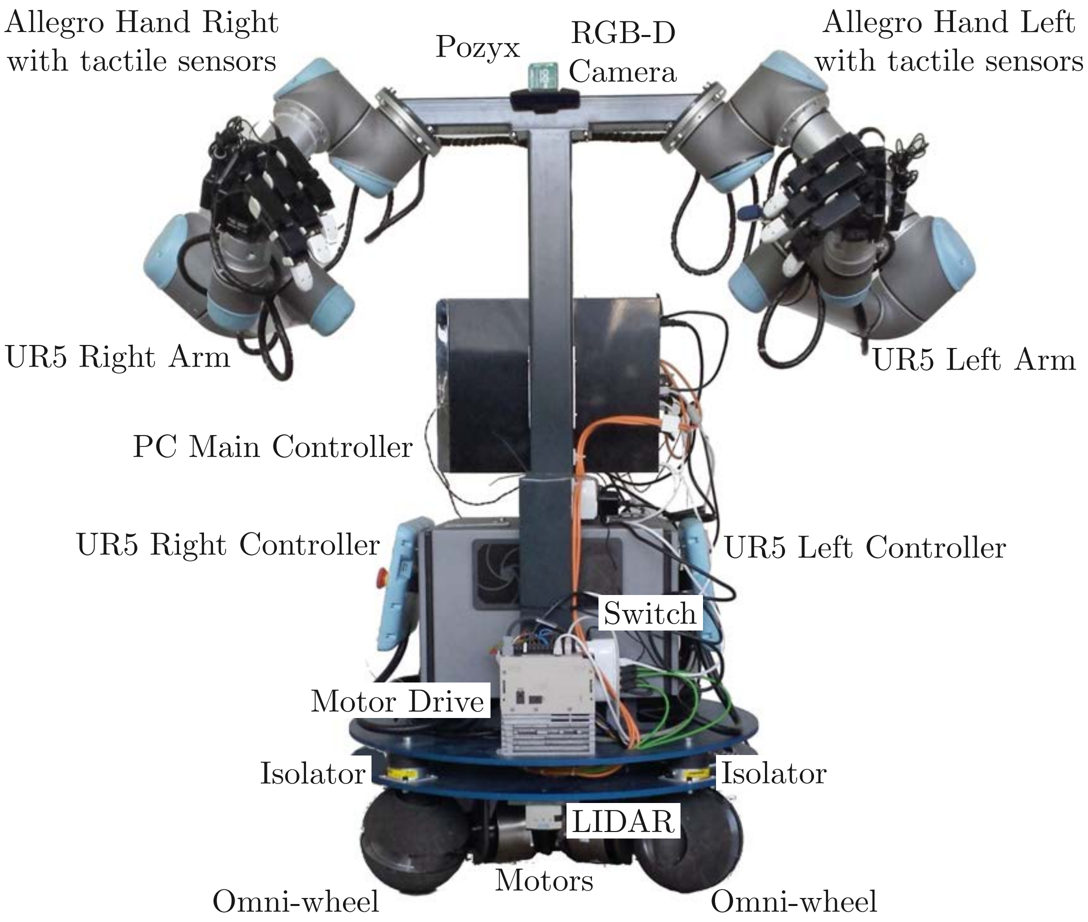


Figure 1.1: Front view of the MADAR robot.

Each arm is equipped with an Allegro Hand from Simlab[2] as the end-effector, see Fig. 1.3. The Allegro Hand has four fingers and sixteen independent torque-controlled joints (4 joints per finger). This hand weights 1.5 kg and it is able to hold up to 5 kg. Each fingertip is provided with a tactile sensor WTS-FT 0408 from Weiss[3]. Each sensor has a sensing matrix with 8×4 cells and includes the complete signal processing electronics.

In addition, the robot is also equipped with three laser-range sensors (TIM-561 LIDAR sensors, from Sick[4]), a radio positioning-system (Pozyx[5]) and an RGB-D camera to detect potential obstacles and navigate safely. All the devices are connected to and controlled by an on-board central PC. Besides, the robot can also be remotely controlled using a 5 GHz wireless connection.

---

[2] www.simlab.co.kr/Allegro-Hand.htm

[3] www.weiss-robotics.com/en/produkte/tactile-sensing/wts-ft-en

[4] www.sick.com/de/en/detection-and-ranging-solutions/2d-lidar-sensors/tim5xx/tim561-2050101/p/p369446

[5] www.pozyx.io

Figure 1.2: Details of the three groups wheel-motor (left) and details of the omni-wheels developed for the mobile base (right).

Figure 1.3: Details of the left robotic arm and hand.

1.3.2 Motion-Capture devices

A set of motion capture devices have been used to capture the movements of a human operator performing bimanual task, and used to obtain dual-arm human synergies that are later used to accelerate the motion planning for a dual-arm robot.

Two sensorized gloves CyberGlove from CyberGlove Systems[6] have been used to record the configurations of the human hands, see Fig. 1.4-left. Each glove is fully instrumented and provides 22 high-accuracy joint-angle measurements using resistive bend-sensing technology. The gloves are equipped with three flexion sensors per finger, four abduction sensors between the

---

[6] www.cyberglovesystems.com/cyberglove-ii

Figure 1.4: Motion-capture devices used in this work: Sensorized gloves (left), magnetic trackers (middle), and optical trackers system (right).

fingers, two sensors to measure the flexion and the abduction of the wrist, and a palm-arch sensor (i.e. rotation of the little finger across the palm towards the thumb). These sensors present a linear behavior and, once calibrated, have a resolution of 0.5 degrees and its repeatability is of 3 degrees (typical standard deviation between glove wearings). The gloves can transmit data at 90 Hz and, besides, they provide in the wristband a mounting provision for a motion tracking sensor.

The 6 DOF motion tracker Fastrak manufactured by Polhemus[7] has been used to capture the configurations of the arms of the human operator in those experiments in which he/she is asked not to change its location (see Fig. 1.4-middle). This device consists mainly of a transmitter, emitting a pulsed magnetic field, and sensors, perceiving this magnetic field. From the characteristics of the measured magnetic field, each sensor independently computes its position and orientation, relative to the transmitter. The tracking system is sensitive to the electromagnetic environment. Therefore, the sensors and the transmitter must be used far from other electromagnetic fields and metallic elements. Otherwise, the transmitter signals are distorted and the resulting measurements are not accurate. The sensors transmit data at 120 Hz and have a resolution of 8 mm in position and 0.15 degrees in orientation, when they are within a range of 0.75 m of the transmitter. In this work, a sensor have been placed on each wrist and each elbow of the human operator. In this way, if he/she remains in a known position without moving his/her back during the whole experiment (i.e. the shoulder positions are known), the complete arm configurations can be recovered from the sensor measurements.

---

[7] polhemus.com/motion-tracking/all-trackers/fastrak

For those experiments in which the human synergies have been obtained while the human operator changes his/her position, an optical motion-capture system formed by reflective tracking markers and 16 infrared Flex 3 cameras from OptiTrack[8] has been used (see Fig, 1.4-right). This system allows a 3D localization of the markers with sub-millimeter accuracy and with a rate of 100 Hz, after a post-processing of the recorded signals. A total of 12 optical markers have been placed on the human operator to capture his/her position and the complete arm configurations: a marker on each shoulder and each elbow, another marker at each waist side, and three markers on each palm (in order to record both the position and orientation of the hand).

1.4 Software

Several software tools have been used in this work in order to capture and analyze human movements, as well as to plan the motions of a dual-arm anthropomorphic robot, validating thereby the proposed approach. This section describes briefly the most important used software tools.

1.4.1 The Kautham Project

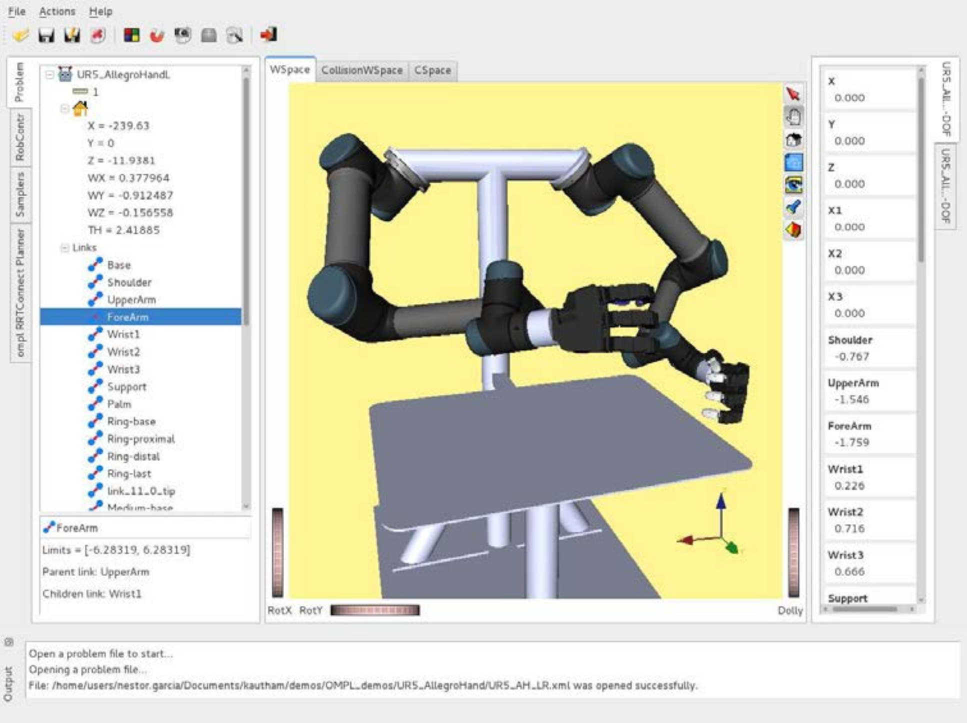
The motion-planning algorithms proposed in this doctoral thesis have been implemented within The Kautham Project (Rosell et al., 2014), a motion-planning and simulation environment developed at the IOC-UPC[9] for teaching and research purposes. This simulation environment includes tools for collision detection, motion planning and graphical visualization of the whole system (see a snapshot of the GUI in Fig. 1.5). This software package, developed in C++ with the open-source and cross-platform directives in mind, is available at sir.upc.edu/kautham. Besides, The Kautham Project can be flexibly configured (e.g. planner type, planner parameters, query to be solved, etc.) from a task level perspective, and the obtained results can be used as inputs for further task reasoning.

The Kautham Project allows to cope with problems with one or more robots, which may range from simple free-flying robots to mobile manipulators equipped with mechanical hands,

---

[8] optitrack.com/products/flex-3

[9] robotics.upc.edu



UR5_AllegroHandL
1
X = -239.63
Y = 0
Z = -11.9381
WX = 0.377964
WY = -0.912487
WZ = -0.156558
TH = 2.41885
Links
Base
Shoulder
UpperArm
ForeArm
Wrist1
Wrist2
Wrist3
Support
Palm
Ring-base
Ring-proximal
Ring-distal
Ring-last
link_11_0_tip
Medium-base

ForeArm
Limits = [-6.28319, 6.28319]
Parent link: UpperArm
Children link: Wrist1

X 0.000
Y 0.000
Z 0.000
X1 0.000
X2 0.000
X3 0.000
Shoulder -0.767
UpperArm -1.546
ForeArm -1.759
Wrist1 0.226
Wrist2 0.716
Wrist3 0.666
Support

```
Open a problem file to start...
Opening a problem file...
File: /home/users/nestor.garcia/Documents/kautham/demos/OMPL_demos/UR5_AllegroHand/UR5_AH_LR.xml was opened successfully.
```

Figure 1.5: Graphical user interface of The Kautham Project.

see Fig. 1.6. Based on the Open Motion Planning Library suite of planners, OMPL (Suçan et al., 2012), it offers some advanced features like the easy parametrization of the planners, an easy way to define coupled degrees of freedom and tailor the planning accordingly, the dynamic simulation, the integration with task planners. In addition, a console application is provided to run in batch mode and execute different benchmarkings. It facilitates solving a motion-planning problem repeatedly with different planners, different samplers, or even differently configured versions of the same planning algorithm. It also allows the user to run the same benchmarking but with different number of repetitions or subject to different time and memory limits.

Among the advanced features, this environment allows to easily fix DOFs and to define DOF couplings within the motion-planning phase. Thereby, certain couplings could be set to obtain a certain appearance of the planned robot-motions, which is very appealing for this work. For instance, the joints of a mechanical hand could be coupled to mimic the observed couplings that there exist in the human hand motions. In addition, since the motion planning is performed in the space defined by the couplings, the dimension of the problem can be significantly reduced

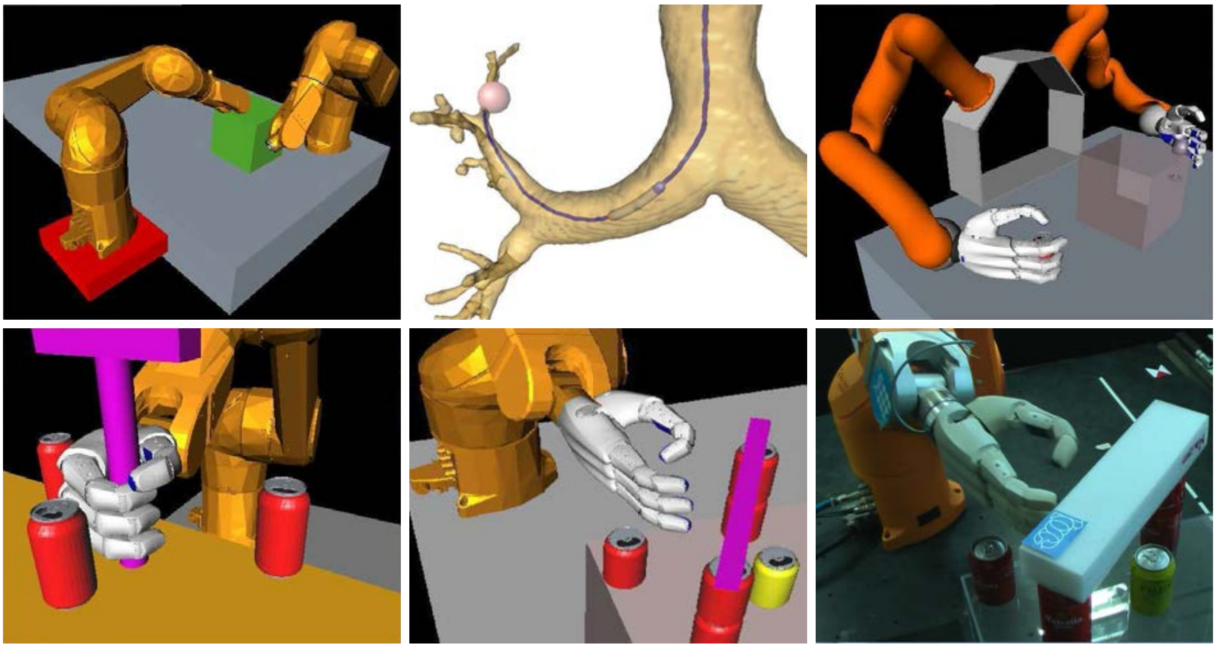


Figure 1.6: Examples of complex planning problems managed with The Kautham Project (from left to right, and top to bottom): cooperation in an industrial robotic cell with a fixed and a mobile manipulator, motion of a bronchoscope modeled as a robot with kinematics constrains, coordination of a dual-arm torso and grasping in cluttered environments.

when using less DOF couplings than the total number of DOFs. For example, in the case of the mechanical hand that mimics the human hand, the consideration of up to 5 couplings (i.e. synergies) is enough to cover a high percentage of the hand motions that a human comfortably performs (Rosell et al., 2011).

1.4.2 Robot Operating System

Robot Operating System, ROS (Quigley et al., 2009), is an open-source flexible framework that aims to simplify the development of general-purpose collaborative software for robots. In addition, ROS encourages their users to contribute to the community with new software packages. Besides, it eases the collaboration between different research institutions by developing new software based on each other's work. Although ROS is not an operating system, it provides services designed for a heterogeneous computer cluster such as hardware abstraction, low-level device control, implementation of commonly used functionality, message-passing between processes, and package management. In the ROS framework, the processing takes place in nodes

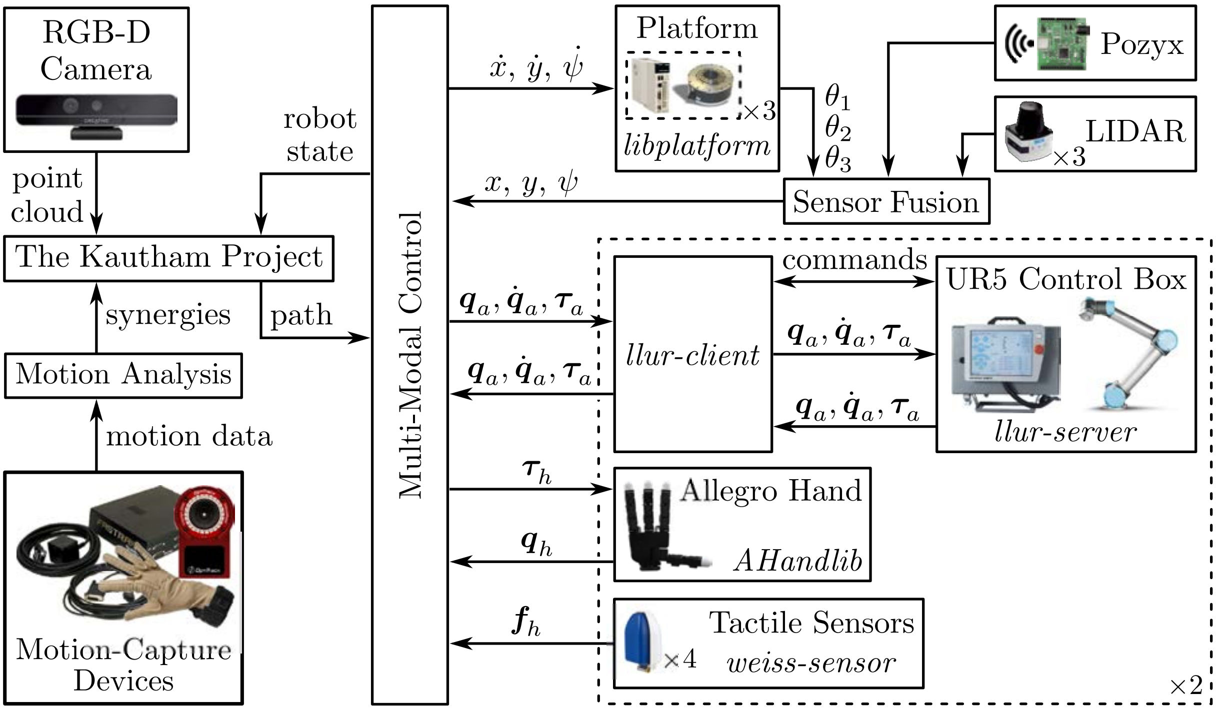


Figure 1.7: General scheme of the different software modules developed for this work, as well as the ROS connections between them.

that are seamlessly distributed, in the same or in different cores and machines. These nodes, coordinated by a master node, send data streams to each other and can be dynamically configured. Although originally oriented to UNIX-like systems (e.g. Ubuntu), it is also adapted to work in other operating systems (e.g. macOS and Microsoft Windows).

ROS has been the framework used to develop and to connect all the software modules designed for this work, see Fig. 1.7. The Kautham Project has been encapsulated within a ROS service and it is used to generate collision-free robot trajectories based on the environment information provided by the RGB-D camera mounted on the robot and using the current robot state as start configuration. Human-like robot motions are obtained by using the synergies obtained analyzing some captured human motions. The planned trajectory is sent to a multi-modal control layer that continuously reads the robot state and properly commands the wheels, the arms and the hands. This layer controls the robot platform with Cartesian velocity commands $(\dot{x}, \dot{y}, \dot{\psi})$, obtaining Cartesian position feedback (x, y, ψ). An accurate robot navigation is obtained by fusing the information provided by the wheels encoders θ_i, the radio positioning-system Pozyx and the LIDAR sensors (using the library called *libplatform*) . The arms are commanded by joint position $\boldsymbol{q}_a$, velocity $\dot{\boldsymbol{q}}_a$ and torque $\boldsymbol{\tau}_a$, obtaining feedback from all these measures (using the *llur-client* and *llur-server* libraries of the developed low-level module, called Low Level Universal Robot Control, LLUR-Control). The hands are commanded by joint torque $\boldsymbol{\tau}_h$, obtaining feedback of joint position $\boldsymbol{q}_h$ and fingertip force distribution $\boldsymbol{f}_h$ (using the libraries called *AHandlib* and *weiss-sensor*).

1.5 Outline of the thesis

The presented doctoral thesis is organized in seven chapters. A brief description of their content is the following:

Chapter 1 describes the motivation and objectives of the present work and gives an insight of the basic concepts used along the document. In addition, it gives a brief description of the experimental setup as well as the software tools used in the validation of the proposed approach.

Chapter 2 discusses the state of the art in motion-planning algorithms, and especially techniques to plan human-like motions. The main developments of this chapter have been published in MIC (García et al., 2017b).

Chapter 3 presents a proper definition of the synergies and proposes their use to characterize tasks and robot motions. The main developments of this chapter have been published in ICRA (Suárez et al., 2015), IROS (García et al., 2015a), T-SMC (García et al., 2017a) and T-RO (García et al., 2017d).

Chapter 4 presents different approaches to plan the motions of a dual-arm robot performing coordinately bimanual tasks. The proposed planning procedures are detailed and compared with other state-of-the-art planners in conceptual and application examples to demonstrate the applicability of the proposed motion-planning techniques, both in simulated and in real executions. The main developments of this chapter have been published in ICRA (Suárez et al., 2015), IROS (García et al., 2015a), ETFA (García et al., 2015b), T-SMC (García et al., 2017a), IFAC World Congress (García et al., 2017c) and also T-RO (García et al., 2017d).

Chapter 5 extends the synergy-based planning procedures presented in the previous chapter to include also the hands and the mobile base in the motion planning. A detailed description of the approach is included, as well as experimental results. The main developments of this chapter have been published in SIMPAR (García et al., 2018a) and IROS (García et al., 2018b).

Chapter 6 summarizes the thesis contributions and states some concluding remarks. In addition, it lists the derived publications and discusses the potential research lines opened by this work.

2

State of the Art

This chapter collects the most relevant works found in the motion-planning literature. The first section outlines the most outstanding motion-planning techniques and two more sections are dedicated to review some interesting works related to the efficient motion-planning for highly-articulated robotic systems and the planning of motions with human-like appearance.

2.1 Motion-planning techniques

For more than three decades in the study of robotics, there has been great interest in the development of motion-planning algorithms (Latombe, 1991). To define the motion-planning problem, some concepts must be introduced. Consider a robotic system, whose movements need to be planned, with d degrees of freedom (DOF). Note that robotic system can refer to a mobile robot, a robotic arm and even a multi-robot system. The configuration $\boldsymbol{q} \in \mathbb{R}^d$ represents unequivocally a kinematic state of the robotic system. Then, the space defined by the values $\boldsymbol{q}$ is the configuration space $\mathcal{C} \subseteq \mathbb{R}^d$. Let $\mathcal{C}_{\text{free}} \subseteq \mathcal{C}$ be the subspace of collision-free configurations, i.e. without autocollisions or collisions between robots or between robots and the environment. And let $\mathcal{C}_{\text{obs}} \subseteq \mathcal{C}$ be the subspace of configurations in collision, i.e. $\mathcal{C}_{\text{obs}} = \mathcal{C} \setminus \mathcal{C}_{\text{free}}$. The configuration space $\mathcal{C}$ usually does not match with the Cartesian Space. A gantry crane or a gyroscope are examples where this perfect matching occurs.

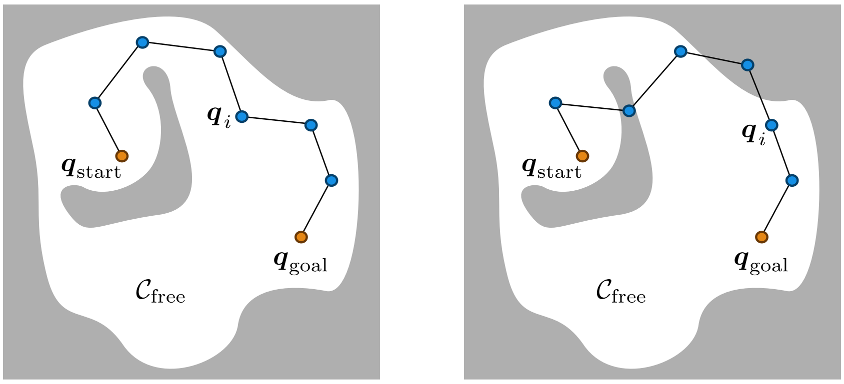


Figure 2.1: Example of a motion-planning problem: valid (left) and invalid (right) paths $\mathcal{P}$ depicted in the robot configuration space $\mathcal{C}$.

Let $\mathcal{P}$ be a path represented as a sequence of N configurations $\boldsymbol{q}_i$ consecutively connected by $N-1$ rectilinear segments, i.e.

$$\mathcal{P} = \bigcup_{i=1}^{N-1} \textsc{Segment}(\boldsymbol{q}_i, \boldsymbol{q}_{i+1}) \tag{2.1}$$

Then, given an initial configuration $\boldsymbol{q}_{\text{start}} \in \mathcal{C}_{\text{free}}$ and a final configuration $\boldsymbol{q}_{\text{goal}} \in \mathcal{C}_{\text{free}}$, the motion-planning problem consists in finding a valid path $\mathcal{P}$ that takes the robotic system from $\boldsymbol{q}_{\text{start}}$ to $\boldsymbol{q}_{\text{goal}}$ (i.e. $\boldsymbol{q}_1 = \boldsymbol{q}_{\text{start}}$ and $\boldsymbol{q}_N = \boldsymbol{q}_{\text{goal}}$) so that all the configurations defining the path, and the segments connecting them, belong to $\mathcal{C}_{\text{free}}$ (see Fig. 2.1). In addition, there exist some variants of this basic problem, such as considering differential constraints (e.g. motion planning of a car-like robot), considering uncertainty (e.g. the obstacle localizations are not known with precision) or looking for a valid path optimal w.r.t. some given path cost $c_{\mathcal{P}}$ (e.g. minimizing the time duration or the energy consumption of the path execution).

Motion planning is considered as a major problem in robotics since its complexity increases exponentially with the dimension of the configuration space. There are several techniques for path planning and, considering the most relevant ones, the following classification can be considered:

a) Classical methods that either find a solution or prove that a solution does not exist. The main disadvantage of these methods is the need of explicitly constructing the configuration space (which precludes them to be used in high-dimensional problems), their com-

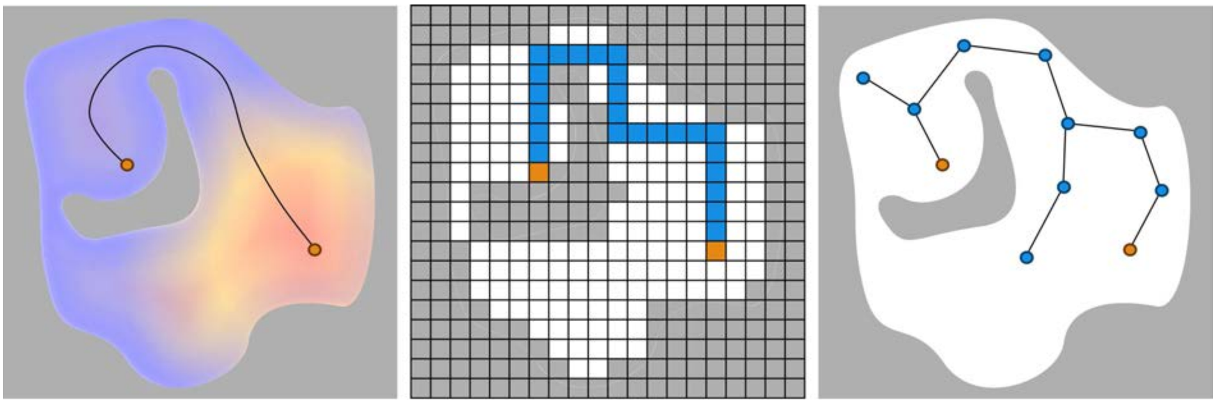


Figure 2.2: Examples of motion-planning techniques applied to the same problem: potential-field method (left), cell-decomposition method (middle) and sampling-based method (right).

putationally intensiveness and their inability to cope with uncertainty. Frequently, such disadvantages make their use not robust enough in real-world applications. Within this group the following methods lie (Latombe, 1991):

- Roadmaps methods that capture the free configuration space using a roadmap connecting the start and the end configurations, obtaining the solution path as a trajectory within that map.
- Potential field techniques that construct features that attract the robot towards the final configuration and take it away from obstacles present in the scene, combining different strength vector fields (see Fig. 2.2-left).
- Cell decomposition methods that decompose the free configuration space into smaller convex polytopes, which are then connected by a graph and searched using a graph search (see Fig. 2.2-middle).

b) Sampling-based methods that avoid the explicit construction of the configuration space, by representing the connectivity of the collision-free space with a graph of sampled configurations (see Fig. 2.2-right). Quite effective general strategies have been developed using sampling-based techniques (Elbanhawi and Simic, 2014), being the most relevant approaches the Rapidly-exploring Random Trees planners, RRT (Kuffner and LaValle, 2000), and the Probabilistic Road Map planners, PRM (Kavraki et al., 1996). However, these original approaches have some problems when, for instance, there are narrow passages. Hence, several variations were then presented proposing different improvements, dealing with constraints (Stilman, 2010), or biasing the sampling towards more promising regions of the configuration space using a sampling based on Principal Component Analysis (Rosell et al., 2013), an adaptive workspace biasing (Zucker et al., 2008), a multi-resolution PRM planner (Yang and Brock, 2004), dynamic domains (Yershova et al., 2005), or retraction steps (Zhang and Manocha, 2008). These algorithms do not take into account any path quality while planning as opposite to other sampling-based approaches like, for instance, the Transition-based RRT, T-RRT (Jaillet et al., 2010), and the Vector-Field RRT, VF-RRT (Ko et al., 2014). Nevertheless, neither these algorithms are guaranteed to find the optimal path. Thus, some sampling-based variants like the PRM*, the RRT* and the RRT$^\#$ algorithms have been proposed to find an optimal solution (Karaman and Frazzoli, 2011; Arslan and Tsiotras, 2013). In addition, approaches like, for instance, the Synergistic Combination of Layers of Planning, SyCLoP (Plaku et al., 2010), and the Kinodynamic Planning by Interior-Exterior Cell Exploration, KPIECE (Suçan and Kavraki, 2012), have been proposed to efficiently solve motion-planning problems with dynamics.

c) Optimization-based methods that treat the motion planning problem as a pure numerical optimization problem. These algorithms represent the requirement of obstacle avoidance with a set of inequalities on the configuration parameters, i.e. an optimization method starts from one or more trajectories (Pan et al., 2014), that may contain collisions or perhaps violates constraints and tries to converge to a high-quality trajectory satisfying the constraints. However, the cost functions that optimize these methods typically have a large number of local minima, thus, finding the global solution strongly depends on the initial guess (the general formulation is NP-hard, although when the given initial guesses are sufficiently close to the global solution, the optimization converges in polynomial time). In addition to this, unless a good-enough initial guess is given (which is already a high-consuming computational problem), these methods usually produce jerky movements in cluttered environments (if they are able to find a feasible solution). Some of these approaches based on optimization techniques are the CHOMP (Ratliff et al., 2009), the STOMP (Kalakrishnan et al., 2011) and the TrajOpt (Schulman et al., 2014) algorithms.

2.2 Motion planning of coupled multi-robot systems

During the last decade, robots are getting closer to humans, introducing thus the need for anthropomorphic motion to allow better and safer human-robot interactions (i.e. in case robots move anthropomorphically, human operators can more easily predict the robot motions avoiding thus possible injuries as well as correlate the robot configurations with the task execution, and seamlessly collaborate with the robot). In general, dual-arm systems are used to perform coordinated manipulation tasks including regrasping (Vahrenkamp et al., 2009), either arriving to a close kinematic chain, e.g. assembling a nut and a bolt or inserting a peg in a hole (Shauri and Nonami, 2011; Edsinger and Kemp, 2008), with each part being manipulated by a different arm in both cases; or cooperating with open chain coordinated movements, e.g. when the dual-arm system needs to use its hands to remove potential obstacles in order to obtain a free access to grasp a desired object (Rodríguez et al., 2014). Dual-Arm robot manipulation has been widely investigated (Quispe et al., 2015; Zöllner et al., 2004). Nevertheless, it still belongs to the most demanding challenges in robotics. Besides, this challenge gains more interest if robots are to become useful in household settings which are tailored for human arms and hands.

2.2.1 Motion planning of dual-arm systems

Motion planning for dual-arm systems is a complex problem due to the associated high dimensional configuration space (see an example in Fig. 2.3). The developed approaches to deal with these systems can be classified into centralized and decoupled (LaValle, 2006). On the one hand, the centralized approaches consider the dual-arm system as a single multi-element robot with all the DOFs of the whole system, and the planning algorithms find a collision-free path for the arms properly coordinated (Suárez et al., 2015). On the other hand, the decoupled approaches treat each arm as a single independent system. In this case, the motion planning algorithm finds first a path for each arm independently and then requires a coordination phase to avoid potential collisions between the arms when they execute their movements simultaneously (Rodríguez et al., 2014). Therefore, even when the decoupled approaches may be more efficient since they decompose the general problem into smaller problems (i.e. one for each

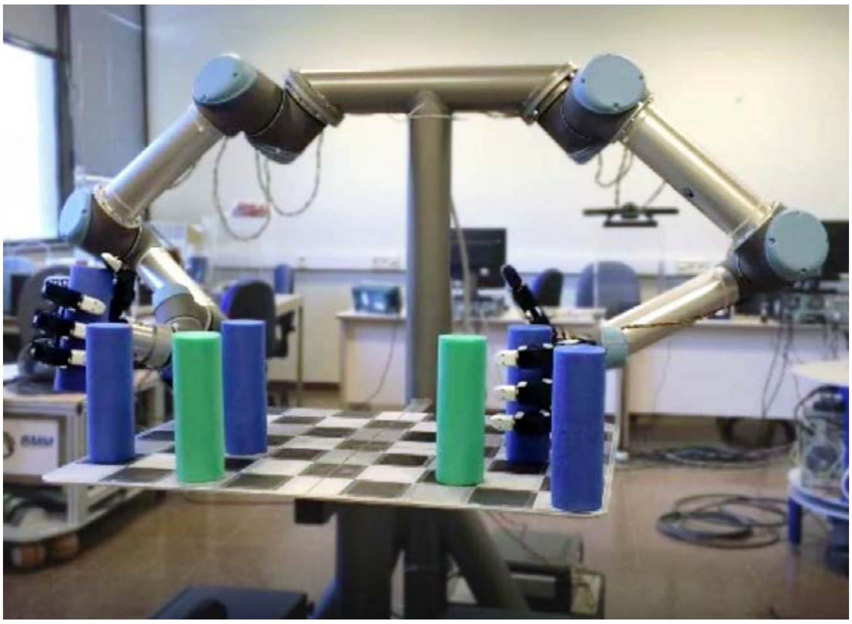


Figure 2.3: Dual-Arm robotic systems performing coordinately a bimanual task.

arm), they need an additional synchronization phase and, as a drawback, these type of approaches are not able to establish arm coordinations.

2.2.2 Motion planning of mobile manipulators

Robotic systems in which a mobile platform is combined with a robot arm are commonly known as mobile manipulators (see Fig. 2.4). Such a combined system is able to perform manipulation tasks in much larger workspaces than a manipulator with a fixed base. However, to fully exploit the advantages offered by a mobile manipulator, it is necessary to understand how to properly and effectively plan the motions of the mobile platform and the manipulator. The first approach to this task was the uncoordinated motion planning of the mobile base and the robot arm. In this case, either the platform or the arm is actuated but not both at the same time; i.e. the mobile base is first moved to a specific pose that allows the manipulator to perform its task while some quality measure may be optimized and then, with the mobile base in a fixed position, the motion planning of the robot arm is performed (Shin et al., 2003; Neo et al., 2006; Zacharias et al., 2008). Thus, the problem complexity is extremely reduced but the plan is likely non-optimal and the planner could get stuck in cluttered environments.

Figure 2.4: Mobile manipulator performing a manipulation task.

The approaches dealing with the coordinated motion of the base and the arm robot attempted first the motion planning in two levels: first, the pose of the end effector of the manipulator is obtained and then the pose of the mobile platform is selected to maximize an specific performance function, e.g. maximizing the manipulability of the robot arm (Yamamoto and Yun, 1994) or the stability of the whole system (Huang et al., 1998). However, in order to use the full redundancy of the robot system, other works used a whole body planner to solve the motion planning in the complete configuration space of the robot. For instance, the RRT path planning approach was adapted and combined with inverse kinematics algorithms for motion planning along given end-effector paths (Oriolo and Mongillo, 2005). The PRM algorithm was also adapted to plan task-consistent collision-free motion for mobile manipulators (Yang and Brock, 2010). Besides, a graph-based representation of the planning problem was used for door opening problems (Chitta et al., 2010).

2.3 Planning human-like motions

Towards the goal of robots mimicking the human arm movements, the multi-joint coordination of the human arms has been analyzed and modeled, either pursuing the direct on-line teleoperation of the robot arms (Basañez and Suárez, 2009), or with the aim of analyzing these movements and getting some valuable information to be applied later in a planning phase, allowing a lower planning complexity and/or looking for more human-like movements. Some approaches were proposed to solve the motion planning of dual-arm manipulation tasks like, for instance, using "parallel" roadmaps constructed on different layers of the search space, one per each class of the inverse-kinematics solutions and connected through singular configurations (Gharbi et al., 2008); constructing a graph using pre-defined and runtime-generated motion-primitives (Cohen et al., 2012); or combining the motion planning itself with a gradient-descent approach in the pre-computed reachability-space of the robot (Vahrenkamp et al., 2009). Nevertheless, when the motion planning of an anthropomorphic system is attempted, the direct use of the real movements of a human being as a reference is more than common (Argall et al., 2009).

Several studies have been done in order to generate human-like motions by imitating human arm postures as closely as possible (Wang and Artemiadis, 2013; Artemiadis et al., 2010; Kim et al., 2006). In most of these studies, the human movements are measured using a motion-capture system and then converted to the motion of an anthropomorphic robot arm by minimizing the posture difference between the robot and the human arm (Liarokapis et al., 2013). However, these approaches are not able to directly describe the dependencies among the degrees of freedom of the human movements.

2.3.1 Optimization-based methods

A common assumption of many approaches analyzing human motion is that humans try to minimize an unknown cost function while doing everyday manipulation tasks. Hence, minimizing this cost function, human-like motions can be obtained. Typical examples of cost function assumed to be minimized by human in everyday tasks are the following:

- Hand jerk (Flash and Hogans, 1985), i.e. the third time-derivative of the hand position.
- Joint jerk (Rosenbaum et al., 1995), i.e. the third time-derivative of the joint position.
- Torque changes (Uno et al., 1989).
- A convex combination of several cost functions with the weighting factors of the considered costs chosen to describe an observed human motion (Albrecht et al., 2011).

Other works obtained anthropomorphic configurations by constraining the elbow position when solving the inverse-kinematics problem of redundant robotic arms (Wang and Artemiadis, 2013; Kim et al., 2006). These works used previous knowledge from motion recordings of human arms to fit a polynomial expression of the variables defining the desired transformation of the end-effector to estimate the necessary swivel angle (i.e. the rotation angle of the plane defined by the upper and lower arm around a virtual axis that connects the shoulder and the wrist), and then used this angle as a parameter for the inverse-kinematics algorithm, simplifying the problem. Other approaches described the dependencies among the human joint angles using a Bayesian network and then used an objective function, based on this model, in a closed-loop inverse kinematic algorithm (Artemiadis et al., 2010). Hidden Markov Models (HMM) were also used for modeling arm motions with imitation purposes (Asfour et al., 2006a; Calinon et al., 2010). These models are able to extract redundancies across multiple demonstrations and build time-independent models to reproduce the dynamics of the observed movements. However, most of the works using this approach are based on cost functions and optimization techniques that drive the robots based on a finite recorded set. Hence, these models may have difficulties to generate a variety of new human-like motions. Moreover, these functions are quite complex to be used for the inverse kinematics of the robots.

2.3.2 Methods based on synergies

Pioneering works used the multi-joint coordination of the human hands in the planning of motions and grasp configurations of anthropomorphic robotic hands. The basic idea was to establish couplings between the DOFs of the robotic fingers equivalent to those couplings existing in the human hand. Relevant works dealt first with the grasping problem analyzing the correlations of the finger joints when the hand was grasping virtual objects (Santello et al., 1998) and real objects (Santello et al., 2002), and called these joint correlations “hand postural synergies”. Other works used the same concept to find pre-grasp hand configurations (Ciocarlie and Allen, 2009), calling “eigengrasp” to each independent hand movement involving correlated movements of all the joints. These works allowed a reduction of the grasp space down to a bidimensional space (see Fig. 2.5).

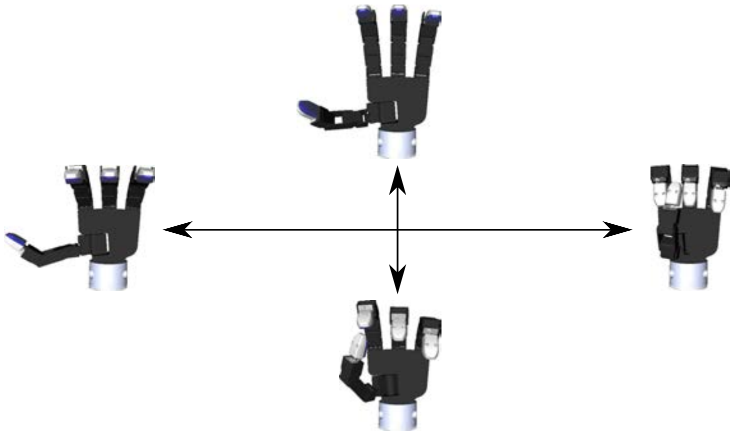


Figure 2.5: Example of grasp configurations reduced to a bidimensional space using eigengrasps.

The concept of synergy (in Greek *synergos* means working together) has been defined differently and in different contexts. Although synergies originally captured the idea that there exists a set of fixed combinations of human muscles that are preferably controlled as functional units (Bernstein, 1967), this term has been generalized to indicate the common patterns observed in the behaviors of joint positions, velocities, forces, muscle actions, among others. The synergies existing in the human hand were also used for other objectives (Bicchi et al., 2011). Some relevant examples are the analysis and design of robotic hands in order to mimic human grasps (Ficuciello et al., 2014), the design of specific hand control systems (Palli et al., 2014; Wimböck et al., 2011), the selection of grasping forces (Gabiccini et al., 2011; Prattichizzo et al., 2013), the telemanipulation of high-DOF robotic systems (Tsoli and Jenkins, 2007), or the identification of the hand pose using low-cost gloves (Bianchi et al., 2013b) and the design of the gloves for this purpose (Bianchi et al., 2013a). Several of these works used synergies for grasp synthesis, thus the synergies were determined from the human hand analysis while performing graspings. Nevertheless, there were also works using the same concept to perform motion planning trying to mimic human hand postures (Rosell et al., 2011, 2013). In this case the correlations between the finger joints were used to determine movement directions so they were called “principal motion directions” (PMDs). Motion planning requires the determination of the correlation of the finger movements when they are freely moved trying to cover the whole hand workspace without any external constraint (Sun et al., 2010). Other applications dealt with the synthesis of human-like motions in graphic applications (Safonova et al., 2004). In addition, a related alternative work fixed artificial synergies to impose a common behavior to a team of mobile robots (Rosell and Suárez, 2014).

A related problem needed to be faced when trying to obtain human synergies is the capture of the human configurations in order to get proper information. In the works mentioned above, the problem of obtaining the hand synergies was mainly addressed using: a) vision systems (Infantino et al., 2005; Wachs et al., 2005), which have the frequent problem of visual occlusions and usually require special marks on the hand to facilitate the identification of the hand configuration; and, b) sensorized gloves (Miners et al., 2005; Cobos et al., 2010; Geng et al., 2011), including the use of non-anthropomorphic robotic hands (Wojtara and Nonami, 2004). In the particular application of moving a prosthetic hand, the analysis of forearm electromyogram signals (EMG) was proposed (Kondo et al., 2008), but this is not practical in most general-purpose robotic applications. Moreover, in order to obtain the synergies of the robot system there exists two different approaches:

(a) The real human synergies are obtained and then mapped in a non-trivial way to the robotic dual-arm system (Gioioso et al., 2013).

(b) The human arm movements are first mapped onto the used robotic system and then the synergies are directly obtained for the used robotic system in correspondence with the movements of the human operator (Rosell et al., 2011).

Concepts equivalent to the postural synergies used in hand motion planning, can be also applied to the motion planning of a dual-arm system with anthropomorphic features, with the aim of reducing the planning complexity and looking for movements that mimic the human movements. This application was not reported yet. Moreover, all of the above mentioned works dealt only with synergies involving correlations between joint positions. Nevertheless, it seems natural to complement the information embedded in these traditional synergies with the information obtained from samples captured in the velocity space, i.e. the space of the first derivative of the configuration trajectories, generalizing thereby the concept of postural synergies. Profiles of postural synergies varying over time were derived from joint-velocity profiles of rapid grasping movements (Vinjamuri et al., 2010). Some studies expressed the angular velocities of finger joints as linear combinations of a small number of “kinematic synergies”, which were also angular velocities of finger joints (Grinyagin et al., 2005; Vinjamuri et al., 2007). The kinematic synergies were also found in 7-DOF arm movements (d'Avella et al., 2006). However, the velocity synergies have not been investigated in dual-arm movements neither used in the motion planning of bimanual tasks. Therefore, this is an open field of research.

3

Using synergies to characterize tasks and robot motions

This chapter introduces the concept of zero-order and first-order synergies as well as their computation method (which implies in this work the use of a mapping from the human configuration space to the robot configuration space, also described in this chapter). Besides, the use of the synergies is proposed to characterize tasks and robot motions.

3.1 Joint synergies

Although the concept of synergies (from the Greek *synergos*, working together) has been defined in different contexts and used with different meanings, along this document synergies are used to indicate common patterns and joint correlations within human demonstrations of movements. Depending on whether these patterns are observed in joint positions or joint velocities, a distinction is made between zero-order synergies and first-order synergies, respectively. This section presents these two concepts, including a description of how they are computed.

3.1.1 Zero-Order synergies

Zero-Order synergies have been given different names in the related bibliography, depending on the application they were used for, like for instance “postural synergies” (Santello et al., 2002), “eigengrasps” (Ciocarlie and Allen, 2009), or “principal motion directions” (Rosell et al., 2013); in this work “zero-order synergies” is used, in contrast to the “first-order synergies” introduced in the next subsection.

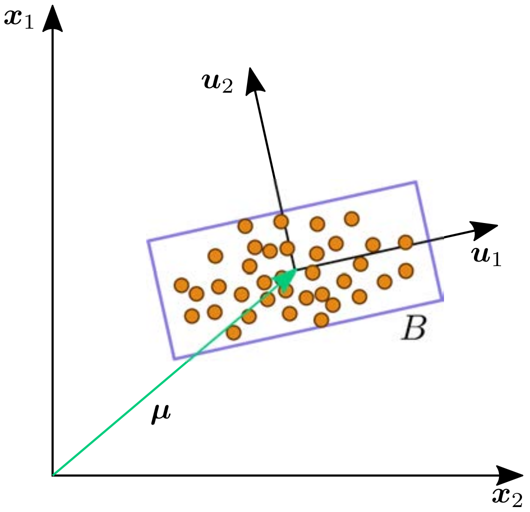


Figure 3.1: Zero-Order basis ${}^0\mathcal{S}$ obtained after applying a Principal Component Analysis to a set of configuration samples. The basis ${}^0\mathcal{S}$ is characterized by the sample barycenter $\boldsymbol{\mu}$ and the axes $\boldsymbol{u}_1$ and $\boldsymbol{u}_2$. Note that each axis $\boldsymbol{u}_j$ represents a zero-order *synergy* and implies the correlated movement of all the system DOFs, i.e. $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$.

Zero-Order synergies represent correlations between DOFs of the system under study (for instance the joint positions of a human hand or a human arm), and they are obtained from an analysis of a set of configuration samples of such system. The analysis of these samples is done with a Principal Component Analysis (PCA) that returns a new basis of the configuration space (eigenvectors) with the axes sorted in decreasing order of the associated dispersion of the samples along each axis (eigenvalues), see Fig. 3.1. Each axis of this new basis represents a zero-order synergy, i.e. the movement along one particular axis, equivalent to one single DOF, implies a correlated movement of several (or all) the actual DOFs of the system. A basis of the configuration space representing zero-order synergies will be called from now on a zero-order synergy basis ${}^0\mathcal{S}$.

It must be highlighted that non-linear approaches to obtain synergies have been also proposed (Ficuciello et al., 2018). Some examples are the *Functional Principal Component Analysis* (Huang and Sun, 2015) and the *Gaussian Process Latent Variable Model* (Romero et al., 2013). However, the simple linear approximation of the PCA is enough to capture the subspace where the demonstrated motions lie. In addition, it has been verified to be useful and implementable by a drive mechanism (Chen et al., 2015) and a real-time algorithm (Wimböck et al., 2011). Hence, the PCA is used in this document as a first attempt to integrate the synergies in the planning of human-like robot motions.

3.1.2 First-Order synergies

As a generalization of the zero-order synergies, the concept of first-order synergies is introduced here, considering as such the result of applying the PCA to a set of *velocity samples* instead of applying it to a set of configuration samples. These synergies are named here as first-order synergies because they are obtained in the space of the first derivative of the configuration space, i.e. the velocity space.

In practice, the motion-capture devices used in this work (see Subsection 1.3.2) only record successive configurations of the human-demonstrated movements, i.e. no information regarding the velocities is captured. Nevertheless, if a set of M configuration samples $\boldsymbol{q}_k$ is captured at a given constant rate, the associated velocities $\dot{\boldsymbol{q}}_k$ can be approximated using m-order accuracy finite differences (assuming that m is a positive, even integer strictly lower than M) as:

$$\dot{\boldsymbol{q}}_k = \begin{cases} \sum_{i=0}^{m} f_i \, \boldsymbol{q}_{k+i} & \text{if } k - \frac{m}{2} \leq 0 \\ \sum_{i=-\frac{m}{2}}^{\frac{m}{2}} c_i \, \boldsymbol{q}_{k+i} & \text{if } k - \frac{m}{2} > 0 \text{ and } k + \frac{m}{2} \leq M \\ \sum_{i=-m}^{0} b_i \, \boldsymbol{q}_{k+i} & \text{if } k + \frac{m}{2} > M \end{cases} \quad \forall k \in [1, M] \tag{3.1}$$

where the coefficients f_i, c_i and b_i are respectively the m-order accuracy finite difference coefficients of the forward, central and backward methods (Fornberg, 1988). Note that the central method is applied to most samples, except for the first and last $\frac{m}{2}$ samples where the central method cannot be employed and the forward and the backward methods are used, respectively. In this work, $m = 2$ is chosen. Thereby, when a PCA is applied to the computed set of velocity samples, a new basis of the velocity space is obtained. This basis represents first-order synergies and, in accordance with the previous reasoning, it is called from now on a first-order basis ${}^{1}\mathcal{S}$.

3.1.3 The synergy box

For a robotic system with n DOFs, an n-dimensional box is defined using the synergies, which is called the synergy box B. This box B is centered at the barycenter of the configurations used to obtain the synergies (see blue box in Fig. 3.1) and each side of this box is aligned with a synergy $\boldsymbol{u}_j$ (i.e. an axis of the basis returned by the PCA) and measures 2λ times the standard deviation σ_j of the configurations measured along the corresponding direction. Depending on the value of λ, the box B contains a different percentage of the configuration samples distribution (i.e. a robot configuration inside the box would then be similar, in a certain degree, to the ones used to compute the synergies). The parameter λ is set, for a given α, according to the following expression

$$\lambda = \sqrt{2}\,\mathrm{erf}^{-1}\big(\sqrt[n]{1-\alpha}\,\big) \tag{3.2}$$

where $\mathrm{erf}^{-1}(\cdot)$ is the inverse error function and the samples used to obtain the synergies are assumed to be normally distributed. In this way, the box B contains $(100-\alpha)\%$ of the distribution of the configuration samples, which is proven below.

The dimension of the box can be reduced by using only $k<n$ synergies (picking them in decreasing order of the associated sample variance) such that k is the minimum value making the accumulated variance be above a confidence level of $(100-\beta)\%$ for a given β. Despite the simplification, the resulting boxes B_k still represent accurately the motions used to obtain the synergies. Thereby, if the planning of the robot motions is performed in the corresponding B_k, obtained from some given human-demonstrated movements, the planning complexity is significantly reduced and the obtained motions are still similar to the movements demonstrated by the human, i.e. they are human-like.

Proof that the synergy box B contains $(100-\alpha)\%$ of the distribution of the configuration samples

Let $\mathcal{N}$ be a multivariate normal distribution (i.e. the generalization of the univariate normal distribution to a higher-dimensional Euclidean space). The multivariate distribution $\mathcal{N}$ of a given n-dimensional continuous random vector $\boldsymbol{x} = [x_1, \ldots, x_n]^\intercal$, $\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma)$, is parametrized by the n-dimensional barycenter vector $\boldsymbol{\mu}$ and the $(n \times n)$-dimensional positive semi-definite covariance matrix Σ defined as

$$\boldsymbol{\mu} = \mathrm{E}[\boldsymbol{x}] \quad \text{and} \quad \Sigma = \mathrm{E}\big[(\boldsymbol{x}-\boldsymbol{\mu})(\boldsymbol{x}-\boldsymbol{\mu})^\intercal\big] \tag{3.3}$$

where $\mathrm{E}[\cdot]$ is the expected-value function. If Σ is invertible, the probability density function of

$\mathcal{N}$ exists and can be expressed as

$$\frac{e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^{\intercal}\Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu})}}{\sqrt{(2\pi)^n|\Sigma|}} \tag{3.4}$$

Note that the equation above reduces to the probability density function of the univariate normal distribution if $n = 1$, as it was expected.

In order to prove that the synergy box B contains $(100-\alpha)\%$ of the distribution of the configuration samples, the integral of the probability density function of the sample distribution, which is assumed to be a Gaussian $\mathcal{N}$, must be equal to $1-\alpha$ within the synergy box B. For this purpose, the definition of the error function

$$\operatorname{erf}(x) = \frac{1}{\sqrt{\pi}} \int_{-x}^{x} e^{-t^2}\,\mathrm{d}t \tag{3.5}$$

and the following change-of-variables have been used

$$\boldsymbol{x} = \boldsymbol{\mu} + \sum_{j=1}^{n} \sigma_j\, \boldsymbol{u}_j\, z_j\ , \quad \mathrm{d}\boldsymbol{x} = \prod_{j=1}^{n} \sigma_j\ \mathrm{d}\boldsymbol{z} = \sqrt{|\Sigma|}\,\mathrm{d}\boldsymbol{z}\ , \quad \boldsymbol{x} \in B \iff \boldsymbol{z} \in [-\lambda, \lambda]^n \tag{3.6}$$

$$z_j = \sqrt{2}\,t\ , \quad \mathrm{d}z_j = \sqrt{2}\,\mathrm{d}t\ , \quad z_j \in [-\lambda, \lambda] \iff t \in \left[-\frac{\lambda}{\sqrt{2}}, \frac{\lambda}{\sqrt{2}}\right] \tag{3.7}$$

Then,

$$\begin{aligned}
\int_B \mathcal{N}\,\mathrm{d}x &\overset{\substack{\text{Eq. (3.4)}\\ \downarrow}}{=} \int_B \frac{e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^{\intercal}\Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu})}}{\sqrt{(2\pi)^n|\Sigma|}}\,\mathrm{d}x \overset{\substack{\text{Eq. (3.6)}\\ \downarrow}}{=} \int_{-\lambda}^{\lambda}\!\cdots\!\int_{-\lambda}^{\lambda} \frac{e^{-\frac{1}{2}\boldsymbol{z}^{\intercal}\boldsymbol{z}}}{\sqrt{(2\pi)^n}}\,\mathrm{d}z_1\cdots\mathrm{d}z_n \\
&= \int_{-\lambda}^{\lambda}\!\cdots\!\int_{-\lambda}^{\lambda} \frac{e^{-\frac{1}{2}\sum_{j=1}^{n} z_j^2}}{\sqrt{(2\pi)^n}}\,\mathrm{d}z_1\cdots\mathrm{d}z_n = \int_{-\lambda}^{\lambda}\!\cdots\!\int_{-\lambda}^{\lambda} \prod_{j=1}^{n} \frac{e^{-\frac{1}{2}z_j^2}}{\sqrt{2\pi}}\,\mathrm{d}z_1\cdots\mathrm{d}z_n \\
&= \prod_{j=1}^{n} \int_{-\lambda}^{\lambda} \frac{e^{-\frac{1}{2}z_j^2}}{\sqrt{2\pi}}\,\mathrm{d}z_j \underset{\substack{\uparrow\\ \text{Eq. (3.7)}}}{=} \prod_{j=1}^{n} \int_{-\frac{\lambda}{\sqrt{2}}}^{\frac{\lambda}{\sqrt{2}}} \frac{e^{-t^2}}{\sqrt{\pi}}\,\mathrm{d}t \underset{\substack{\uparrow\\ \text{Eq. (3.5)}}}{=} \prod_{j=1}^{n} \operatorname{erf}\left(\frac{\lambda}{\sqrt{2}}\right) = \operatorname{erf}\left(\frac{\lambda}{\sqrt{2}}\right)^{n} \underset{\substack{\uparrow\\ \text{Eq. (3.2)}}}{=} 1-\alpha
\end{aligned} \tag{3.8}$$

□

3.1.4 Determination of robot synergies

In order to map the human movements, regarding the configurations of the arms and the position of the robot base, it must be taken into account: (a) functional constraints, i.e. the robotic wrists should be positioned and oriented similarly as the human wrists, and, (b) anthropomorphic requirements, i.e. the complete pose of the robot should look similar to the human pose. For manipulation tasks, the exact reproduction of the wrist positions is considered more critical than the human-likeness of the robot configuration. Thereby, the mapping of the movements of the human arms is formulated here as a constrained optimization problem where the second observation above is subordinated to the first one. For this, given a mobile anthropomorphic dual-arm robot (e.g. the MADAR robot, see Subsection 1.3.1), let:

- $\boldsymbol{\chi} = [x, y]^{\mathsf{T}}$ be a position of the robot base, expressed in a given world reference system.
- $\boldsymbol{\theta} = [\theta_t, \boldsymbol{\theta}_l^{\mathsf{T}}, \boldsymbol{\theta}_r^{\mathsf{T}}]^{\mathsf{T}}$ be an upper-body configuration, i.e. the concatenation of the value of the torso joint θ_t (or the orientation of the robot base) and the joint values $\boldsymbol{\theta}_l$ and $\boldsymbol{\theta}_r$ of the left and right arms. Note that the torso joint θ_t is not a real joint but this rotation is achieved with movements of the mobile base.
- $\mathcal{C}$ be the robot configuration space (see Section 2.1).
- $\boldsymbol{q} = [\boldsymbol{\chi}^{\mathsf{T}}, \boldsymbol{\theta}^{\mathsf{T}}]^{\mathsf{T}} \in \mathcal{C}$ be a configuration of the robot.
- $\mathcal{C}_{\text{free}} \subseteq \mathcal{C}$ be the subspace of the configurations $\boldsymbol{q} \in \mathcal{C}$ in which the robot is not in collision, neither with itself nor with the environment (see Section 2.1).

Besides, for each arm $i \in \{l, r\}$, with l and r denoting the left and right arms respectively, and omitting the dependence on time t and robot configuration $\boldsymbol{q}$ in the following descriptions to simplify the used notation, let:

- $\boldsymbol{p}_{\mathrm{W}_i}^{\mathrm{h}}$, $\boldsymbol{p}_{\mathrm{W}_i} \in \mathbb{R}^3$ be, respectively, the positions of the human and robotic wrists, with respect to a given world reference frame (see Fig. 3.2).
- $R_{\mathrm{W}_i}^{\mathrm{h}}$, $R_{\mathrm{W}_i} \in SO(3)$ be, respectively, the rotation matrices of the human and robotic wrists, expressed in a common reference system.

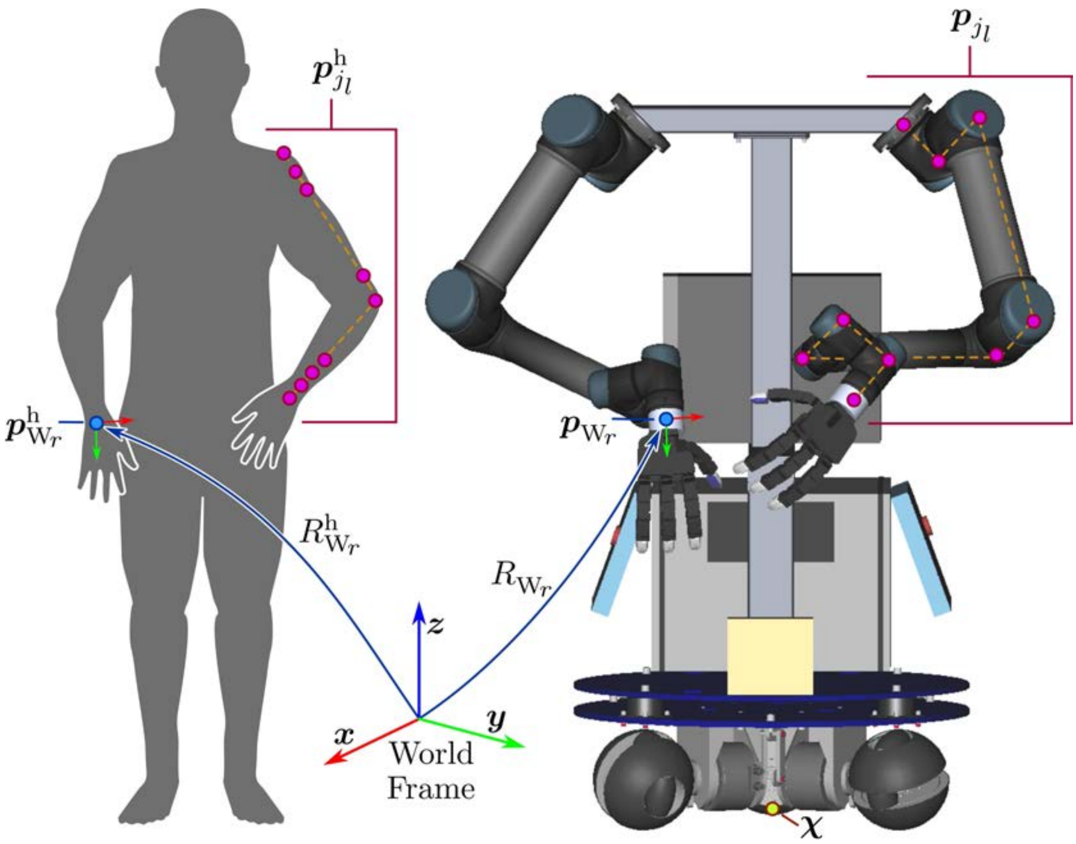


Figure 3.2: Human operator and mobile anthropomorphic dual-arm robot: points $\boldsymbol{p}_{j_i}$ on the robotic arm with their equivalent points $\boldsymbol{p}^{\mathrm{h}}_{j_i}$ on the human arm; wrist positions $\boldsymbol{p}_{\mathrm{W}_i}$ and $\boldsymbol{p}^{\mathrm{h}}_{\mathrm{W}_i}$; wrist orientations R_{W_i} and $R^{\mathrm{h}}_{\mathrm{W}_i}$; and position $\boldsymbol{\chi}$ of the robot base.

- $E_i \geq 0$ be the tracking error of the robotic wrist, considering both position and orientation differences with respect to the human wrist (see Fig. 3.2 and Fig. 3.3). Given a weight $\rho \in (0, 1)$ balancing the relative importance of the position and orientation errors, E_i is computed as

$$E_i = \rho \left\| \boldsymbol{p}^{\mathrm{h}}_{\mathrm{W}_i} - \boldsymbol{p}_{\mathrm{W}_i} \right\|^2 + (1 - \rho) \left\| \ln \left(R^{\intercal}_{\mathrm{W}_i} R^{\mathrm{h}}_{\mathrm{W}_i} \right) \right\|^2_F \tag{3.9}$$

  where $\|X\|_F$ is the Frobenius norm of a given matrix X and $\ln(R)$ is the logarithm of a given rotation matrix R (Engø, 2001). Note that bigger values of ρ penalize more the errors of position with regard to those of orientation, and vice versa. The more similar the poses of the robotic and human wrists are, the lower E_i is. Besides, E_i reaches its minimum value 0 if, and only if, both $\boldsymbol{p}_{\mathrm{W}_i} = \boldsymbol{p}^{\mathrm{h}}_{\mathrm{W}_i}$ and $R_{\mathrm{W}_i} = R^{\mathrm{h}}_{\mathrm{W}_i}$ hold (Huynh, 2009).

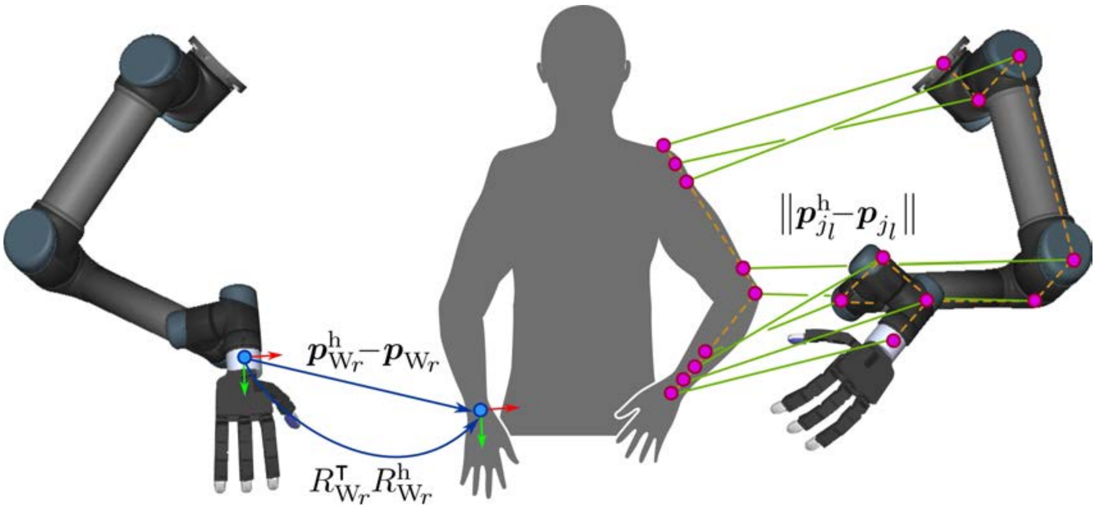


Figure 3.3: Position and orientation differences in the tracking error E_i (left side), and distances involved in the anthropomorphic dissimilarity A_i (right side). Despite showing the robotic arms free-flying, E_i and A_i are computed for both arms with the robot assembled and sharing the same base position $\boldsymbol{\chi}$.

- $A_i \geq 0$ be the anthropomorphic dissimilarity between the poses of the robotic arm and the human arm (i.e. the whole kinematic chains and not only the wrist poses) as defined below. In order to measure the anthropomorphic dissimilarity, a given number m of points $\boldsymbol{p}_{j_i}$ are selected along the robotic arm (from the shoulder to the wrist) and also their kinematically equivalent points $\boldsymbol{p}^{\mathrm{h}}_{j_i}$ on the human arm (see Fig. 3.2 and Fig. 3.3). Then, A_i is computed as the weighted sum of the squared distances between the paired $\boldsymbol{p}^{\mathrm{h}}_{j_i}$ and $\boldsymbol{p}_{j_i}$, i.e. given some positive weights ω_j

$$A_i = \textstyle\sum_{j=1}^{m} \omega_j \left\| \boldsymbol{p}^{\mathrm{h}}_{j_i} - \boldsymbol{p}_{j_i} \right\|^2 \tag{3.10}$$

  Hence, as A_i decreases, the human-likeness of the robot configuration improves. Here, $m = 9$ points $\boldsymbol{p}_{j_i}$ were chosen matching them up with the intersections of the cylinders forming the robotic arm (see Fig. 3.2). Notice that with the motion-capture devices used in this work (see Subsection 1.3.2), only the points on the shoulders, the elbows and the wrists are actually captured from the human arms. The other points $\boldsymbol{p}^{\mathrm{h}}_{j_i}$ have been selected along the shoulder-elbow and elbow-wrist rectilinear segments such that, in each segment, the ratio $\|\boldsymbol{p}_{j+1_i} - \boldsymbol{p}_{j_i}\| / \|\boldsymbol{p}^{\mathrm{h}}_{j+1_i} - \boldsymbol{p}^{\mathrm{h}}_{j_i}\|$ has the same value $\forall j \in [0, m)$.

Then, given the position of the human shoulders, elbows and wrists plus both wrist orientations, the mapping problem involves finding a collision-free configuration $\boldsymbol{q}^*$ that minimizes the anthropomorphic dissimilarity while guaranteeing at the same time the minimum tracking error between the human and the robot. This can be formalized as

$$\begin{aligned} \boldsymbol{q}^* = \underset{\boldsymbol{q} \,\in\, \mathcal{C}_{\text{free}}}{\arg\min} \;\; & A_l(\boldsymbol{q}) + A_r(\boldsymbol{q}) \\ \text{s.t.} \quad & E_l(\boldsymbol{q}) + E_r(\boldsymbol{q}) \leq E_l(\tilde{\boldsymbol{q}}) + E_r(\tilde{\boldsymbol{q}}) \quad \forall \tilde{\boldsymbol{q}} \in \mathcal{C}_{\text{free}} \end{aligned} \tag{3.11}$$

Note that finding local minima is good enough for the considered purposes and that, in most cases, a perfect tracking is feasible (i.e. $E_l = E_r = 0$). In addition to this, if closed-form solutions are available for the inverse kinematics of the robotic arms, then the dimension of the optimization error is reduced by a factor of 12, with the consequent speed up. Moreover, to speed up the collision checking, simplified models of the robot and the environment (based on boxes, spheres and capsules) can be used. Thereby, the proposed mapping procedure ensures the correct placement of the robotic wrists and, simultaneously, it handles the robot redundancy by maximizing its human-likeness. Nevertheless, the human-likeness is dependent on the differences in size and kinematic structure of the robot with respect the human (i.e. smaller robots, for instance, may have to extend the arms more than the human does to reach the desired wrist poses, leading to not so human-like arm configurations).

If the given dual-arm robot does not have a mobile base or it is preferred to have the mobile robot in a fixed position, the presented procedure is still valid if a little change is made: the constrained optimization problem in Eq. (3.11) must be then forced to consider the same position $\boldsymbol{\chi}$ of the robot base for all the time instants t in the computation of the mapped robot configurations $\boldsymbol{q}^*$. This position $\boldsymbol{\chi}$ of the robot base can be given or it can be optimized through the mapping process to reduce the overall tracking errors and anthropomorphic dissimilarities associated to the mapped robot configurations $\boldsymbol{q}$.

Regarding the hands, the mapping of their movements depends on the kinematic structure and particularities of the used robotic system to a greater extent than in the case of mapping arm movements (i.e. different mapping strategies must be used for different mechanical hands, depending on the number of fingers and the number and types of joints, or even if a non-anthropomorphic gripper is used instead). In this work, the robot used for the experiments is equipped with 16-DOF *Allegro Hand* mechanical hands (see Subsection 1.3.1) and the captured movements are mapped as follows:

1) The captured motion information regarding the little finger is discarded since the robotic hand has only three fingers besides the thumb.

2) The values of the robotic flexion/extension joints of the fingers and the thumb are computed with a joint-to-joint mapping (i.e. each one of these robotic joints and its equivalent human joint share the same value), since these robotic joints are copies of human joints.

3) The values of the remaining joints of the mechanical hand (i.e. the thumb-opposition joint and the abduction/adduction joints of the fingers and the thumb) are computed with a fingertip-position mapping (i.e. the values of these joints are chosen such that the distances between the robot fingertips and the equivalent human fingertips are minimized), since these robotic joints are not present in the human hand.

It is important to remark that other and more sophisticated mapping procedures could be used instead, like, for instance, the one proposed by Colasanto et al. (2013). Nevertheless, slight differences in the mapping strategy do not affect excessively the overall performance of the proposed approach.

Once all the captured human configurations are mapped to the robotic system and before applying a PCA to obtain the synergies, it is necessary to adjust the mapped values of those robotic joints that have a movement range wider than 2π rad, e.g. all the joints of the *UR5* robotic arms can move in the $[-2\pi,\, 2\pi]$ rad range. The idea is to add or subtract 2π rad (if needed) to the values of these joints in such a way that the mapped configurations are then grouped with a minimal variance (see Fig. 3.4). This does not modify either the resulting tracking error or the anthropomorphic dissimilarity, but it does improve the results of the PCA. Thereby, the values of these joints are adjusted as follows, assuming without loss of generality that these joints have their movement range centered in 0.

The average angle $\bar{\theta}_j$ of all the resulting mapped values θ_{j_k} of the j-th joint, see Fig. 3.4, is

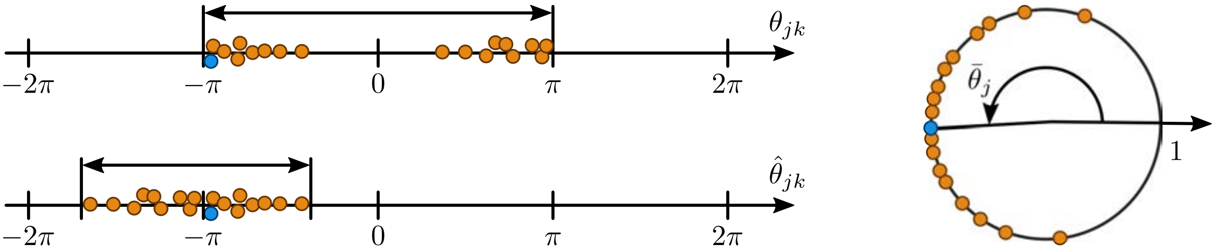


Figure 3.4: Example of sample adjustment.

computed as[1]:

$$\bar{\theta}_j = \text{atan2}\left(\bar{S}_j, \bar{C}_j\right) \quad \text{with} \quad \bar{S}_j = \frac{1}{M}\sum_{k=1}^{M} \sin\left(\theta_{j_k}\right) \quad \text{and} \quad \bar{C}_j = \frac{1}{M}\sum_{k=1}^{M} \cos\left(\theta_{j_k}\right) \tag{3.12}$$

where M is the number of processed samples. The function atan2 (y,x) is the arctangent with two arguments and its use is mandatory here to get $\bar{\theta}_j$ in the appropriate quadrant. The resulting average angle lies in the range $[-\pi, \pi)$. Then, the value $R_j = \sqrt{\bar{S}_j^{\,2} + \bar{C}_j^{\,2}}$ is a measure of the distribution of the joint values θ_{j_k}, i.e. if $\forall k \;\; \theta_{j_k} = \bar{\theta}_j$ then $R_j = 1$, and if the values of θ_{j_k} are uniformly distributed then $R_j = 0$ and $\bar{\theta}_j$ is not defined (note that $R_j = 0$ implies that $\bar{C}_j = \bar{S}_j = 0$ and atan2 $(0{,}0)$ is not defined); in this case, $\bar{\theta}_j = 0$ is arbitrarily chosen. Finally, the average angle $\bar{\theta}_j$ of the j-th joint is used to adjust the joint values θ_{j_k} obtained from the mapping to new values $\hat{\theta}_{j_k}$ as,

$$\hat{\theta}_{j_k} = \begin{cases} \theta_{j_k} & \text{if } \left|\theta_{j_k} - \bar{\theta}_j\right| \leq \pi \\ \theta_{j_k} - \text{sign}(\theta_{j_k})\, 2\pi & \text{otherwise} \end{cases} \tag{3.13}$$

minimizing in this way the variance of the samples.

---

[1] Note that the arithmetic mean cannot be used here since it does not take into account that any angle plus $\pm 2\pi$ rad represents the same joint position.

3.2 Applications

Synergies are mostly used in this work as a tool to obtain human-like robot motions in an efficient way. Nevertheless, the synergies can also be used, for example, to characterize tasks and robot motions. In this section, two original measures are introduced to compute the similarity between two tasks and to evaluate the human-likeness of a movement.

3.2.1 Task-Likeness measure

In this subsection, a measure of the likeness between two tasks is presented. Given two tasks A and B and their associated synergy bases, ${}^0\mathcal{S}_A$ and ${}^0\mathcal{S}_B$ respectively, the likeness measure $\mathcal{L}({}^0\mathcal{S}_A,\, {}^0\mathcal{S}_B)$ computes how similar are these two tasks. Then, with the obtained likeness value, it can be easily guessed, for example, if the task A could be solved with the movements used to solve task B.

Let $\mathcal{N}_A$ and $\mathcal{N}_B$ be the multivariate normal distributions associated to each synergy basis ${}^0\mathcal{S}_A$ and ${}^0\mathcal{S}_B$ (i.e. from the information in a synergy basis, it can be recovered the barycenter $\boldsymbol{\mu}$ and the covariance matrix Σ of the normal distribution $\mathcal{N}$ of the samples used to obtain the synergies). Then, the likeness measure $\mathcal{L}({}^0\mathcal{S}_A,\, {}^0\mathcal{S}_B)$ is defined as

$$\mathcal{L}({}^0\mathcal{S}_A,\, {}^0\mathcal{S}_B) = \frac{\Phi_{AB}}{\Phi_{AB_{\text{max}}}} \tag{3.14}$$

where Φ_{AB} is a measure of the overlap between $\mathcal{N}_A$ and $\mathcal{N}_B$ all over the entire n-dimensional configuration space, and $\Phi_{AB_{\text{max}}}$ is the upper bound value of Φ_{AB}. Then, $\mathcal{L}({}^0\mathcal{S}_A,\, {}^0\mathcal{S}_B) = 0$ represents the minimum likeness (i.e. the task A can be hardly solved using the movements associated to the task B, and vice versa, since the tasks are completely different) and $\mathcal{L}({}^0\mathcal{S}_A,\, {}^0\mathcal{S}_B) = 1$ represents the maximum likeness (i.e. the task A can be presumably solved using the movements associated to the task B, and vice versa, since tasks A and B could actually be the same task).

Φ_{AB} is defined as the integral of the product of $\mathcal{N}_A$ and $\mathcal{N}_B$ over the entire space (see Fig. 3.5), i.e.

$$\Phi_{AB} = \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} \mathcal{N}_A \mathcal{N}_B \, d\boldsymbol{x} \tag{3.15}$$

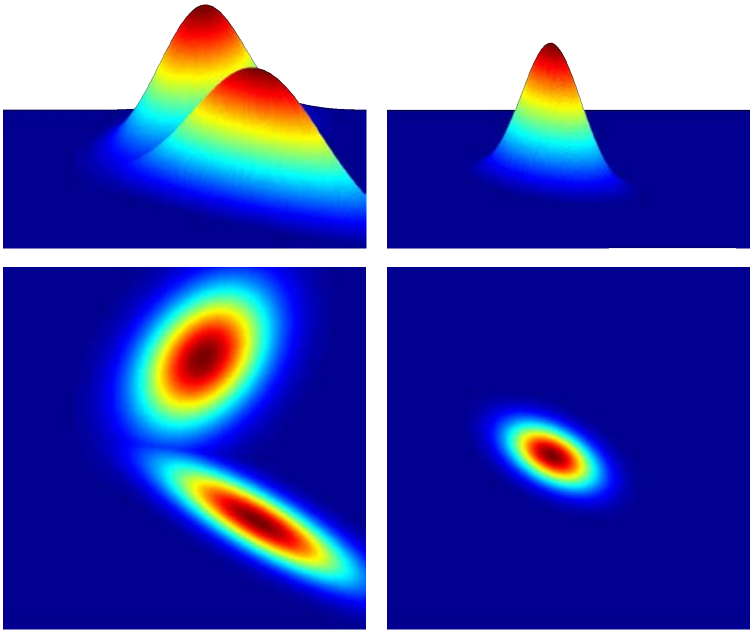


Figure 3.5: Surface plots of two bivariate normal distributions $\mathcal{N}_A$ and $\mathcal{N}_B$ (left), and their product (right), in both cases the top row is a perspective view and the bottom row is a top view. Note that the product of the two normal distributions, is also a normal distribution, up to a scale factor.

In practice, as it is proven below, Φ_{AB} can be computed as

$$\Phi_{AB} = \frac{e^{-\frac{1}{2}(\boldsymbol{\mu}_A-\boldsymbol{\mu}_B)^{\intercal}(\Sigma_A+\Sigma_B)^{-1}(\boldsymbol{\mu}_A-\boldsymbol{\mu}_B)}}{\sqrt{(2\pi)^n\,|\Sigma_A+\Sigma_B|}} \tag{3.16}$$

where n is the number of DOF of the robotic system and the components of $(\boldsymbol{\mu}_A - \boldsymbol{\mu}_B)$ are expressed in the range $[-\pi,\, \pi)$ rad. Note that even when the joint values μ_{A_j} and μ_{B_j} lie in the range $[-\pi,\, \pi)$ rad, the simple signed difference angle between them could lie outside this range. Besides, as it is also proven below, Φ_{AB} has the following bounds

$$0 < \Phi_{AB} \leq \Phi_{AB_{\text{max}}} = \frac{1}{\pi^{\frac{n}{2}}\prod_{j=1}^{n}(\sigma_{A_j} + \sigma_{B_j})} \tag{3.17}$$

where $\sigma_{A_j}^2$ and $\sigma_{B_j}^2$ are the eigenvalues of the covariance matrices Σ_A and Σ_B recovered from the synergy bases ${}^0\mathcal{S}_A$ and ${}^0\mathcal{S}_B$, respectively (i.e. σ_{A_j} is the standard deviation of the samples used to compute ${}^0\mathcal{S}_A$, measured along $\boldsymbol{u}_{A_j}$).

The introduced task-likeness measure can also be used to cluster tasks and to evaluate numerically the effect of using synergies in the performance of the motion planning, as it is done in the next two chapters.

Proof of Eq. (3.16)

First, using Eq. (3.4), the product $\mathcal{N}_A\mathcal{N}_B$ results

$$\begin{aligned}
\mathcal{N}_A\mathcal{N}_B &= \frac{e^{-\frac{1}{2}\left(\boldsymbol{\mu}_A^\intercal\Sigma_A^{-1}\boldsymbol{\mu}_A+\boldsymbol{\mu}_B^\intercal\Sigma_B^{-1}\boldsymbol{\mu}_B+\boldsymbol{x}^\intercal\left(\Sigma_A^{-1}+\Sigma_B^{-1}\right)\boldsymbol{x}-2\left(\boldsymbol{\mu}_A^\intercal\Sigma_A^{-1}+\boldsymbol{\mu}_B^\intercal\Sigma_B^{-1}\right)\boldsymbol{x}\right)}}{(2\pi)^n\sqrt{|\Sigma_A|\,|\Sigma_B|}} \\
&= \frac{e^{-\frac{1}{2}\left(\boldsymbol{\mu}_A^\intercal\Sigma_A^{-1}\boldsymbol{\mu}_A+\boldsymbol{\mu}_B^\intercal\Sigma_B^{-1}\boldsymbol{\mu}_B-\boldsymbol{\mu}_C^\intercal\Sigma_C^{-1}\boldsymbol{\mu}_C+(\boldsymbol{x}-\boldsymbol{\mu}_C)^\intercal\Sigma_C^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_C)\right)}}{(2\pi)^n\sqrt{|\Sigma_A|\,|\Sigma_B|}} \\
&= \frac{\sqrt{|\Sigma_C|}\,e^{-\frac{1}{2}\left(\boldsymbol{\mu}_A^\intercal\Sigma_A^{-1}\boldsymbol{\mu}_A+\boldsymbol{\mu}_B^\intercal\Sigma_B^{-1}\boldsymbol{\mu}_B-\boldsymbol{\mu}_C^\intercal\Sigma_C^{-1}\boldsymbol{\mu}_C\right)}}{\sqrt{(2\pi)^n\,|\Sigma_A|^{-\frac{1}{2}}\,|\Sigma_B|}}\mathcal{N}_C
\end{aligned} \tag{3.18}$$

where $\Sigma_C = \left(\Sigma_A^{-1}+\Sigma_B^{-1}\right)^{-1}$, and $\boldsymbol{\mu}_C = \Sigma_C(\Sigma_A^{-1}\boldsymbol{\mu}_A + \Sigma_B^{-1}\boldsymbol{\mu}_B)$. Notice that the product $\mathcal{N}_A\mathcal{N}_B$ results into another normal distribution $\mathcal{N}_C$ (with barycenter $\boldsymbol{\mu}_C$ and covariance matrix Σ_C) multiplied by a scale factor, which can be further simplified.

On the first hand, from the *matrix inversion lemma* (Woodbury, 1950) the following equalities are derived

$$\Sigma_C = \left(\Sigma_A^{-1}+\Sigma_B^{-1}\right)^{-1} = \begin{cases} \Sigma_A - \Sigma_A(\Sigma_A+\Sigma_B)^{-1}\Sigma_A \\ \Sigma_B - \Sigma_B(\Sigma_A+\Sigma_B)^{-1}\Sigma_B \end{cases} \tag{3.19}$$

$$\begin{aligned}
\Sigma_A^{-1}\Sigma_C\Sigma_B^{-1} &= \Sigma_A^{-1}\left(\Sigma_C\left(\Sigma_A^{-1}+\Sigma_B^{-1}\right)-\Sigma_C\Sigma_A^{-1}\right) = \Sigma_A^{-1}\left(I-\Sigma_C\Sigma_A^{-1}\right) \\
&= \Sigma_A^{-1}-\Sigma_A^{-1}\Sigma_C\Sigma_A^{-1} = (\Sigma_A+\Sigma_B)^{-1}
\end{aligned} \tag{3.20}$$

where I is the identity matrix. Therefore,

$$\begin{aligned}
&\boldsymbol{\mu}_A^\intercal\Sigma_A^{-1}\boldsymbol{\mu}_A + \boldsymbol{\mu}_B^\intercal\Sigma_B^{-1}\boldsymbol{\mu}_B - \boldsymbol{\mu}_C^\intercal\Sigma_C^{-1}\boldsymbol{\mu}_C \\
&= \boldsymbol{\mu}_A^\intercal\left(\Sigma_A+\Sigma_B\right)^{-1}\boldsymbol{\mu}_A + \boldsymbol{\mu}_B^\intercal\left(\Sigma_A+\Sigma_B\right)^{-1}\boldsymbol{\mu}_B - 2\boldsymbol{\mu}_A^\intercal\left(\Sigma_A+\Sigma_B\right)^{-1}\boldsymbol{\mu}_B \\
&= (\boldsymbol{\mu}_A-\boldsymbol{\mu}_B)^\intercal\left(\Sigma_A+\Sigma_B\right)^{-1}(\boldsymbol{\mu}_A-\boldsymbol{\mu}_B)
\end{aligned} \tag{3.21}$$

On the other hand, it is known by the *matrix determinant lemma* (Harville, 1997) that

$$|\Sigma_C| = \left|\Sigma_A^{-1} + \Sigma_B^{-1}\right|^{-1} = |\Sigma_A||\Sigma_B||\Sigma_A + \Sigma_B|^{-1} \tag{3.22}$$

These statements simplify Eq. (3.18) to

$$\mathcal{N}_A\mathcal{N}_B = \frac{e^{-\frac{1}{2}(\boldsymbol{\mu}_A-\boldsymbol{\mu}_B)^\intercal(\Sigma_A+\Sigma_B)^{-1}(\boldsymbol{\mu}_A-\boldsymbol{\mu}_B)}}{\sqrt{(2\pi)^n|\Sigma_A+\Sigma_B|}}\mathcal{N}_C \tag{3.23}$$

And since the integral of a probability density function over the entire space is 1 by definition, Φ_{AB} can be expressed as

$$\Phi_{AB} = \int_{-\infty}^{\infty}\cdots\int_{-\infty}^{\infty}\mathcal{N}_A\mathcal{N}_B\, d\boldsymbol{x} = \frac{e^{-\frac{1}{2}(\boldsymbol{\mu}_A-\boldsymbol{\mu}_B)^\intercal(\Sigma_A+\Sigma_B)^{-1}(\boldsymbol{\mu}_A-\boldsymbol{\mu}_B)}}{\sqrt{(2\pi)^n|\Sigma_A+\Sigma_B|}} \tag{3.24}$$

□

Proof of Eq. (3.17)

On the first hand, it is known by the results of Fiedler (1971) that

$$|\Sigma_A + \Sigma_B| \geq \textstyle\prod_{j=1}^{n}\left(\sigma_{A_j}^2 + \sigma_{B_j}^2\right) \tag{3.25}$$

On the other hand, $e^{-x^2} \in (0,\, 1]$, $x \in \mathbb{R}$. With this in mind, lower and upper bounds for Φ_{AB} can be obtained

$$0 < \Phi_{AB} \leq \widetilde{\Phi}_{AB_{\text{max}}} = \left((2\pi)^n\textstyle\prod_{j=1}^{n}\left(\sigma_{A_j}^2 + \sigma_{B_j}^2\right)\right)^{-\frac{1}{2}} \tag{3.26}$$

Note that the equality is held when $\boldsymbol{\mu}_A = \boldsymbol{\mu}_B$ and $\boldsymbol{u}_{A_j} = \boldsymbol{u}_{B_j}\forall j$. Nevertheless, $\widetilde{\Phi}_{AB_{\text{max}}}$ is not the minimum upper bound since the inequality becomes an equality no matter what the eigenvalues of Σ_A and Σ_B are; $\mathcal{L}({}^0\mathcal{S}_A,\, {}^0\mathcal{S}_B)$ is desired to be 1 (i.e. $\Phi_{AB} = \Phi_{AB_{\text{max}}}$) if and only if ${}^0\mathcal{S}_A$ and ${}^0\mathcal{S}_B$ are exactly the same synergy basis (i.e. $\boldsymbol{\mu}_A = \boldsymbol{\mu}_B$ and $\Sigma_A = \Sigma_B$). Therefore, using the fact that $2(x^2 + y^2) \geq (|x| + |y|)^2$ (the equality holds $\forall x = y$), the minimum upper bound can be defined as

$$0 < \Phi_{AB} \leq \widetilde{\Phi}_{AB_{\text{max}}} \leq \Phi_{AB_{\text{max}}} = \frac{1}{\pi^{\frac{n}{2}}\prod_{j=1}^{n}(\sigma_{A_j} + \sigma_{B_j})} \tag{3.27}$$

□

3.2.2 Human-Likeness index

In order to evaluate the human-likeness of different paths, this subsection introduces a human-likeness index $\mathcal{H}$. This index computes the misalignment of a path with respect to first-order bases ${}^{1}\mathcal{S}$ obtained from human movements. The first-order synergies are not similar all over the configuration space, i.e. different correlations are observed in the joint velocity space for movements performed in different regions of $\mathcal{C}$. Thus, it is assumed that there exists a given basis ${}^{1}\mathcal{S}$ assigned to each robot configuration $\boldsymbol{q}$ of the configuration space $\mathcal{C}$. Then, the value $\mathcal{H}$ of a path $\mathcal{P}$ is defined as

$$\mathcal{H} = 1 - \frac{1}{L} \int_{\mathcal{P}} \textsc{Misalignment}(\boldsymbol{q}, \boldsymbol{v}) \, \mathrm{d}\boldsymbol{q} \tag{3.28}$$

where L is the path length, $\boldsymbol{v}$ is the advance direction of the path at the configuration $\boldsymbol{q}$, and $\textsc{Misalignment}(\boldsymbol{q}, \boldsymbol{v})$ is a function that returns the misalignment η of the direction $\boldsymbol{v}$ with respect to the basis ${}^{1}\mathcal{S}$ associated to $\boldsymbol{q}$. This misalignment η is measured as

$$\eta = \frac{1}{\pi} \operatorname{acos}\big((1-\rho)\Phi_{\boldsymbol{\mu}} + \rho\Phi_{\Sigma}\big) \tag{3.29}$$

where:

- $\Phi_{\boldsymbol{\mu}} \in [-1, 1]$ is a measure that represents the alignment between $\boldsymbol{v}$ and $\boldsymbol{\mu}$ (see Fig. 3.6-left), where $\boldsymbol{\mu}$ is the barycenter of the samples used to obtain ${}^{1}\mathcal{S}$. $\Phi_{\boldsymbol{\mu}}$ is computed as

  $$\Phi_{\boldsymbol{\mu}} = \operatorname{sgn}(\boldsymbol{v} \cdot \boldsymbol{\mu}) e^{-\frac{1}{2}(\boldsymbol{w} - \boldsymbol{\mu})^{\intercal}\Sigma^{-1}(\boldsymbol{w} - \boldsymbol{\mu})} \tag{3.30}$$

  where $\operatorname{sgn}(x)$ is the sign function. $\Phi_{\boldsymbol{\mu}}$ is positive if $(\boldsymbol{v} \cdot \boldsymbol{\mu}) > 0$, and negative otherwise. Besides, $|\Phi_{\boldsymbol{\mu}}|$ is proportional to the value of the probability density function of $\mathcal{N}(\boldsymbol{\mu}, \Sigma)$ evaluated at $\boldsymbol{w}$, which is a scaled version of $\boldsymbol{v}$ so that the projection of $\boldsymbol{w}$ onto $\boldsymbol{\mu}$ is $\boldsymbol{\mu}$ itself (see Fig. 3.6-left), i.e. $\boldsymbol{w} = \frac{\boldsymbol{\mu} \cdot \boldsymbol{\mu}}{\boldsymbol{v} \cdot \boldsymbol{\mu}} \boldsymbol{v}$. Therefore, $|\Phi_{\boldsymbol{\mu}}| = 1$ when $\boldsymbol{v}$ and $\boldsymbol{\mu}$ are parallel and $|\Phi_{\boldsymbol{\mu}}| = 0$ when $\boldsymbol{v}$ and $\boldsymbol{\mu}$ are orthogonal.

- $\Phi_{\Sigma} \in [-1, 1]$ is a measure that represents the alignment of $\boldsymbol{v}$ and the direction $\boldsymbol{u}_1$ of largest

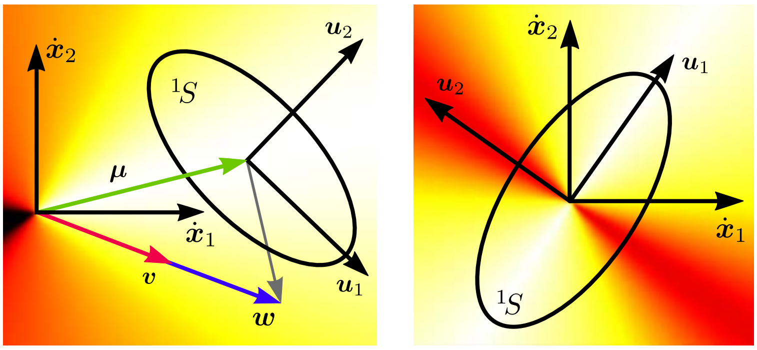


Figure 3.6: Misalignment of each direction $\boldsymbol{v} = [\dot{x}_1, \dot{x}_2]$, when the first-order basis ${}^1\mathcal{S}$ is far from the origin, i.e. $\Phi \approx \Phi_{\boldsymbol{\mu}}$ (left), and when ${}^1\mathcal{S}$ is exactly centered at the origin, i.e. $\Phi = \Phi_\Sigma$ (right). Bright yellow colors denote better alignments than dark red colors. Σ is represented by an ellipse oriented according to the eigenvectors $\boldsymbol{u}_j$ of Σ and with semiaxes proportional to the square roots of the eigenvalues of Σ. A sample of the velocities $\boldsymbol{v}$ and $\boldsymbol{w}$ is also shown.

variance of Σ (see Fig. 3.6-right), where Σ is the covariance matrix of the samples used to obtain ${}^1\mathcal{S}$. Φ_Σ is computed as

$$\Phi_\Sigma = 2\,\frac{\hat{\boldsymbol{v}}^{\mathsf{T}}\Sigma\hat{\boldsymbol{v}}}{\boldsymbol{u}_1^{\mathsf{T}}\Sigma\boldsymbol{u}_1} - 1 \quad \text{with} \quad \hat{\boldsymbol{v}} = \frac{\boldsymbol{v}}{\|\boldsymbol{v}\|} \tag{3.31}$$

where $\hat{\boldsymbol{v}}^{\mathsf{T}}\Sigma\hat{\boldsymbol{v}}$ is the variance of Σ in the direction of $\boldsymbol{v}$, and $\boldsymbol{u}_1^{\mathsf{T}}\Sigma\boldsymbol{u}_1$ is the variance of Σ in the direction of $\boldsymbol{u}_1$. The quotient of these two variances takes the maximum value 1 when $\boldsymbol{v}$ and $\boldsymbol{u}_1$ are parallel, and the minimum value 0 when $\boldsymbol{v}$ is parallel to $\boldsymbol{u}_n$, the direction of smallest variance of Σ. To obtain Φ_Σ, this quotient is then expanded from the interval $[0,1]$ to the interval $[-1,1]$ with a linear transformation.

- $\rho \in [0,1]$ weights the importance of Φ_Σ, with respect to $\Phi_{\boldsymbol{\mu}}$, in the computation of η. The weight ρ grows as the origin of ${}^1\mathcal{S}$ gets closer to the origin of the velocity space. Therefore, it represents the proximity of the basis ${}^1\mathcal{S}$ to the origin of the velocity space. The weight ρ is computed as two times the probability P that a random vector $\boldsymbol{x}$ obtained from the normal multivariate distribution $\mathcal{N}$ recovered from ${}^1\mathcal{S}$ (i.e. with barycenter $\boldsymbol{\mu}$ and covariance matrix Σ, see previous subsection) satisfies $\boldsymbol{\mu} \cdot \boldsymbol{x} < 0$. The probability P is given by

$$\mathrm{P}\big(\boldsymbol{\mu} \cdot \boldsymbol{x} < 0 \,|\, \boldsymbol{x} \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma)\big) = \frac{1}{2} - \frac{1}{2}\,\mathrm{erf}\left(\frac{\boldsymbol{\mu} \cdot \boldsymbol{\mu}}{\sqrt{2\boldsymbol{\mu}^{\mathsf{T}}\Sigma\boldsymbol{\mu}}}\right) \tag{3.32}$$

Algorithm 1: MISALIGNMENT function

```
    Input  : Configuration $\boldsymbol{q}$ and advance direction $\boldsymbol{v}$
    Output: Misalignment value $\eta$ of moving from $\boldsymbol{q}$ in the direction $\boldsymbol{v}$
1:  ${}^1\mathcal{S}(\boldsymbol{\mu}, \Sigma) \leftarrow$ FoSBasis($\boldsymbol{q}$)
2:  if ${}^1\mathcal{S} = \emptyset$ then
3:      $\eta \leftarrow 1$
4:  else
5:      Compute $\rho$ according to Eq. (3.33)
6:      Compute $\Phi_{\boldsymbol{\mu}}$ according to Eq. (3.30)
7:      Compute $\Phi_{\Sigma}$ according to Eq. (3.31)
8:      Compute $\eta$ according to Eq. (3.29)
9:  return $\eta$
```

where $\mathrm{erf}(x)$ is the error function. Then,

$$\rho = 1 - \mathrm{erf}\left(\frac{\boldsymbol{\mu} \cdot \boldsymbol{\mu}}{\sqrt{2\boldsymbol{\mu}^{\intercal}\Sigma\boldsymbol{\mu}}}\right) \tag{3.33}$$

Therefore, when the first-order basis ${}^1\mathcal{S}$ is exactly centered at the origin (i.e. $\|\boldsymbol{\mu}\| = 0$), half of the values of $\boldsymbol{x}$ satisfy the inequality and, hence, $\rho = 1$. As the first-order basis ${}^1\mathcal{S}$ gets away from the origin (i.e. $\|\boldsymbol{\mu}\| \to \infty$), only a reduced region of $\mathcal{N}$ satisfies the inequality and, therefore, $\rho \to 0$.

Note that the misalignment value η lies in the range $[0, 1]$ and it is small when the advance direction $\boldsymbol{v}$ is similar to the main synergy direction. When the difference between $\boldsymbol{v}$ and the main synergy direction increases, the misalignment increases.

The pseudocode of the MISALIGNMENT function is shown in Algorithm 1. First, the first-order synergy basis ${}^1\mathcal{S}$ associated with $\boldsymbol{q}$ is obtained with the function FoSBasis($\boldsymbol{q}$) (Line 1), which returns $\emptyset$ if no first-order basis is available for this particular $\boldsymbol{q}$. If ${}^1\mathcal{S} = \emptyset$ the misalignment η is set to the maximum value 1 (Line 2); otherwise, ρ is computed following Eq. (3.33) (Line 4), $\Phi_{\boldsymbol{\mu}}$ is computed according to Eq. (3.30) (Line 5), and Φ_{Σ} is computed according to Eq. (3.31) (Line 6). Finally, η is computed according to Eq. (3.29) (Line 7).

Now, since the path $\mathcal{P}$ is composed of a sequence of N consecutive configurations $\boldsymbol{q}_i$ connected by rectilinear motions (see Section 2.1), $\mathcal{H}$ as defined in Eq. (3.28) can be approximated as

$$\mathcal{H} \approx 1 - \sum_{i=1}^{N-1} \textsc{Misalignment}\left(\boldsymbol{q}_i, \boldsymbol{q}_{i+1} - \boldsymbol{q}_i\right) \frac{\|\boldsymbol{q}_{i+1} - \boldsymbol{q}_i\|}{L} \tag{3.34}$$

Therefore, a path with a high $\mathcal{H}$ value, is highly aligned in $\mathcal{C}$ with the human movements. Then, if the robot kinematic structure is anthropomorphic (and the more similar to the human operator, the better), the position and velocity of the robot wrists and the human wrists are similar.

The human-likeness index $\mathcal{H}$ depends on the ${}^{1}\mathcal{S}$ bases used. $\mathcal{H}$ can be tailored to any given particular task by using the corresponding ${}^{1}\mathcal{S}$ bases, and used for the evaluation of the human-likeness of the execution of that particular task. In this work, it is proposed the use of ${}^{1}\mathcal{S}$ obtained from natural *free-movements* of the operator while he/she freely moves both arms and hands in an unconstrained way (i.e. without performing any specific task) trying to cover the whole workspace in front of the body. There is no guarantee that the operator actually covers the whole workspace, but it is expected that he/she performs his/her most natural and evident movements. Note that the ${}^{1}\mathcal{S}$ bases depend on the mapping of the human movements to the robot configuration space, thus a mapping preserving the human-likeness should be used to make ${}^{1}\mathcal{S}$ really represent the human-like movements (e.g. the mapping presented in Subsection 3.1.4). In the next chapters, this index is used to numerically compare the human-likeness of the paths obtained using different planning algorithms.

4

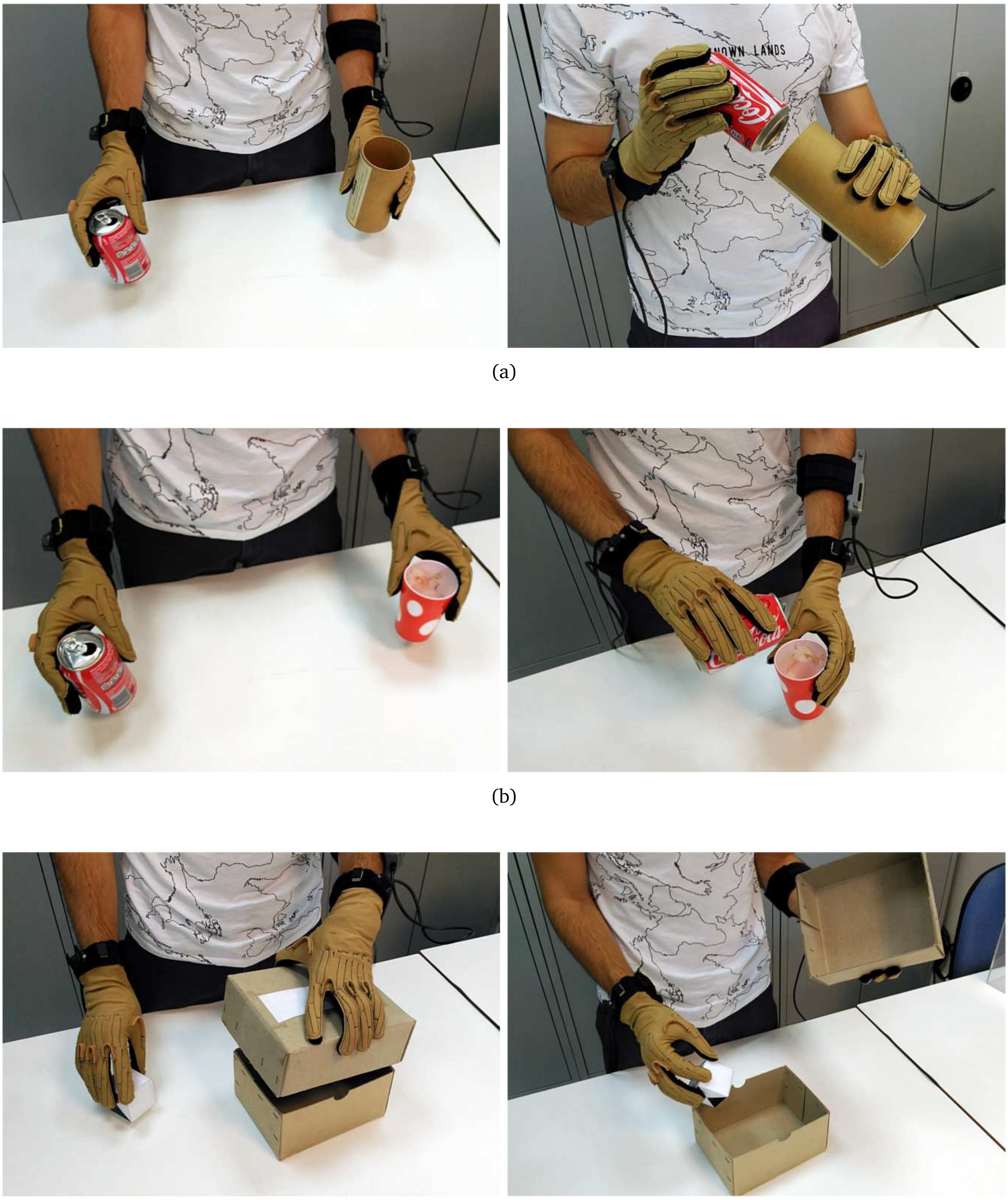
Planning human-like motions of anthropomorphic dual-arm robots

This chapter presents a synergy-based approach to plan the coordinated arm motions of anthropomorphic dual-arm robots in order to perform bimanual tasks. The proposed planning procedure is detailed and compared with other state-of-the-art planners in conceptual and application examples, both in simulated and in real executions, to demonstrate its applicability.

4.1 Motion capture

In this work, human motions are used as a reference to obtain human-like robot movements for anthropomorphic dual-arm manipulators. The focus is set in this section on the arm motions needed to perform bimanual tasks with objects standing on a table in front of the body. Three bimanual tasks have been selected as representative examples to serve as a testbed for the presented approach and, hence, have been demonstrated by different human operators while wearing a motion-capture system (described in Subsection 1.3.2). These demonstrated tasks are depicted in Fig. 4.1 and described below:

- The *Assembly* task in which the human operators must grasp, from a table in front of the body, a soda can with one hand and a cylindrical box with the other, and then move both objects to a pre-assembly pose that allows the insertion of the can into the box.



(a)

(b)

(c)

Figure 4.1: A human operator performing the demonstrations tasks while wearing the measurement equipment: (a) Assembly task, (b) Pouring task, (c) Box task. The left and right columns show, respectively, examples of the start and goal configurations.

- The *Pouring* task in which the human operators must grasp, from a table in front of the body, a soda can with one hand and a glass with the other, and then pour the drink into the grasped glass.
- The *Box* task in which the human operators must grasp, from a table in front of the body, a cube with one hand and simultaneously open a box with the other hand, and then introduce the cube into the box.

Besides, looking for a general and practical application of the approach, the movements of the operators freely moving both arms in an unconstrained way (i.e. without performing any specific task), trying to cover the whole natural workspace in front of the body, have also been captured. These movements are referred as the *Free-movement* task. This task is finished when the operator considers that she/he has covered the whole workspace. There is no guarantee that the operator actually covers the whole workspace, but it is expected that she/he performs her/his most natural and evident movements.

Each of the demonstrated tasks have been executed 10 times by 3 human operators moving only the arms. The human motions have been captured using the devices described in Subsection 1.3.2, generating representative sets of more than 10,000 samples per task. Pictures of the task goals have been shown to the operators to instruct them. The initial positions of the objects have been randomly located in given areas of the table, and the final goal positions have been those where the operators comfortably execute the task in a natural way. This gives variability to the data while preserving the essence of the task motions. Once the human movements have been captured, they have been mapped to the MADAR robotic system, described in Subsection 1.3.1, applying the mapping proposed in Subsection 3.1.4. Thus, for each captured human configuration, a robot configuration is obtained.

The described tasks are used as a testbed to test the validity of the proposed approach as well as to compare with other start-of-the-art motion-planning algorithms. Besides, the following non-demonstrated task is used to test the performance of the proposed approach in new scenarios never seen before (i.e. the movements of this task have not been learned):

- The non-demonstrated *Bottle* task that consists in grasping, from a table in front of the body, a half-full bottle with one hand and the bottle cap with the other hand and then tapping the bottle.

4.2 Motion planning using zero-order synergies

This Section proposes the use of zero-order synergies (see Subsection 3.1.1), obtained from human movements, to solve, efficiently and in a human-like fashion, bimanual tasks with anthropomorphic dual-arm robots. Besides, it proposes the use of the task-likeness introduced in Subsection 3.2.2 to select proper arm synergies for a given task, improving the planning performance and the resulting motion plan.

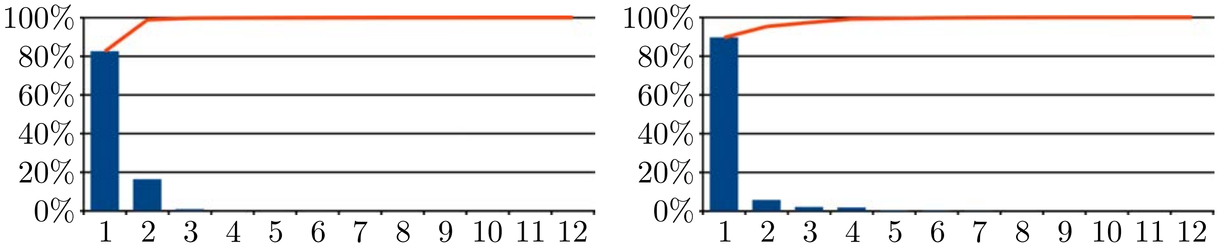
4.2.1 Motion analysis

A zero-order synergy basis ${}^{0}\mathcal{S}$ has been obtained for each of the tasks described in the previous section applying the method described in Subsection 3.1.4. Note that, for the Free-movement task, the set of mapped configurations has been artificially increased by adding to the task data set the original captured human configurations reflected in the human sagittal plane. In this way, the obtained synergies are symmetric, i.e. the same amount of sample variance is associated to each arm and, hence, the synergies are not adversely affected by the fact that the human operators involved in the data capture were mostly right-handed. It is important to highlight that the approach would mimic the demonstrated right-handedness of the human operators if the data set was not artificially increased.

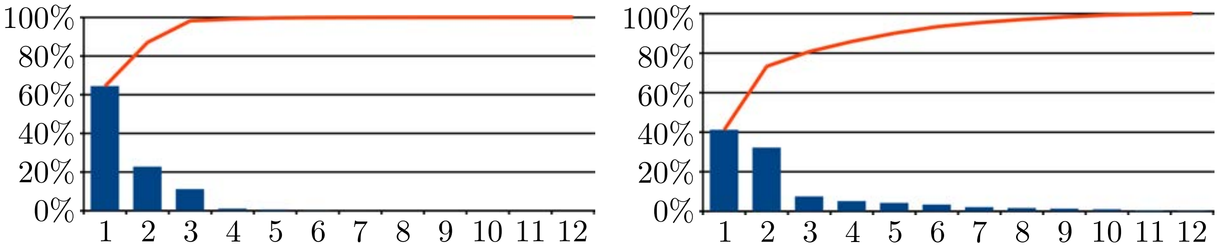
Table 4.1 shows the resulting variances along each synergy, which are graphically represented in Fig. 4.2. For the Pouring task, almost the 90% of the sample variance is associated with the first synergy, the second synergy has still some (low) relevance, but the other synergies have a very small dispersion. This means that the task executions were quite repetitive, and that the task could (almost) be done considering only the first synergy (which implies the coordinated movement of all the system joints, but since this is done in a fixed coordinated way it is equivalent to a single DOF). In the Assembly task there are two synergies with non-negligible variance while in the Box task there are three synergies with non-negligible variance. Regarding the Free-movement task, it can be seen that the first two synergies concentrate the main sample variance although the dispersion is still relevant along the first six or seven synergies. This is an expected effect since the operators have more freedom to perform the movements, which can also be seen in the total sample variance, clearly greater than in the other tasks.

Table 4.1: Sample variance and accumulated sample variance along the j-th synergy and total sample variance for each task. Bold values indicate the minimum number of synergies, taken in decreasing order of the associated sample variance, to cover at least the 95% of the sample variance.

j-th synergy	Assembly		Pouring		Box		Free-movement	
	var.	acc.	var.	acc.	var.	acc.	var.	acc.
1	82.6%	82.6%	89.6%	89.6%	64.4%	64.4%	41.2%	41.2%
2	**16.3%**	**98.9%**	**5.7%**	**95.3%**	22.7%	87.1%	32.2%	73.3%
3	0.8%	99.7%	2.1%	97.4%	**11.1%**	**98.2%**	7.5%	80.8%
4	0.1%	99.8%	1.8%	99.2%	1.0%	99.2%	5.1%	85.9%
5	0.1%	99.9%	0.3%	99.5%	0.5%	99.7%	4.2%	90.1%
6	0.0%	99.9%	0.2%	99.7%	0.2%	99.9%	3.3%	93.4%
7	0.0%	100%	0.1%	99.9%	0.0%	99.9%	**2.0%**	**95.4%**
8	0.0%	100%	0.1%	99.9%	0.0%	100%	1.6%	97.0%
9	0.0%	100%	0.0%	100%	0.0%	100%	1.3%	98.3%
10	0.0%	100%	0.0%	100%	0.0%	100%	0.9%	99.2%
11	0.0%	100%	0.0%	100%	0.0%	100%	0.4%	99.7%
12	0.0%	100%	0.0%	100%	0.0%	100%	0.3%	100%
Total	7.942 rad^2		7.569 rad^2		6.660 rad^2		9.994 rad^2	



(a) Assembly (b) Pouring



(c) Box (d) Free-movement

Figure 4.2: Accumulated sample variance versus the number of synergies.

4.2.2 Motion planning

As explained in Subsection 3.1.3, the arm synergies define a n-dimensional box B in the configuration space, where n is the number of arm DOFs of the robot. Here, $n = 12$ due to the used hardware, i.e. the MADAR robot described in Subsection 1.3.1. The box dimension can be decreased by using only $k < n$ synergies (picking them in decreasing order of the associated sample variance) such that k is the minimum value making the accumulated variance be above a confidence level of 95% (see bold values in Table 4.1).

Let $\boldsymbol{q}_{\text{start}} \in \mathcal{C}_{\text{start}}$ and $\boldsymbol{q}_{\text{goal}} \in \mathcal{C}_{\text{goal}}$ be respectively a start and a goal configuration of a given task to be carried out by the dual-arm system, where $\mathcal{C}_{\text{start}}$ and $\mathcal{C}_{\text{goal}}$ are the sets of collision-free configurations satisfying the constraints affecting the poses of the objects grasped by the hands at the initial and final states of the task, respectively. Let the pair $\boldsymbol{c} = \{\boldsymbol{q}_{\text{start}}, \boldsymbol{q}_{\text{goal}}\}$ be a motion planning query, composed of a start and a goal configuration. Then, in order to plan the robot motions to solve the given task, a large enough set of N_c queries is selected from $\mathcal{C}_{\text{start}}$ and $\mathcal{C}_{\text{goal}}$, satisfying the following conditions (see Fig. 4.3):

a) The motion planning query $\boldsymbol{c}$ is near the synergy subspace B_k used to solve the task, i.e. the distances from $\boldsymbol{q}_{\text{start}}$ and $\boldsymbol{q}_{\text{goal}}$ to the corresponding closest configurations in B_k, $\boldsymbol{q}'_{\text{start}}$ and $\boldsymbol{q}'_{\text{goal}}$ respectively, are below a given threshold.

b) The configurations $\boldsymbol{q}'_{\text{start}}$ and $\boldsymbol{q}'_{\text{goal}}$ as well as the rectilinear paths in the configuration space connecting $\boldsymbol{q}_{\text{start}}$ with $\boldsymbol{q}'_{\text{start}}$ and $\boldsymbol{q}_{\text{goal}}$ with $\boldsymbol{q}'_{\text{goal}}$ are collision-free.

A reduced set of n_c samples is selected from the N_c sampled queries. n_c is an arbitrarily predefined number selected according to the available computational capacity, and the selection is done such that the selected samples are the closest ones to B_k. Then, for each of the selected n_c queries, a planner instance (explained below) is run in the synergy subspace to find a path between the configurations $\boldsymbol{q}'_{\text{start}}$ and $\boldsymbol{q}'_{\text{goal}}$ corresponding to the query configurations $\boldsymbol{q}_{\text{start}}$ and $\boldsymbol{q}_{\text{goal}}$, respectively, as illustrated in Fig. 4.3. All the planner instances run in parallel and once a solution path is found by one of them, the motion planning is stopped and all the other threads are killed. Note that the motions are planned in B_k, which is more efficient than doing it in the whole configuration space, because on the one hand it is done in a lower dimensional space and, on the other hand, less self-collision occur (since the collision-free samples mapped from

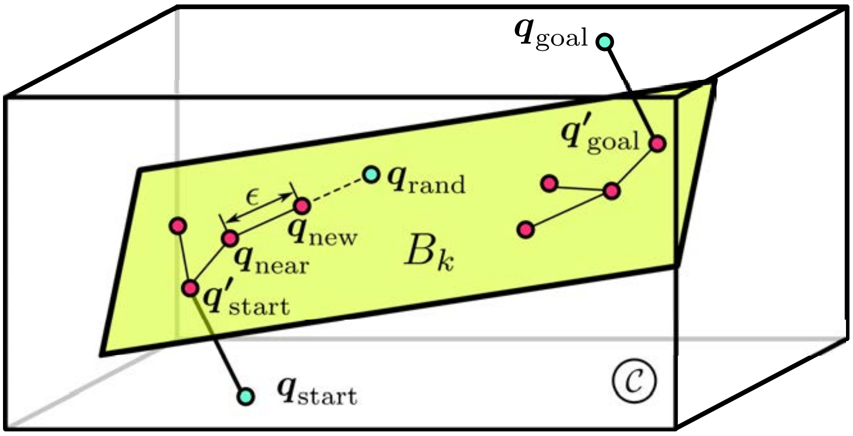


Figure 4.3: Conceptual representation of the configuration space $\mathcal{C}$ and the synergy subspace B_k, where the planner works, with the start and goal configurations, $\boldsymbol{q}_{\text{start}}$ and $\boldsymbol{q}_{\text{goal}}$, and their closest configurations in B_k, $\boldsymbol{q}'_{\text{start}}$ and $\boldsymbol{q}'_{\text{goal}}$.

the human motions lies there). In addition, the motions planned inside B_k retain a significant human-likeness of the human samples used to compute the subspace B_k (Suárez et al., 2015).

Note that the motions between $\boldsymbol{q}_{\text{start}}$ and $\boldsymbol{q}'_{\text{start}}$ and between $\boldsymbol{q}'_{\text{goal}}$ and $\boldsymbol{q}_{\text{goal}}$ are collision-free, by definition. Then, the configurations $\boldsymbol{q}'_{\text{start}}$ and $\boldsymbol{q}'_{\text{goal}}$ are connected using the algorithm RRT-Connect (Kuffner and LaValle, 2000), which has been widely used in motion planning since it obtains good results even in problems with cluttered environments and robots with a lot of DOFs. This planner, outlined in Algorithm 2, maintains two trees of configurations, $\mathcal{T}_{\text{A}}$ and $\mathcal{T}_{\text{B}}$, one rooted at the start configuration $\boldsymbol{q}_{\text{start}}$ (Line 1) and the other rooted at the goal configuration $\boldsymbol{q}_{\text{goal}}$ (Line 2). In each iteration, one of the trees grows towards a random configuration $\boldsymbol{q}_{\text{rand}}$ (Line 4), reaching a new configuration $\boldsymbol{q}_{\text{new}}$ (Line 5), using the function EXTEND explained below. Then, the connection between $\boldsymbol{q}_{\text{new}}$ and the other tree is attempted, using the function CONNECT explained below. If the connection of the trees succeeds, the planning process stops since a collision-free path connecting $\boldsymbol{q}_{\text{start}}$ and $\boldsymbol{q}_{\text{goal}}$ already exists through the trees. Otherwise, the trees swap their roles (Line 7) and the process is repeated until a solution is found or some termination condition is satisfied (Line 3), e.g. surpassing a limit of planning time, number of iterations or size of the configuration trees $\mathcal{T}_{\text{A}}$ and $\mathcal{T}_{\text{B}}$.

The function EXTEND, described in Algorithm 3, extends a given configuration tree $\mathcal{T}$ from $\boldsymbol{q}_{\text{near}}$, the configuration in the tree closest to $\boldsymbol{q}_{\text{rand}}$ (Line 1), by taking a single step of a maximum length ϵ towards $\boldsymbol{q}_{\text{rand}}$ (Line 2), reaching $\boldsymbol{q}_{\text{new}}$. Only if the rectilinear segment connecting $\boldsymbol{q}_{\text{near}}$

Algorithm 2: RRTConnect

```
Input  : Start configuration $\boldsymbol{q}_{\text{start}} \in \mathcal{C}_{\text{free}}$ and goal configuration $\boldsymbol{q}_{\text{goal}} \in \mathcal{C}_{\text{free}}$
Output: Collision path $\mathcal{P} \in \mathcal{C}_{\text{free}}$ connecting $\boldsymbol{q}_{\text{start}}$ and $\boldsymbol{q}_{\text{goal}}$

$\mathcal{T}_{\text{A}} \leftarrow$ InitTree($\boldsymbol{q}_{\text{start}}$)
$\mathcal{T}_{\text{B}} \leftarrow$ InitTree($\boldsymbol{q}_{\text{goal}}$)
while not StopCriteria($\mathcal{T}_{\text{A}}$, $\mathcal{T}_{\text{B}}$) do
    $\boldsymbol{q}_{\text{rand}} \leftarrow$ RandConf( )
    $\boldsymbol{q}_{\text{new}} \leftarrow$ Extend($\mathcal{T}_{\text{A}}, \boldsymbol{q}_{\text{rand}}$)
    if $\boldsymbol{q}_{\text{new}} \neq \emptyset$ and Connect($\mathcal{T}_{\text{B}}, \boldsymbol{q}_{\text{new}}$) then
        return Path($\mathcal{T}_{\text{A}}, \mathcal{T}_{\text{B}}$)
    Swap($\mathcal{T}_{\text{A}}, \mathcal{T}_{\text{B}}$)
return $\emptyset$
```

Algorithm 3: Extend

```
Input  : Tree of configurations $\mathcal{T}$ and configuration $\boldsymbol{q}_{\text{target}}$
Output: Configuration $\boldsymbol{q}_{\text{new}}$

$\boldsymbol{q}_{\text{near}} \leftarrow$ NearestConf($\mathcal{T}, \boldsymbol{q}_{\text{target}}$)
$\boldsymbol{q}_{\text{new}} \leftarrow \boldsymbol{q}_{\text{near}} + \min\big(\epsilon, \|\boldsymbol{q}_{\text{target}} - \boldsymbol{q}_{\text{near}}\|\big) \frac{\boldsymbol{q}_{\text{target}} - \boldsymbol{q}_{\text{near}}}{\|\boldsymbol{q}_{\text{target}} - \boldsymbol{q}_{\text{near}}\|}$
if CollisionFree($\boldsymbol{q}_{\text{near}}, \boldsymbol{q}_{\text{new}}$) then
    AddSegment($\mathcal{T}, \boldsymbol{q}_{\text{near}}, \boldsymbol{q}_{\text{new}}$)
    return $\boldsymbol{q}_{\text{new}}$
else
    return $\emptyset$
```

Algorithm 4: Connect

```
Input  : Tree of configurations $\mathcal{T}$ and bridge configuration $\boldsymbol{q}_{\text{bridge}}$
Output: true, if $\mathcal{T}$ suceeds in reaching $\boldsymbol{q}_{\text{bridge}}$, and false, if $\mathcal{T}$ fails in reaching $\boldsymbol{q}_{\text{bridge}}$

do
    $\boldsymbol{q}_{\text{new}} \leftarrow$ Extend($\mathcal{T}, \boldsymbol{q}_{\text{bridge}}$)
    if $\boldsymbol{q}_{\text{new}} = \emptyset$ then
        return false
while $\boldsymbol{q}_{\text{new}} \neq \boldsymbol{q}_{\text{bridge}}$
return true
```

and $\boldsymbol{q}_{\text{new}}$ is collision-free, the segment is added to the tree and $\boldsymbol{q}_{\text{new}}$ is returned (Lines 4-5). Otherwise, $\emptyset$ is returned (Line 6). The function CONNECT, outlined in Algorithm 4, attempts to connect a given configuration tree $\mathcal{T}$ to a given configuration $\boldsymbol{q}_{\text{bridge}}$, which will be the bridge between the two trees, if their connection succeeds (note that this function is called with $\boldsymbol{q}_{\text{new}}$ as argument in the RRT-Connect algorithm). This function returns `true` or `false` whether it succeeded or not, respectively. The connection consists on forcing the tree $\mathcal{T}$ to take successive steps towards $\boldsymbol{q}_{\text{new}}$ (Line 2), by calling the function EXTEND, until $\boldsymbol{q}_{\text{new}}$ is reached (Line 4) or the function EXTEND returns $\emptyset$ (Line 3), which means that a collision was found.

It is important to highlight that the original RRT-Connect algorithm samples the random configuration $\boldsymbol{q}_{\text{rand}}$ in the complete $\mathcal{C}$ space (Line 4 in Algorithm 2). Nevertheless, in order to plan the motions inside the synergy box B_k, $\boldsymbol{q}_{\text{rand}}$ is sampled here in B_k. In this case, the start and goal configurations, which the planner is provided with, must lie already in B_k and, consequently, $\boldsymbol{q}'_{\text{start}}$ and $\boldsymbol{q}'_{\text{goal}}$ are used.

4.2.3 Approach validation

The proposed planning procedure has been tested in the *Assembly*, *Pouring* and *Box* tasks described in Section 4.1. For each of theses tasks, the planning procedure has been run using:

(a) The whole 12-dimensional dual-arm configuration space (i.e. using the original algorithm RRT-Connect, planning in the complete $\mathcal{C}$ space).

(b) The k-dimensional subspace B_k of the planned task (i.e. planning the Assembly task, for instance, using the subspace B_k of the Assembly task itself, and so on successively).

(c) The k-dimensional subspace B_k of the other demonstrated tasks (i.e. planning the Assembly task, for instance, using the subspaces B_k of the Pouring and Box tasks but also the subspace B_k of the Free-movement task, and so on successively).

Notice that while the full dimension of the problem is always $n = 12$ (the total number of DOFs of the two arms), the reduced dimension k has different values depending on the distribution of the sample variance among the synergies, see Subsection 3.1.3. Hence, it is obtained $k = 2$ for the Assembly and Pouring tasks, $k = 3$ for the Box task and $k = 7$ for the Free-movement task (see Table 4.1). Besides, for each task, $n_c = 10$ different task queries have been selected from an initial set of $N_c = 100$, see previous subsection.

For the Assembly task, the set of N_c goals was generated satisfying the geometrical constraints necessary for the assembly of the soda can into the cylindrical box (see Fig. 4.1-a). This results in a 7-dimensional goal space, i.e. at the pre-assembly moment the cylindrical box can be in different positions and orientations (six DOFs) and the soda can is allowed to rotate around its axis while satisfying the pre-assembly pose constraints (one additional DOF).

For the Pouring task, the goal space is also 7-dimensional, the glass must be vertical and resting on the table (three DOFs) and the soda can must have the opening exactly above the glass within a predefined height range (one additional DOF) and with any orientation (three additional DOFs). Notice that the rotation about the can axis impose constraints on the can grasping, and the proper rotation about an horizontal axis depends on the quantity of liquid in the can, but for illustrative purposes in this work this degree of freedom is simply considered by imposing a small predefined rotation range (see Fig. 4.1-b).

For the Box task, the goal space is 6-dimensional, the box can be in any position and orientation on the table (three DOFs) and the cube must be placed at a predefined position inside the box with any orientation (three additional DOFs). It is assumed that the left and right hands are already grasping the box cover and the cube respectively at the start configuration (see Fig. 4.1-c). Therefore, the start configuration depends on the position of the box for the left hand while for the right hand it is fixed. Similarly, the goal configuration depends on the position of the box for the right hand while for the left hand it is fixed.

The proposed planning procedure has been encapsulated within The Kautham Project (see Subsection 1.4.1). For each task, an instance of the planner was run in parallel for each sampled motion-planning query, stopping the motion planner when a valid solution path was found by one of the instances. If the planner instances could not solve the task within a predefined time restriction of 100 seconds the run is considered as a failure. The manipulated objects are considered to be firmly grasped. Hence, in simulation, they are part of the last arm links and, in the real experimentation, the hand controller and the system detecting the objects are considered robust enough to ensure such assumption. Table 4.2 shows the average planning results obtained after 100 executions for each case of each task, running in a 3.40-GHz Intel i7-3770, 8-GB RAM PC. The experimental results demonstrate that the use of synergies:

- Increases significantly the probability of obtaining collision-free configurations (fewer self-

Table 4.2: Average results of the motion planning when the tasks are solved without synergies, with task-specific synergies and with other tasks synergies.

Solved task	Assembly				
with synergies of	None	Assembly	Pouring	Box	Free-movement
# Synergies	0	2	2	3	7
Space dimension	12	2	2	3	7
Success rate	100%	100%	100%	100%	100%
Planning time	8.293 s	4.205 s	6.209 s	7.342 s	6.654 s
# Tree nodes	6.27	4.02	4.49	5.04	4.95
Solution length†	3.761 rad	2.112 rad	2.343 rad	2.954 rad	2.691 rad
Valid motion rate*	80.34%	100%	100%	100%	97.77%
# Collision checks	65.28	22.01	35.02	45.08	37.71
Path in B_k §	0%	89.91%	89.83%	90.09%	90.62%

Solved task	Pouring				
with synergies of	None	Assembly	Pouring	Box	Free-movement
# Synergies	0	2	2	3	7
Space dimension	12	2	2	3	7
Success rate	100%	100%	100%	100%	100%
Planning time	12.731 s	8.466 s	4.099 s	6.525 s	10.300 s
# Tree nodes	8.50	6.12	4.00	4.85	6.93
Solution length†	5.835 rad	4.012 rad	2.240 rad	2.608 rad	4.859 rad
Valid motion rate*	84.64%	100%	100%	100%	100%
# Collision checks	161.76	61.43	19.17	37.21	72.92
Path in B_k §	0%	89.49%	87.43%	85.53%	92.07%

Solved task	Box				
with synergies of	None	Assembly	Pouring	Box	Free-movement
# Synergies	0	2	2	3	7
Space dimension	12	2	2	3	7
Success rate	100%	100%	100%	100%	100%
Planning time	14.314 s	9.388 s	8.323 s	4.622 s	10.336 s
# Tree nodes	11.70	8.05	6.39	4.04	8.84
Solution length†	9.186 rad	5.868 rad	4.773 rad	3.603 rad	6.624 rad
Valid motion rate*	81.75%	95.80%	100%	100%	98.78%
# Collision checks	212.76	85.16	64.61	30.49	123.66
Path in B_k §	0%	93.83%	81.11%	95.57%	94.93%

† Evaluated as the summation of the joints movements in radians

* Proportion of iterations in which no collisions occur and the tree actually grows

§ Percentage of the whole path contained in B_k, i.e. without considering the segments from $\boldsymbol{q}_{\text{start}}$ to $\boldsymbol{q}'_{\text{start}}$ and from $\boldsymbol{q}'_{\text{goal}}$ to $\boldsymbol{q}_{\text{goal}}$ (see Fig. 4.3).

Table 4.3: Likeness $\mathcal{L}({}^0\mathcal{S}_A,\ {}^0\mathcal{S}_B)$ between the considered tasks.

Tasks	Assembly	Pouring	Box	Free-movement
Assembly	1	0.1081	0.0114	0.6104
Pouring	0.1081	1	0.0035	0.5699
Box	0.0114	0.0035	1	0.6829
Free-movement	0.6104	0.5699	0.6829	1

collisions occur), thus reducing the planning time.

- Allows a reduction of the dimension of the search space, which reduces the number of nodes and edges of the tree and hence reduces the memory requirements.

It must be highlighted that these aspects are more pronounced when task-specific synergies are used (i.e. case b), but, for general applications, the utility of the synergies obtained with the Free-movement task is still relevant, since they also improve the results compared with planning without using synergies.

Table 4.3 shows the likeness $\mathcal{L}({}^0\mathcal{S}_A,\ {}^0\mathcal{S}_B)$ between the demonstration tasks obtained with the procedure presented in Subsection 3.2.1. It can be seen that the Free-movement task is more similar to all the other tasks, while these tasks are more dissimilar between them.

In order to provide a graphical representation of the likeness between the tasks, a proximity index $\mathcal{D}({}^0\mathcal{S}_A,\ {}^0\mathcal{S}_B)$ is defined as

$$\mathcal{D}({}^0\mathcal{S}_A,\ {}^0\mathcal{S}_B) = 1 - \mathcal{L}({}^0\mathcal{S}_A,\ {}^0\mathcal{S}_B) \in [0,\ 1] \tag{4.1}$$

Even when $\mathcal{D}$ is not a real *distance*, since it does not satisfy the triangle inequality (i.e. the distance between two points must be the shortest distance along any path between them), it is still possible to represent the synergy bases of the four considered tasks as points in a 3D Euclidean space such that the Euclidean distances $d({}^0\mathcal{S}_i,\ {}^0\mathcal{S}_j)$ between these points minimize the maximum relative error with respect to the corresponding proximity indices $\mathcal{D}$, i.e minimizing

$$\max_{i \neq j} \left(\frac{d({}^0\mathcal{S}_i,\ {}^0\mathcal{S}_j) - \mathcal{D}({}^0\mathcal{S}_i,\ {}^0\mathcal{S}_j)}{\mathcal{D}({}^0\mathcal{S}_i,\ {}^0\mathcal{S}_j)} \right) \tag{4.2}$$

By doing this, it results that the four points are approximately coplanar (the Free-movement task

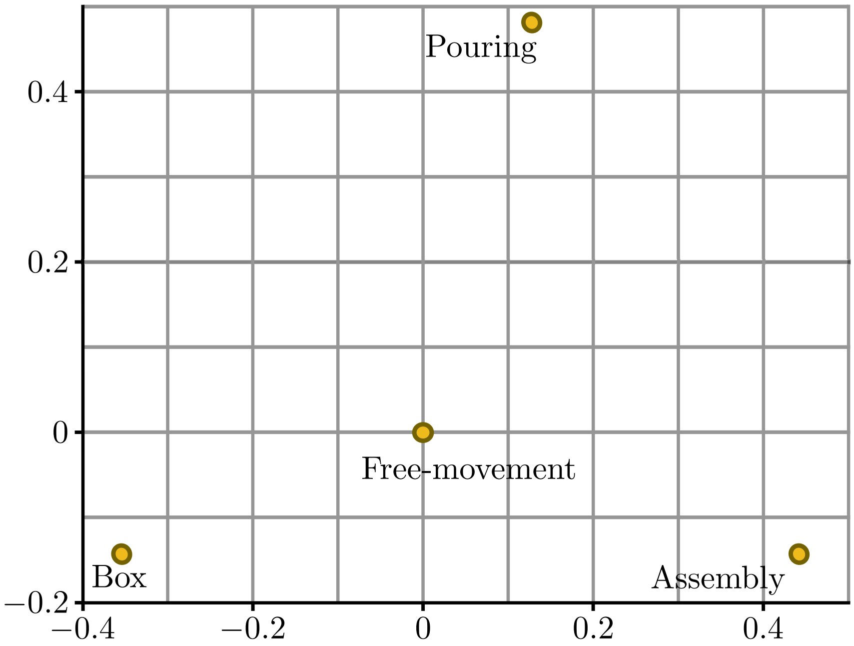


Figure 4.4: Representation of the demonstrated tasks in a 2-dimensional space based on the $\mathcal{L}$ values between the tasks. The origin has been set in the Free-movement task and the axes orientation was arbitrarily selected. The distance between two tasks is, by definition, constrained to the $[0, 1]$ interval, therefore the axes are unit-less.

is at 0.0001 distance units far from the plane defined by the other three tasks, while the distance between any two tasks is higher by about three orders of magnitude), thus the synergies can actually be represented in a 2D Euclidean space, as shown in Fig. 4.4. This representation gives a clear intuitive view of the relation between the tasks by approximating the likeness $\mathcal{L}$ with the distances between them. Note that the Free-movement task lies inside the triangle defined by the other tasks. This confirms the idea of using this synergy basis for general applications.

The likeness $\mathcal{L}$ can be used to classify the tasks into families or clusters, and it is expected that solving a task with the synergies of a more alike task would result in a faster and better motion planning.

In order to test the real usefulness of $\mathcal{L}$, it has been checked whether there is a relation between t_{A_B}, the average time employed to solve the motion planning of task A using the synergies of task B (see Table 4.2), and $\mathcal{L}({}^0\mathcal{S}_A, {}^0\mathcal{S}_B)$, the likeness between tasks A and B (see Table 4.3). Since the tasks have different degrees of difficulty they inherently require different times t, thus the t values were normalized to $\hat{t} \in [0, 1]$ using the minimum time $t_{A_{\text{min}}}$ and the

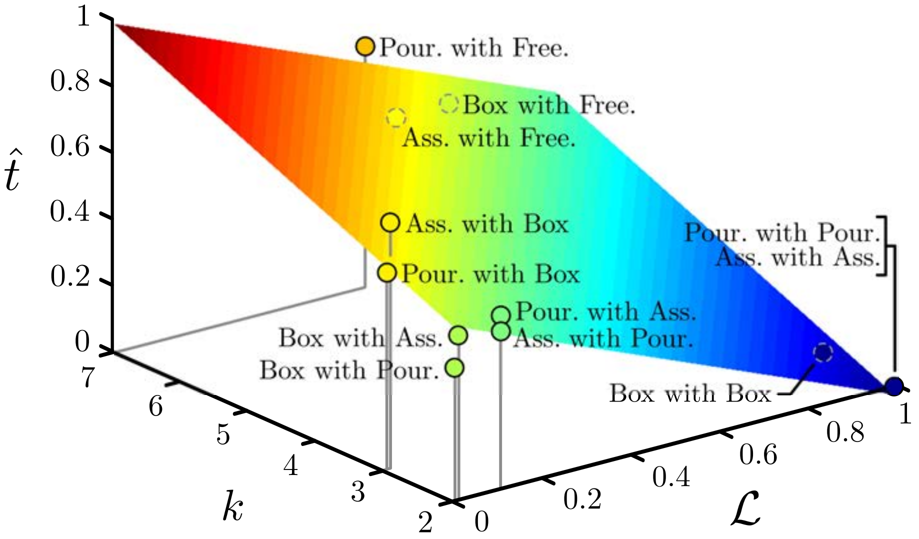


Figure 4.5: Normalized time $\hat{t} \in [0, 1]$ as a function of the likeness $\mathcal{L} \in [0, 1]$ and the number k of synergies used, for the considered tasks solved with the own synergies and the ones of the other tasks (Ass. = Assembly, Pour. = Pouring, Free = Free-movement). It is also shown the fitted plane $\hat{t} = \kappa_0 + \kappa_{\mathcal{L}}\mathcal{L} + \kappa_k k$.

maximum time $t_{A_{\text{max}}}$ needed to solve a task A. Then, the values $\hat{t}_{A_B}$ were computed as:

$$\hat{t}_{A_B} = \frac{t_{A_B} - t_{A_{\text{min}}}}{t_{A_{\text{max}}} - t_{A_{\text{min}}}} \tag{4.3}$$

It must be highlighted that for all the tasks, the minimum and maximum times are obtained respectively when the task-specific synergies are used and when no synergies are used. Furthermore, since the dimension of the search space plays a very important role, the number k of synergies used in the motion planning is also considered in the study of $\hat{t}$, i.e. $\hat{t} = \hat{t}(\mathcal{L}, k)$.

Using the data in Table 4.2 and Table 4.3, Fig. 4.5 shows the normalized time $\hat{t}$ as a function of the likeness $\mathcal{L}$ and the space dimension k for the three tasks solved with their task-specific synergies, the synergies of the other two tasks and the synergies of the Free-movement task. A plane $\hat{t} = \kappa_0 + \kappa_{\mathcal{L}}\mathcal{L} + \kappa_k k$ has been fitted to the values of $\hat{t}$ as a first-order approximation. As it was expected the coefficient $\kappa_{\mathcal{L}}$ is negative, i.e. $\hat{t}$ decreases with $\mathcal{L}$. In addition, κ_k is positive. Similar results are obtained when, instead of using the average time $\hat{t}$, the average path length or the average number of collisions are plotted. This verifies the hypothesis that using the synergies of an alike task according to $\mathcal{L}$ produces better motion planning results (i.e. less planning time and shorter paths) for the same value of k.

Table 4.4: Likeness $\mathcal{L}({}^{0}\hat{\mathcal{S}}_{V},\, {}^{0}\mathcal{S}_{A})$ between the virtual and the demonstrated synergy bases[†].

Tasks	Assembly	Pouring	Box	Free-movement
Virtual Assembly	**0.2416**	0.1916	0.1021	0.4966
Virtual Pouring	0.2079	**0.3448**	0.0645	0.4739
Virtual Box	0.0999	0.1005	**0.2146**	0.5132
Virtual Bottle	**0.4020**	0.1082	0.1254	0.6559

[†] The likeness values between each virtual synergy basis and its closest demonstrated synergy basis are marked in bold, regardless of the basis of the Free-movement task (i.e. the synergy basis with which the virtual synergy basis was obtained).

As an application to the found results, consider that there is a new task to be solved but it has not been previously demonstrated, so a task-specific synergy basis is not available for it. The procedure proposed to improve the motion planning for this new task is:

(a) Use of the synergies of the Free-movement task to obtain a first plan that solves this non-demonstrated task.

(b) Run a PCA using the samples of this first plan to obtain a new set of synergies, i.e. obtaining a synergy basis ${}^{0}\hat{\mathcal{S}}_{V}$ from only one virtual execution.

(c) Search for the synergy basis ${}^{0}\mathcal{S}_{A}$ most alike to ${}^{0}\hat{\mathcal{S}}_{V}$. This is done by looking for the basis ${}^{0}\mathcal{S}_{A}$ that maximizes $\mathcal{L}({}^{0}\hat{\mathcal{S}}_{V},\, {}^{0}\mathcal{S}_{A})$ among all the demonstrated tasks.

(d) Use of ${}^{0}\mathcal{S}_{A}$ in a new motion planning process.

In order to check the validity of this procedure, a PCA was run on the samples obtained from one motion plan of each task generated using the Free-movement synergies. This generates a new synergy basis for each task, and the likeness between this basis and the synergy bases obtained from the human demonstrations is given in Table 4.4. In all the cases the most alike synergy basis is the task-specific one, e.g. the likeness index between the virtual Assembly and the Assembly tasks is higher than the likeness index between the virtual Assembly and the other tasks, disregarding the Free-movement task, that was used to generate the virtual synergies. The experimental results show that this approach improves the planning process. Nevertheless, since there are random searches in the whole process, the improvement can not be always ensured.

For illustrative purposes, the introduced procedure was also used to plan the motions of the non-demonstrated *Bottle* task described in Section 4.1. Similarly to the Assembly task, the

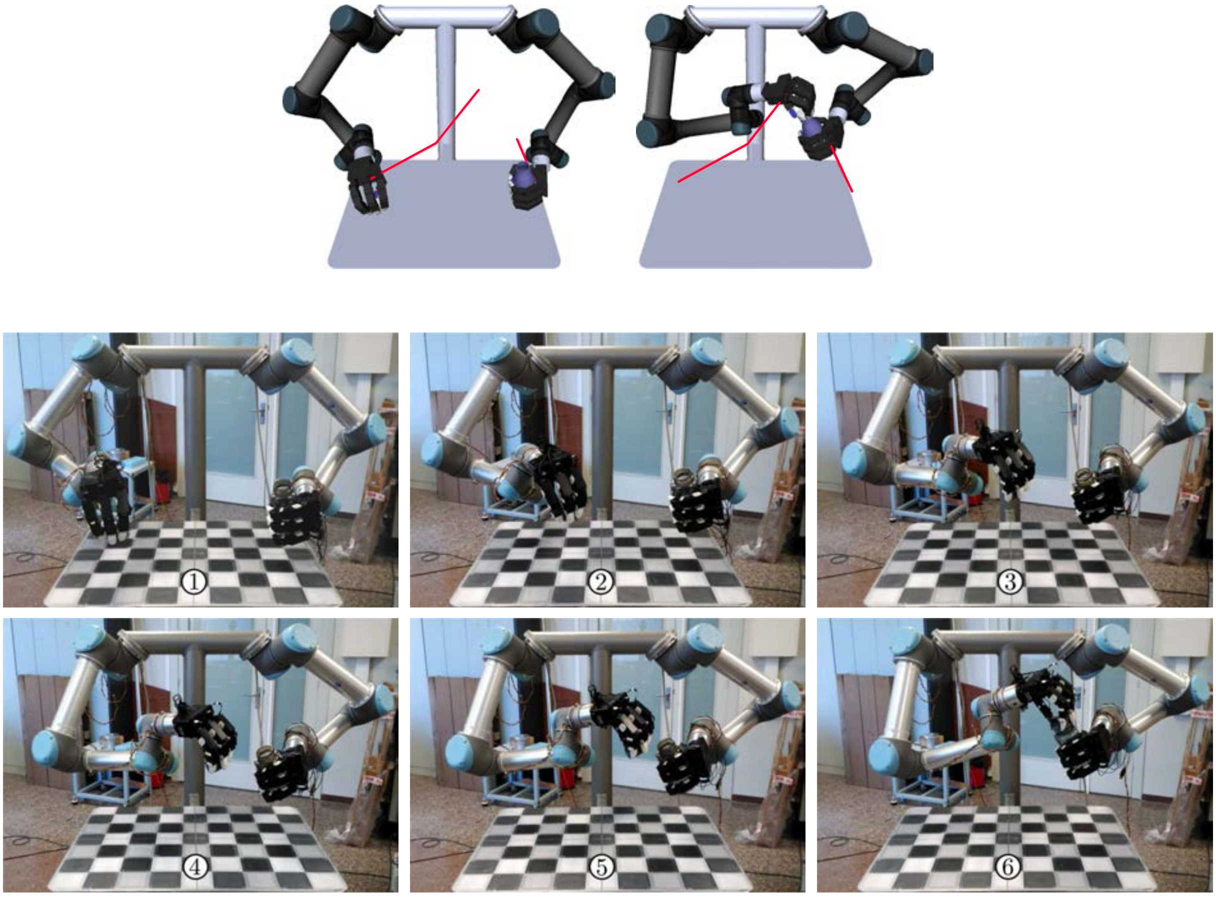


Figure 4.6: Example of a solution path for the Bottle task obtained using the first two synergies of the Assembly task (i.e. its closest task): start and goal configurations in the simulation environment, showing the planned paths (top); and snapshots of the path execution with the real dual-arm system (bottom).

goal space is also 7-dimensional. However, in this case the goal pose of the assembled objects, i.e. the bottle tapped by the cap, is highly restricted in orientation to avoid pouring the liquid. Following the proposed procedure, a virtual synergy basis for the Bottle task was obtained using the Free-movement synergies and the likeness indices with respect to the demonstrated synergy bases were computed. The Assembly task results to be the most alike task, as it was intuitively expected (see Table 4.4). Finally, the motion planning is solved again using the Assembly synergies. The solutions obtained with this procedure results in movements of the robotic arms that have a natural appearance, even though the motion planning was solved using the synergies of a different task. Snapshots of an instance of the obtained solution paths are shown in Fig. 4.6.

4.3 Motion planning using synergies over potential fields

As seen in the previous section, the zero-order synergies are useful to obtain robot paths composed of configurations that are human-like, since they lie in the same subspace of the configurations mapped from real human movements. Nevertheless, these results can be improved if the planned path does not only lie in the same subspace of the demonstrated movements but also runs close to the demonstrated paths. These can be obtained if the planned motions are steered to follow a potential field based on one or a few number of demonstrated paths. Hence, in this section a reduced set of the movements captured from the tasks described in Section 4.1 are used to generate a potential field, that guides the motion planning, and to compute zero-order synergies, that accelerate the motion planning (the analysis of the obtained zero-order synergies is not done here, as they have already been analysed throughly in Subsection 4.2.1).

4.3.1 Motion planning

Potential field techniques have been widely used to guide the robot motions by constructing features that attract the robot towards the final configuration and take it away from static obstacles present in the scene, combining different fields (Latombe, 1991). Here, this approach is used within the motion planning to guide the tree growth towards real human movements, and, hence, obtain human-like motions for dual-arm robots.

The used potential-field is generated in the robot configuration space $\mathcal{C}$ and it is composed of several potential fields. The demonstration paths and the goal configuration $\boldsymbol{q}_{\text{goal}}$ generate attractive potential fields. Besides, the obstacles generate potential fields that repulse the robotic arms while the arms also repel each other. Therefore, to compute the potential-field value $V(\boldsymbol{q})$ of a given robot configuration $\boldsymbol{q}$, let first:

$\lambda \geq 0$ and $\sigma \geq 0$ be, respectively, the strength and the diffusion parameters of each potential field forming $V(\boldsymbol{q})$;

$\mathcal{P}_i \in P$ be the i-th path of the given set of demonstration paths P, obtained from the mapping of the human movements and projected onto $\mathcal{C}$ (see Subsection 3.1.4);

$\mathcal{O}_j \in O$ be the j-th obstacle of the set of obstacles O with which the dual-arm system can collide;

$\mathrm{d}(\boldsymbol{q}, \mathcal{P}_i)$ be the minimum distance in $\mathcal{C}$, between the configuration $\boldsymbol{q}$ and the demonstration path $\mathcal{P}_i$;

$\mathrm{d}(\mathcal{L}_{\boldsymbol{q}}, \mathcal{R}_{\boldsymbol{q}})$ be the minimum distance in the workspace between both arms of the robotic system, when the dual-arm robot configuration is $\boldsymbol{q}$;

$\mathrm{d}(\mathcal{L}_{\boldsymbol{q}}, \mathcal{O}_j)$ and $\mathrm{d}(\mathcal{R}_{\boldsymbol{q}}, \mathcal{O}_j)$ be the minimum distances in the workspace between the obstacle $\mathcal{O}_j$ and the left and the right arms, respectively, when the dual-arm robot configuration is $\boldsymbol{q}$.

Then, the resultant potential-field value $V(\boldsymbol{q})$ is defined as the sum of four potential fields values:

$$V(\boldsymbol{q}) = V_{\text{goal}}(\boldsymbol{q}) + V_{\text{paths}}(\boldsymbol{q}) + V_{\text{arms}}(\boldsymbol{q}) + V_{\text{obs}}(\boldsymbol{q}) \tag{4.4}$$

where:

$$V_{\text{goal}}(\boldsymbol{q}) = \lambda_{\text{goal}} \left(1 - e^{-\sigma_{\text{goal}} \|\boldsymbol{q} - \boldsymbol{q}_{\text{goal}}\|^2}\right) \tag{4.5}$$

is the potential-field value of the configuration $\boldsymbol{q}$ regarding the attractive potential field of the goal configuration $\boldsymbol{q}_{\text{goal}}$ (i.e. the closer are $\boldsymbol{q}$ and $\boldsymbol{q}_{\text{goal}}$, the smaller is V_{goal});

$$V_{\text{paths}}(\boldsymbol{q}) = \sum_{i=1}^{|P|} \frac{\lambda_i}{|P|} \left(1 - e^{-\sigma_i \, \mathrm{d}(\boldsymbol{q}, \mathcal{P}_i)^2}\right) \tag{4.6}$$

is the potential-field value of the configuration $\boldsymbol{q}$ regarding the attractive potential fields generated by all the demonstration paths $\mathcal{P}_i \in P$, i.e. V_{paths} decreases when $\boldsymbol{q}$ gets closer to P (it must be noted that, since $\mathcal{P}_i$ and $\boldsymbol{q}$ are both expressed in $\mathcal{C}$, $\mathrm{d}(\boldsymbol{q}, \mathcal{P}_i)$ is simply the minimum Euclidean distance in $\mathcal{C}$ between $\boldsymbol{q}$ and the rectilinear segments representing $\mathcal{P}_i$);

$$V_{\text{arms}}(\boldsymbol{q}) = \lambda_{\text{arms}} e^{-\sigma_{\text{arms}} \, \mathrm{d}(\mathcal{L}_{\boldsymbol{q}}, \mathcal{R}_{\boldsymbol{q}})^2} \tag{4.7}$$

is the potential-field value of the configuration $\boldsymbol{q}$ regarding the repulsive potential field between the arms of the robotic system, i.e. V_{arms} grows if the arms get closer; and

$$V_{\text{obs}}(\boldsymbol{q}) = \sum_{j=1}^{|O|} \frac{\lambda_j}{|O|} \left(e^{-\sigma_j \, \mathrm{d}(\mathcal{L}_{\boldsymbol{q}}, \mathcal{O}_j)^2} + e^{-\sigma_j \, \mathrm{d}(\mathcal{R}_{\boldsymbol{q}}, \mathcal{O}_j)^2}\right) \tag{4.8}$$

is the potential-field value of the configuration $\boldsymbol{q}$ regarding the repulsive potential fields of all the obstacles $\mathcal{O}_j \in O$, if either the left arm or the right arm of the robotic system gets close to any

$\mathcal{O}_j$, then V_{obs} increases. It is important to highlight that in order to speed up the computation of $V(\boldsymbol{q})$, the distances between the robotic arms and the obstacles (i.e. $\mathrm{d}(\mathcal{L}_{\boldsymbol{q}}, \mathcal{R}_{\boldsymbol{q}})$, $\mathrm{d}(\mathcal{L}_{\boldsymbol{q}}, \mathcal{O}_j)$ and $\mathrm{d}(\mathcal{R}_{\boldsymbol{q}}, \mathcal{O}_j)$) are computed using a simplified model of the robot and the obstacles based on planes, spheres and capsules.

In order to be able to compare different robot motions, the $c(\boldsymbol{q}_{\text{i}}, \boldsymbol{q}_{\text{f}})$ motion cost is defined. It computes how well the rectilinear motion between the two given configurations $\boldsymbol{q}_{\text{i}}$ and $\boldsymbol{q}_{\text{f}}$ follows the demonstration paths as the linear combination of three other costs c_{P}, c_{I} and c_{D} with given respective positive weights ω_{P}, ω_{I} and ω_{D}:

$$c(\boldsymbol{q}_{\text{i}}, \boldsymbol{q}_{\text{f}}) = \omega_{\text{P}} \underbrace{\|\boldsymbol{q}_{\text{f}} - \boldsymbol{q}_{\text{i}}\|}_{c_{\text{P}}(\boldsymbol{q}_{\text{i}}, \boldsymbol{q}_{\text{f}})} + \omega_{\text{I}} \underbrace{\int_{\boldsymbol{q}_{\text{i}}}^{\boldsymbol{q}_{\text{f}}} V(\boldsymbol{q})\, \mathrm{d}\boldsymbol{q}}_{c_{\text{I}}(\boldsymbol{q}_{\text{i}}, \boldsymbol{q}_{\text{f}})} + \omega_{\text{D}} \underbrace{\int_{\boldsymbol{q}_{\text{i}}}^{\boldsymbol{q}_{\text{f}}} \left| \frac{\partial V(\boldsymbol{q})}{\partial \boldsymbol{q}} \right| \mathrm{d}\boldsymbol{q}}_{c_{\text{D}}(\boldsymbol{q}_{\text{i}}, \boldsymbol{q}_{\text{f}})} \tag{4.9}$$

where c_{P} calculates the motion length, c_{I} measures the motion effort, computed as the product of the average value of $V(\boldsymbol{q})$ and the motion length, and c_{D} evaluates the variations of $V(\boldsymbol{q})$ along the motion. Note that, considering piece-wise linear paths in $\mathcal{C}$, the cost $c_{\mathcal{P}}$ of a given path $\mathcal{P}$ is computed by adding the costs of the rectilinear segments composing $\mathcal{P}$. Therefore, the path minimizing this motion-cost function connects the start and the goal configurations in the shortest way that avoids the areas with high $V(\boldsymbol{q})$ values (i.e. with repulsive potential fields) and, at the same time, keeps $V(\boldsymbol{q})$ as monotonic as possible along the path (i.e. avoiding unneeded motions from repulsive to attractive potential fields and vice versa).

Given the introduced motion-cost function, the RRT* planner (Karaman and Frazzoli, 2011) is able to construct low-cost paths following the valleys and saddle points of the potential field function $V(\boldsymbol{q})$, and, hence, plan robot motions that follow the demonstration paths. In addition, this algorithm is probabilistically optimal, i.e. the solution converges to the optimal path as the number of iterations tends to infinite, and the computational cost of the algorithm is just proportional to the cost of the RRT. The procedure of the RRT*, described in Algorithm 5, is similar to procedure of the standard RRT: a random configuration $\boldsymbol{q}_{\text{rand}}$ is sampled in $\mathcal{C}$ and a step of length ϵ is taken towards $\boldsymbol{q}_{\text{rand}}$ from $\boldsymbol{q}_{\text{near}}$ (the configuration in the configuration tree being closest to $\boldsymbol{q}_{\text{rand}}$), reaching a new configuration $\boldsymbol{q}_{\text{new}}$. Nevertheless, $\boldsymbol{q}_{\text{new}}$ is not connected

Algorithm 5: RRT*

```
Input  : Start configuration $\boldsymbol{q}_{\text{start}} \in \mathcal{C}_{\text{free}}$ and goal configuration $\boldsymbol{q}_{\text{goal}} \in \mathcal{C}_{\text{free}}$
Output: Path $\mathcal{P} \in \mathcal{C}_{\text{free}}$ connecting $\boldsymbol{q}_{\text{start}}$ and $\boldsymbol{q}_{\text{goal}}$

$\mathcal{T} \leftarrow$ InitTree($\boldsymbol{q}_{\text{start}}$)
while not StopCriteria($\mathcal{T}$) do
    $\boldsymbol{q}_{\text{rand}} \leftarrow$ RandConf( )
    $\boldsymbol{q}_{\text{new}} \leftarrow$ BestExtend($\mathcal{T}, \boldsymbol{q}_{\text{rand}}$)
    if $\boldsymbol{q}_{\text{new}} \neq \emptyset$ then
        Rewire($\mathcal{T}, \boldsymbol{q}_{\text{new}}$)
return Path($\mathcal{T}$)
```

Algorithm 6: BestExtend

```
Input  : Configuration tree $\mathcal{T}$ and configuration $\boldsymbol{q}$
Output: Configuration $\boldsymbol{q}_{\text{new}}$

$\boldsymbol{q}_{\text{near}} \leftarrow$ NearestConf($\mathcal{T}, \boldsymbol{q}$)
$\boldsymbol{q}_{\text{new}} \leftarrow \boldsymbol{q}_{\text{near}} + \min\big(\epsilon, \|\boldsymbol{q} - \boldsymbol{q}_{\text{near}}\|\big) \frac{\boldsymbol{q} - \boldsymbol{q}_{\text{near}}}{\|\boldsymbol{q} - \boldsymbol{q}_{\text{near}}\|}$
$\boldsymbol{q}_{\text{parent}} \leftarrow$ BestParent($\mathcal{T}, \boldsymbol{q}_{\text{new}}$)
if $\boldsymbol{q}_{\text{parent}} \neq \emptyset$ then
    AddSegment($\mathcal{T}, \boldsymbol{q}_{\text{parent}}, \boldsymbol{q}_{\text{new}}$)
    return $\boldsymbol{q}_{\text{new}}$
return $\emptyset$
```

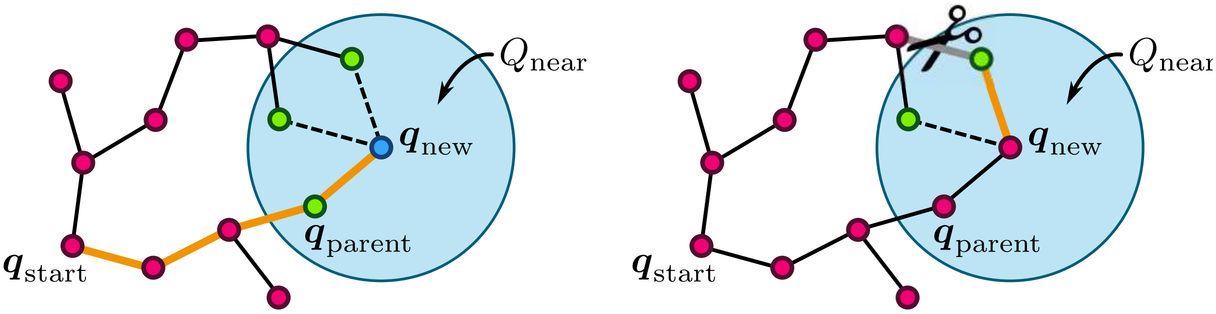


Figure 4.7: RRT* rewiring procedure: The configuration $\boldsymbol{q}_{\text{parent}}$, to which the new configuration $\boldsymbol{q}_{\text{new}}$ is connected, is in the set Q_{near} of neighboring configurations of $\boldsymbol{q}_{\text{new}}$ and is the one that offers the lowest cost path (left). The remaining configurations in Q_{near} are reconnected if the path to $\boldsymbol{q}_{\text{start}}$ via $\boldsymbol{q}_{\text{new}}$ is less expensive than the path through the current route (right).

to the configuration tree directly through $\boldsymbol{q}_{\text{near}}$ (Line 4), i.e. following the standard RRT-function EXTEND (outlined in Algorithm 3). On the contrary, the function BESTEXTEND is used (described in Algorithm 6) and a set Q_{near} of tree configurations neighbouring $\boldsymbol{q}_{\text{new}}$ is considered. In this way, the configuration $\boldsymbol{q}_{\text{parent}}$, the configuration to which $\boldsymbol{q}_{\text{new}}$ is connected, is the configuration in Q_{near} that involves a lower-cost path reaching $\boldsymbol{q}_{\text{new}}$ from the start configuration $\boldsymbol{q}_{\text{start}}$, such that the rectilinear segment between $\boldsymbol{q}_{\text{parent}}$ and $\boldsymbol{q}_{\text{new}}$ is collision-free (Line 3). Then, after the tree extension, back in Algorithm 5, the configuration $\boldsymbol{q}_{\text{new}}$ is considered to replace the predecessor configuration of each configuration in Q_{near} (Line 6). This tree rewiring means that in case of reaching from $\boldsymbol{q}_{\text{start}}$ a given configuration $\boldsymbol{q} \in Q_{\text{near}}$ a lower cost is obtained via $\boldsymbol{q}_{\text{new}}$ than trough the current path in the tree, $\boldsymbol{q}_{\text{new}}$ is then the predecessor configuration of the considered configuration $\boldsymbol{q}$ (see Fig. 4.7). Besides, notice that the RRT* algorithm does not finish when a solution is found, instead it continues finding better and better solutions until some stop criteria is hold (Line 2).

The RRT* algorithm has a big tendency to explore the whole configuration $\mathcal{C}$. In fact, with time enough, this planner finds the optimal paths connecting $\boldsymbol{q}_{\text{start}}$ to every single configuration in $\mathcal{C}$. This is highly inconsistent with the problem being solved, i.e. finding the optimal path only connecting $\boldsymbol{q}_{\text{start}}$ to $\boldsymbol{q}_{\text{goal}}$. In order to cope with the limitations of the standard RRT*, a modified version was proposed with the following changes (Akgun and Stilman, 2011):

- *A sampling bias*: Once a solution has been found, the sampling is biased, with a given probability P_{bias}, towards configurations around the found path. This guides the solution towards local optima, which is enough for the considered objective.

- *A node-rejection criteria*: Those samples that may not be useful in finding a better solution than the current one are discarded. This keeps the tree as reduced as possible, thus reducing the computational cost in the nearest-neighbour search in the computation of the configuration $\boldsymbol{q}_{\text{near}}$.

Besides, the planning algorithm HD-RRT* proposed here, from *Human-Demonstrated RRT**, is based on the RRT* and implements also the next additional modifications of the RRT* planner:

- *The optimization function*: The standard RRT* minimizes the path length measured in $\mathcal{C}$. Here the $c(\boldsymbol{q}_{\text{i}}, \boldsymbol{q}_{\text{f}})$ motion cost is minimized, see Eq. (4.9). Thereby, the solution is guided

Algorithm 7: NEWCONF

```
Input  : Configurations $\boldsymbol{q}_{\text{near}}$ and $\boldsymbol{q}_{\text{rand}}$
Output: Configuration $\boldsymbol{q}_{\text{new}}$
1: if RAND01() $\geq P_{grad}$ then
2:     return $\boldsymbol{q}_{\text{near}} + \min(\epsilon, \|\boldsymbol{q}_{\text{rand}} - \boldsymbol{q}_{\text{near}}\|)$ UNITVECTOR($\boldsymbol{q}_{\text{rand}} - \boldsymbol{q}_{\text{near}}$)
3: $\omega \leftarrow (\omega_{\text{P}} + \omega_{\text{I}} + \omega_{\text{D}})$ RAND01()
4: if $\omega < \omega_P$ then
5:     return $\boldsymbol{q}_{\text{near}} + \min(\epsilon, \|\boldsymbol{q}_{\text{goal}} - \boldsymbol{q}_{\text{near}}\|)$ UNITVECTOR($\boldsymbol{q}_{\text{goal}} - \boldsymbol{q}_{\text{near}}$)
6: else if $\omega \in [\omega_P, \omega_P + \omega_I]$ then
7:     return $\boldsymbol{q}_{\text{near}} - \epsilon$ UNITVECTOR($\nabla V(\boldsymbol{q}_{\text{near}})$)
8: else
9:     return $\boldsymbol{q}_{\text{near}} + \epsilon$ RANDORTHONORMALVECTOR($\nabla V(\boldsymbol{q}_{\text{near}})$)
```

towards short paths that follow as much as possible the demonstrated movements and that move away from obstacles and from self-collisions.

- *The extension procedure*: In the standard RRT* growth of the tree, the selected node is steered towards the sampled random configuration $\boldsymbol{q}_{\text{rand}}$. Here, this is modified to steer the node with a probability P_{grad} towards low-cost directions (with a stochastic gradient-descent method) using the function NEWCONF detailed below (i.e. Line 2 in Algorithm 6 is replaced by a call to the NEWCONF function outlined in Algorithm 7).

The proposed HD-RRT* planning algorithm uses the procedure NEWCONF to compute the new configuration being added to the tree, see Algorithm 7. $\boldsymbol{q}_{\text{new}}$ is computed performing an incremental step ϵ from $\boldsymbol{q}_{\text{near}}$ towards $\boldsymbol{q}_{\text{rand}}$ (Line 2), as it is done in the standard RRT algorithm. However, with a probability $P_{\text{grad}} < 1$, a stochastic gradient-descent method minimizing the motion cost is applied instead. Note that the gradient-descent method can be trapped in local minima of the motion cost. Nevertheless, since the gradient descent is not applied always in all the iterations, the RRT* exploration properties are preserved and the possible local minimum traps are avoided (assuming $P_{\text{grad}} < 1$). The extension direction is chosen randomly (Line 3) between the directions that minimize each component of the motion cost, see c_{P}, c_{I} and c_{D} in Eq. (4.9). Each of these cost components is chosen to be minimized with a probability proportional to its weight value ω_{P}, ω_{I} and ω_{D} respectively (e.g. the greater ω_{P} regarding ω_{I} and ω_{D}, the greater the probability that the tree grows in the direction that minimizes c_{P}). Then:

- Since c_{P} measures the path length, the direction pointing from $\boldsymbol{q}_{\text{near}}$ towards $\boldsymbol{q}_{\text{goal}}$ minimizes c_{P} (Line 5).
- Since c_{I} measures the average value of the potential field along the path, and the gradient $\triangledown V(\boldsymbol{q})$ of the potential field points in the direction of the local greatest growth of $V(\boldsymbol{q})$, then the direction that minimizes c_{I} points in the opposite direction of $\triangledown V(\boldsymbol{q})$, i.e. the direction in which $V(\boldsymbol{q})$ locally decreases (Line 7).
- Since c_{D} measures the variations of $V(\boldsymbol{q})$ along the motion, then any random direction orthogonal to $\triangledown V(\boldsymbol{q})$ minimizes c_{D} because $V(\boldsymbol{q})$ does not locally grow in any direction perpendicular to $\triangledown V(\boldsymbol{q})$ (Line 9).

After several tests with different tasks, the parameters of the HD-RRT* algorithm have been empirically set to $P_{\text{bias}} = 0.1$, and $P_{\text{grad}} = 0.1$, being ϵ dependant on the task. Regarding the motion-cost function, the motion connecting straightly $\boldsymbol{q}_{\text{start}}$ and $\boldsymbol{q}_{\text{goal}}$ has been used to set each weight of the motion-cost function: $\omega_{\Gamma} = c_{\Gamma}(\boldsymbol{q}_{\text{start}}, \boldsymbol{q}_{\text{goal}})^{-1}$ for $\Gamma \in \{\mathrm{P}, \mathrm{I}, \mathrm{D}\}$, see Eq. (4.9). In addition, the parameters of the potential fields have been empirically set to: $\lambda_{\text{goal}} = 0.1$, $\sigma_{\text{goal}} = 0.1$; $\lambda_i = 0.1$, $\sigma_i = 7\ \forall i$; $\lambda_{\text{arms}} = 0.3$, $\sigma_{\text{arms}} = 10$; and $\lambda_j = 0.3$, $\sigma_j = 10\ \forall j$ (see Eq. (4.5)-(4.8)). The sensibility of the system performance with respect to the planner parameters is not high, thus determining them is not a critical issue. Thereby, the same values of the parameters are used in the conceptual and the real examples presented in the next subsection, with the exception of the parameter ϵ that is the unique task-dependant parameter.

4.3.2 Approach validation

The proposed approach has been implemented within The Kautham Project (Rosell et al., 2014) and has been tested in two experiments, described below, running in a 2.13-GHz Intel 2, 4-GB RAM PC.

First, for illustrative purposes, a simple example has been set up. It consists of a 2D scenario where a 2-link planar manipulator must move from the start configuration $\boldsymbol{q}_{\text{start}}$ to the goal configuration $\boldsymbol{q}_{\text{goal}}$ avoiding collisions with circular obstacles (see Fig. 4.8). Three demonstration paths and $\boldsymbol{q}_{\text{goal}}$ were used to generate the attractive potential field, while the circular obstacles

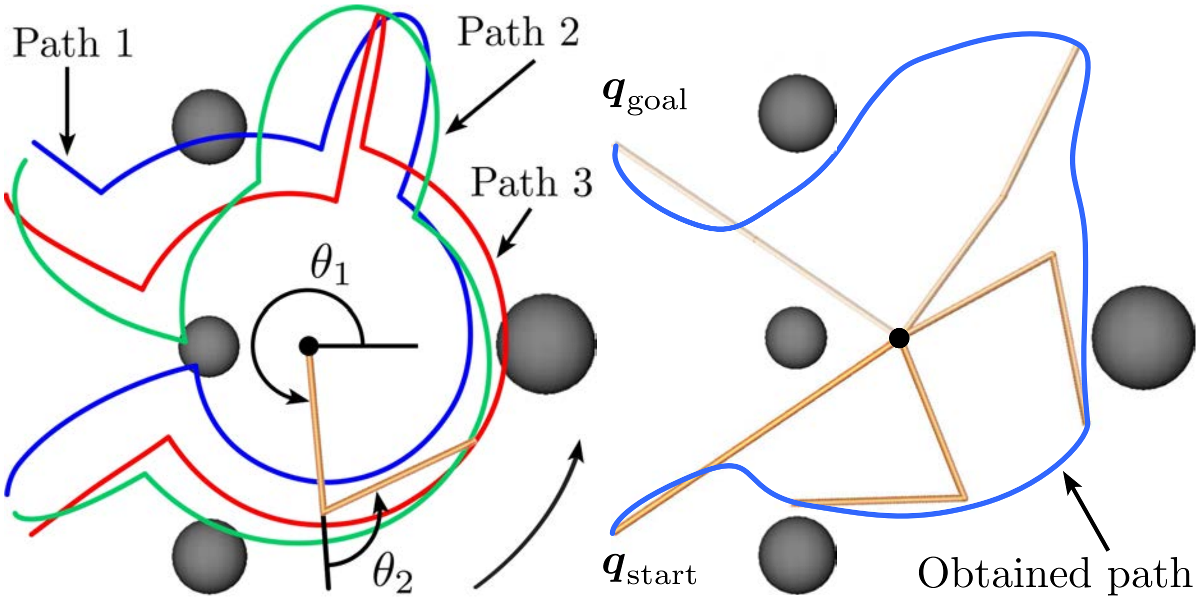


Figure 4.8: 2-link planar manipulator problem: demonstrated paths (left) and obtained solution path (right).

generate repulsive potential fields. The combination of these attractive and repulsive potential fields forms the potential-field function $V(\boldsymbol{q})$. Fig. 4.8-left shows the three demonstration paths in the problem space and Fig. 4.8-right shows the obtained solution. Fig. 4.9 shows resulting potential-field function $V(\boldsymbol{q})$ in the configuration space, including the three demonstration paths in Fig. 4.9-left and the resulting configuration tree and the obtained solution Fig. 4.9-right.

Note that $V(\boldsymbol{q})$ is shaped like a plateau in the regions of $\mathcal{C}$ where the manipulator is in collision with the obstacles (depicted in black in Fig. 4.9), while, on the other hand, the demonstration paths originate valleys (warm colored in Fig. 4.9). Therefore, the use of $V(\boldsymbol{q})$ in the c_{I} and c_{D} cost components in Eq. (4.9) enforce the solution path to follow the demonstrations as close as possible, while c_{P} tries to shorten the path.

The planning procedure assures that a solution path avoiding obstacles and self-collisions is found (if one exists) due to the asymptotic completeness of any RRT-based planning algorithm, even if the demonstration paths are not collision-free (as it actually happens in this example, see Fig. 4.8 and Fig. 4.9). It must be also remarked that the fact that the demonstration paths provide relevant information on a given task is more significant than the number of demonstration paths used, and that the method works well even with a single demonstration (in this case the

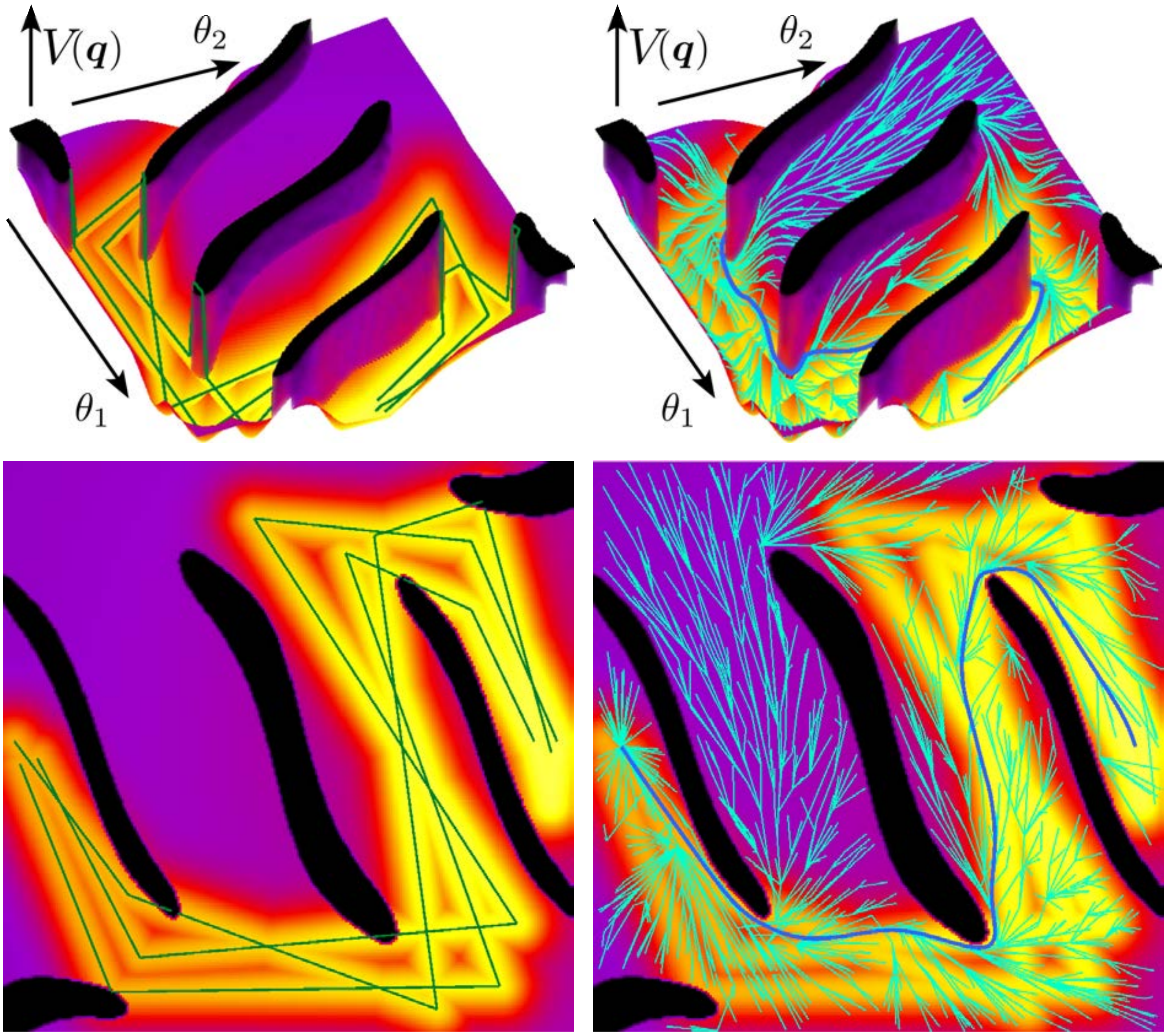


Figure 4.9: Tridimensional and top views of the potential field $V(\boldsymbol{q})$ over the configuration space $\mathcal{C}$ for the example in Fig. 4.8 (warmer colors represent lower values of $V(\boldsymbol{q})$), with the demonstrated paths (left), and the obtained solution path and configuration tree (right).

valleys are sharper, and they become wider when there are several different demonstration paths, with the valley width growing when the dispersion of the demonstration paths grows). On the other hand, the computation time of the motion-cost function grows when the number of demonstration paths increases, as it is expected according to Eq. (4.6), but it does not produce any other negative consequence.

Table 4.5: Average results of the conceptual example motion planning.

Time first solution	Solution length L	Total cost $c_{\mathcal{P}}$	Unit cost $\hat{c}_{\mathcal{P}}$
0.26 s	10.825 rad	1.195	0.111

Numerical results of the 2-link planar manipulation problem using three demonstration paths are summarized in Table 4.5, where the average values after 100 executions are shown (remind that the sampling-based planners rely on a random process and therefore generate a different solution each execution). A maximum time of 10 s was allowed for each execution, this was enough to get a 100% of success rate, i.e. the system finds always a collision-free path avoiding self-collisions and collisions with the obstacles. The collected data include:

- The final solution length L (measured in $\mathcal{C}$ as the summation of joint movements in radians of the path obtained after 10 s of planning).
- The final path cost $c_{\mathcal{P}}$ (defined as the sum of the motion costs of all the segments that form the path obtained after 10 s of planning).
- The unit path cost $\hat{c}_{\mathcal{P}}$ (computed as $c_{\mathcal{P}}$ divided by L).

After this simple example, the planning of the motions of the MADAR robot (introduced in Subsection 1.3.1) for the *Assembly* task described in Section 4.1 is used as a real example of the proposed planning procedure. From the captured data set, five demonstration paths are used to generate the attractive potential fields, see Eq. (4.6). Being the start and the goal dual-arm configurations given, the dual-motions motions are planned using the whole configuration space $\mathcal{C}$ and also using the reduced planning subspace B_k of the demonstrated task (see Subsection 3.1.3), which for this task has dimension 2 (see Subsection 4.2.1), with maximum allowed planning times of 100 s and 10 s, respectively, assuring a 100% of success rate.

Fig. 4.10 shows snapshots of a solution path for the assembly task obtained using the proposed approach, planning in the reduced subspace B_k. Videos of the experiment for the assembly task are available in sir.upc.edu/projects/human-demonstrated/index.html. Table 4.6 shows the average results obtained after 100 executions for each case using five demonstration paths. The table includes, as *valid motion rate*, the proportion of iterations in which no collisions occur and the tree actually grows.

As expected, the best paths (the ones with the lowest cost) are obtained when the whole $\mathcal{C}$ space was used. Nevertheless, planning in B_k allows a much shorter planning time (due to

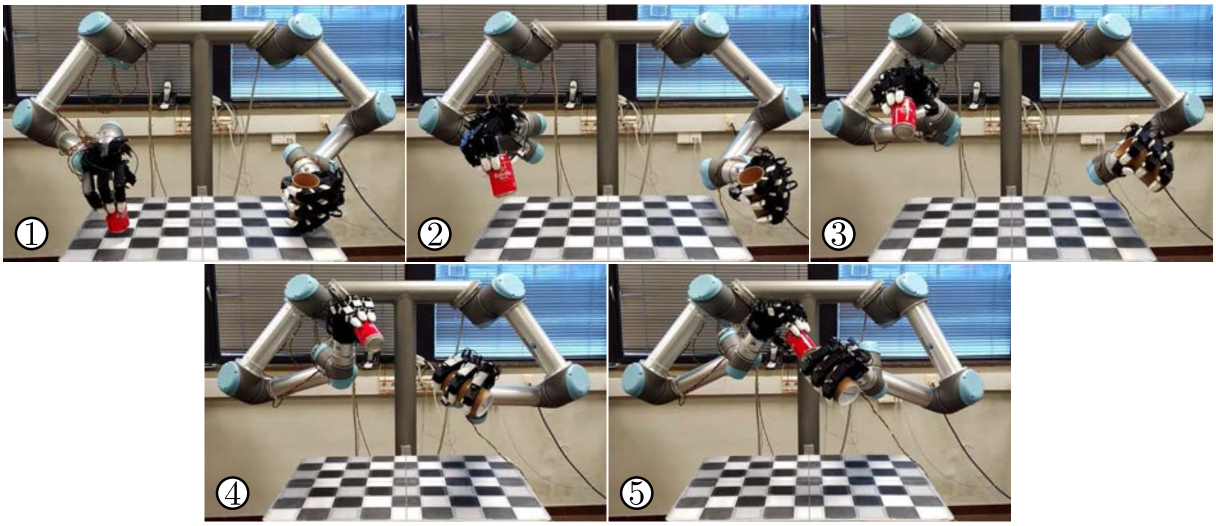


Figure 4.10: Snapshots of a solution path for the assembly example planned in the reduced subspace B_k.

Table 4.6: Average results of the assembly example motion planning.

Planning space / Dimension	$\mathcal{C}$ / 12	B_k / 2
Planning time [s]	100	10
Time first solution [s]	1.35	0.33
Solution length [rad]	5.513	5.141
Total cost $c_{\mathcal{P}}$	1.095	1.186
Unit cost $\hat{c}_{\mathcal{P}}$	0.199	0.231
Valid motion rate [%]	73.49	94.56

the reduced dimension of the subspace) without incrementing excessively the cost of the path. In addition, the use of B_k increases the probability of obtaining collision-free configurations (see the *valid motion rate* in Table 4.6) because fewer self-collisions occur, and therefore the efficiency of the planning procedure increases.

First-order synergy bases, see Subsection 3.1.2, have been computed using the *Free-movement* task described in Section 4.1 (a detailed analysis of the obtained first-order synergies for this task can be found in the next section). These synergy bases were used for the computation of the human-likeness index $\mathcal{H}$ (see Subsection 3.2.2), which was applied to the evaluation of the human-likeness of the solutions found for the Assembly example using:

a) The HD-RRT* planner with several demonstrations, planning in the whole configuration space $\mathcal{C}$.

b) The HD-RRT* planner with several demonstrations, planning in the lower-dimensional

Table 4.7: Average human-likeness value of the Assembly example using the presented approach with several demonstrations (a, b), the RRT (c, d) and an artificial path (e).

Case	Planning dimension	Planning time [s]	Solution length [rad]	Human-Likeness $\mathcal{H}$
a)	12	100	5.513	0.594
b)	2	10	5.141	0.573
c)	12	32.39	6.647	0.433
d)	2	8.38	5.965	0.428
e)	-	-	17.452	0.045

subspace B_k.

c) The standard RRT planner Kuffner and LaValle (2000), planning in the whole configuration space $\mathcal{C}$.

d) The standard RRT planner, planning in the lower-dimensional subspace B_k.

e) Two rectilinear segments in $\mathcal{C}$ connecting the start and the goal configurations through an empirically selected configuration $\boldsymbol{q}_\mathrm{m} \notin B$, i.e. $\boldsymbol{q}_\mathrm{m}$ is not in the subspace of the sampled configurations of the *Free-movement*. Note that in this case no motion planning is performed but even so the path is checked to be free of collisions, either involving the obstacles or both robotic arms.

Videos of paths obtained for the Assembly task with the considered approaches are available in sir.upc.edu/projects/human-demonstrated/index.html.

Table 4.7 shows the average results obtained, for each case, after 100 executions. It can be noted that the proposed planner obtains paths with a significant better $\mathcal{H}$ quality (even though the presented approach needs a longer planning time), i.e. the proposed procedure finds solution paths that are better aligned with the natural movements of the human operator and that therefore are more human-like. Note that the approach presented in this work obtains a better $\mathcal{H}$ even though the human movements used in the motion planning are different to the ones used to compute $\mathcal{H}$. The path with the greatest quality is obtained when the motions are planned in the whole $\mathcal{C}$ space. However, the use of the subspace B_k is the best option since it reduces significantly the planning time without penalizing considerably $\mathcal{H}$. The poorest quality is obtained with the manually-set path (e), denoting that this path is not much human-like. The planner used in cases (c) and (d) does not consider human-likeness nor path length as a quality index. Hence, bad results are obtained for both measures.

4.4 Motion planning using first-order synergies

This section proposes the use of first-order synergies (see Subsection 3.1.2), obtained from captured human movements, to solve efficiently and in a human-like fashion bimanual tasks with anthropomorphic dual-arm robots. Besides, it proposes an automatic partition method to optimally divide the configuration space $\mathcal{C}$ into subregions where the first-order synergies are significantly different, based on the likeness index introduced in Subsection 3.2.2. Using the first-order synergies in each of these subregions, a vector field of desired velocities is generated over the $\mathcal{C}$ space. A bidirectional sampling-based planner, biasing the tree growth towards the directions of the synergy-based vector field, is then proposed.

4.4.1 Motion analysis

As seen in Subsection 3.1.3, the zero-order synergy basis ${}^0\mathcal{S}$ computed from demonstrated human movements of a given task defines a synergy box B. The box B indicates the relevant region of the configuration space $\mathcal{C}$ for this given task, i.e. the region where the demonstrated task movements lie. This is useful because the robot motions planned in this region maintain the human appearance of the mapped movements used to compute these synergies. However, the zero-order synergies only provide information about in which region of $\mathcal{C}$ should the robot move to obtain human-like motions but they do not provide information about with which joint velocity should the robot move. It is for this reason that the first-order synergies are needed. Nevertheless, the first-order synergies are not similar all over the configuration space, i.e. different correlations are observed in the joint velocity space for movements performed in different regions of $\mathcal{C}$. A solution to this problem is to split the synergy box of ${}^0\mathcal{S}$, i.e. B, into different subregions called *synergy cells* such that the first-order synergies associated to each synergy cell are significantly different to the ones associated to the neighboring synergy cells (see Fig. 4.11-left).

The partitions of B are organised as a kd-tree structure $\mathcal{T}_{KD}$, where each non-leaf node represents a partition by an hyperplane normal to a given axis of ${}^0\mathcal{S}$ and each leaf node represents a synergy cell, with an associated first-order synergy basis ${}^1\mathcal{S}$. The kd-tree is built starting with the box B which is being divided into two subcells recursively, based on the mapped con-

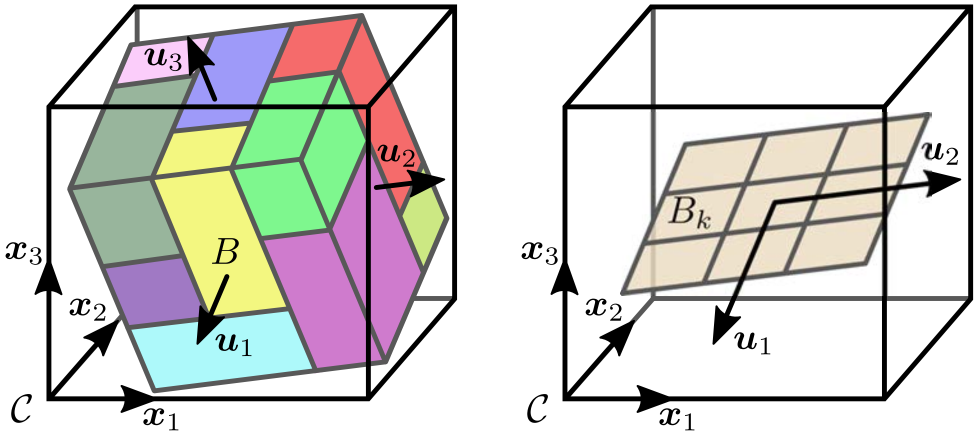


Figure 4.11: The box containing the zero-order synergies, called B, defines the relevant region of $\mathcal{C}$. In turn, B is divided into synergy cells so that the first-order synergies associated to each cell are different (left). A grid of tree cells in B_k, i.e. the subspace spanned by the first k zero-order synergies ($\boldsymbol{u}_1$ and $\boldsymbol{u}_2$ in the figure), is used to estimate the coverage of the whole $\mathcal{C}$ (right).

figurations from some given human motions. In this partition process, the synergy cell being split is called parent cell, and its associated first-order synergy basis is ${}^1\mathcal{S}_P$, and the synergy cells resulting from the partition are called left and right child cells, with ${}^1\mathcal{S}_L$ and ${}^1\mathcal{S}_R$ associated first-order synergies. The differences between the parent basis ${}^1\mathcal{S}_P$ and the child bases ${}^1\mathcal{S}_L$ and ${}^1\mathcal{S}_R$, measured using the likeness index $\mathcal{L}$ proposed in Subsection 3.2.1, is the criterion used to divide a given synergy cell. Note that the likeness index $\mathcal{L}$ has been used so far in this doctoral dissertation to calculate the similarity between zero-order synergies, i.e. using position configurations. In this section the index $\mathcal{L}$ is used with first-order synergies, i.e. using joint velocities, but the definition of the measure is still valid ($\mathcal{L}$ measures the overlapping between two sample distributions, without making assumptions about the type of samples used).

Thereby, given the set Q of mapped configurations $\boldsymbol{q}$ lying in a given parent cell, the first-order basis ${}^1\mathcal{S}_P$ is obtained by applying a PCA to the first derivative of the configurations in Q (see Subsection 3.1.2). After splitting the parent cell with a given hyperplane, the bases ${}^1\mathcal{S}_L$ and ${}^1\mathcal{S}_R$ are obtained applying the same procedure respectively to the subsets Q_L and Q_R, resulting from the division of the set Q by the same given hyperplane. Hence, for a given partition hyperplane π aligned with a given axis of ${}^0\mathcal{S}$, the value of the distance x of the hyperplane to center of B is the one that minimizes the following objective function

$$f = \max\left(\mathcal{L}({}^1\mathcal{S}_P, {}^1\mathcal{S}_L), \mathcal{L}({}^1\mathcal{S}_P, {}^1\mathcal{S}_R)\right) \tag{4.10}$$

Algorithm 8: SynergyTree

```
    Input  : Set of configurations $Q$
    Output: Synergy tree $\mathcal{T}_{KD}$
 1: $\boldsymbol{\pi} \leftarrow$ FindBestPartitionForEachAxis($Q$)
 2: $\boldsymbol{\pi}^* \leftarrow$ Sort($\boldsymbol{\pi}$)
 3: for $j \leftarrow 1$ to $n$ do
 4:     if partitioning by $\pi_j^*$ is valid then
 5:         $\{Q_L, Q_R\} \leftarrow$ Divide($Q, \pi_j^*$)
 6:         $\mathcal{T}_{KD}$.Partition $\leftarrow \pi_j^*$
 7:         $\mathcal{T}_{KD}$.Subtrees $\leftarrow$ (SynergyTree($Q_L$), SynergyTree($Q_R$))
 8:         return $\mathcal{T}_{KD}$
 9: $\mathcal{T}_{KD}$.Synergies $\leftarrow$ Synergies($Q$)
10: return $\mathcal{T}_{KD}$
```

Note that depending on the value of x, the parent cell is split differently and hence different ${}^1\mathcal{S}_L$ and ${}^1\mathcal{S}_R$ are obtained (obviously ${}^1\mathcal{S}_L$ and ${}^1\mathcal{S}_R$ also depend on the axis to which the hyperplane π is normal). This objective function pursues that both child first-order bases be different to the parent synergy basis. It is important to remark that f lies in the range $[0, 1]$ and that $f = 1$ in the limits of the x-domain (when one of the children cells is equal to the parent cell and the other is void). Therefore, there exists always a minimum (unless the objective function f is completely flat for the selected axis to which the partition hyperplane π is normal and in this case another axis is considered). Besides, let a partition dividing a cell be valid if each descendant cell contains at least 100 samples of Q and has an aspect ratio less than 1:5, i.e. the longer side of the resulting sector measures in the $\mathcal{C}$ space at most five times the shorter side. These values have been empirically chosen. Nevertheless, the sensibility of the procedure with respect to these parameters is not high, thus their values are not a critical issue.

The partition procedure, called SynergyTree, is detailed in Algorithm 8. It receives a set Q of configurations and finds the best partition hyperplane normal to each axis of ${}^0\mathcal{S}$ (Line 1), using Eq. (4.10). The computed hyperplanes, stored in $\boldsymbol{\pi}$, are sorted in increasing order of the associated value of the objective function f (Line 2). Then, the sorted list of hyperplanes $\boldsymbol{\pi}^*$ is traversed (Line 3), until a valid partition hyperplane is found. If this occurs (Line 4), the synergy cell is split into two subcells, i.e. we are in a non-leaf node of the kd-tree. The given set Q is also split into two sets Q_L and Q_R, using the selected partition hyperplane π_j^*, and the SYNERGYTREE procedure is self-invoked, expanding the tree in the left and right child cells (Lines 5-7). If no valid partition is found, the kd-tree is not further expanded, i.e. we are in a leaf node, and the first-order synergies associated to the synergy cell are computed (Line 9), computing numerically the derivative of the configurations in Q and applying then a PCA.

After applying the presented partition procedure for a given demonstrated task, the zero- and first-order synergies as well as the synergy cells can be used to generate a vector-field of desired velocities as follows. For a given configuration $\boldsymbol{q}$ in $\mathcal{C}$, its associated desired velocity in the vector-field is randomly chosen from the subspace spanned by the first p first-order synergies associated to the synergy cell where $\boldsymbol{q}$ lies (or the closest one if $\boldsymbol{q}$ lies outside B). Note that the first-order synergies are sorted in decreasing order of associated sample variance and that p is the minimum number of synergies that accumulate more than the 95% of the total variance. Besides, as explained in Subsection 3.1.3, the zero-order synergies define a subspace $B_k \subseteq B$ that is spanned by the first k zero-order synergies (taking them in descendant order of the associated sample variance), such that the accumulated variance is above the 95% of the total sample variance. This subspace can be used to detect the unexplored areas of $\mathcal{C}$ during the motion planning if a cell-based discretization is established in the box B_k and the configuration trees are projected into these cells (see Fig. 4.11-right). Hence, those cells in which no configuration tree is projected denote unexplored areas of $\mathcal{C}$. These cells in B_k, indicating whether a region of $\mathcal{C}$ has been explored, are called *tree cells*, so as not to confuse them with the synergy cells that indicate regions of $\mathcal{C}$ with similar first-order synergies. Both synergy cells and tree cells are used in the next subsection to obtain human-like dual-arm movements.

4.4.2 Motion planning

The motion-planning problems where there is a preferred direction of movement for each configuration are well framed as motion planning on vector fields. To solve this kind of problems, an RRT-based planner, called VF-RRT (Ko et al., 2014), was proposed. This planning algorithm adjusts the randomly sampled nodes towards the vector-field direction as follows. Let $\hat{\boldsymbol{v}}_{\text{rand}}$ be the advance direction resulting from the RRT sampling (i.e. moving towards a random configuration, sampled in $\mathcal{C}$, from its nearest configuration in the tree, see Algorithm 3) and

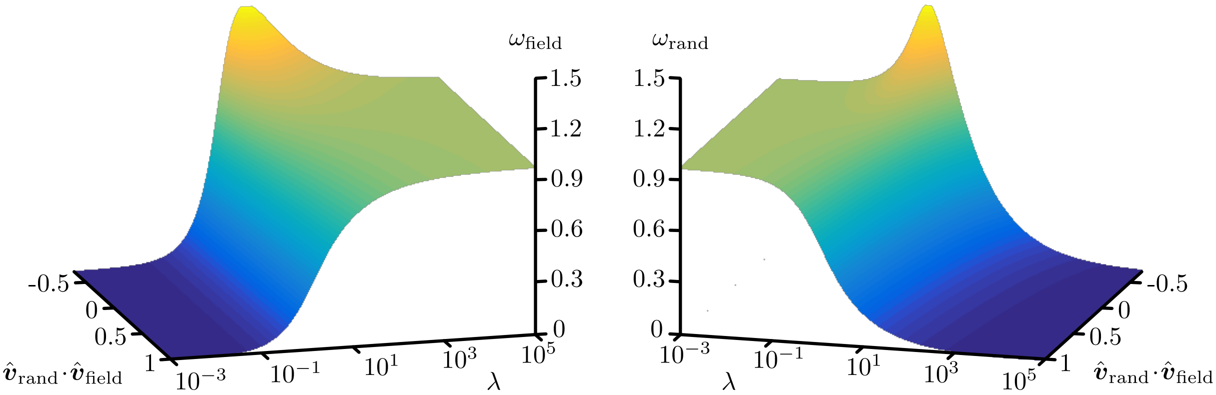


Figure 4.12: Values of the weights ω_{field} (left) and ω_{rand} (right) computed for different values of the parameter λ and for different values of the product $\hat{\boldsymbol{v}}_{\text{rand}} \cdot \hat{\boldsymbol{v}}_{\text{field}}$.

$\hat{\boldsymbol{v}}_{\text{field}}$ be the direction of vector-field that must be followed. Then, both directions are combined to obtain $\hat{\boldsymbol{v}}_{\text{new}}$, the actual advance direction

$$\hat{\boldsymbol{v}}_{\text{new}} = \omega_{\text{rand}}\, \hat{\boldsymbol{v}}_{\text{rand}} + \omega_{\text{field}}\, \hat{\boldsymbol{v}}_{\text{field}} \tag{4.11}$$

with $\omega_{\text{rand}}, \omega_{\text{field}} \geq 0$ and ensuring that $\|\hat{\boldsymbol{v}}_{\text{new}}\| = 1$. The weights ω_{rand} and ω_{field} are controlled by a parameter $\lambda > 0$ such that $\hat{\boldsymbol{v}}_{\text{new}} \to \hat{\boldsymbol{v}}_{\text{field}}$ if $\lambda \to \infty$ and $\hat{\boldsymbol{v}}_{\text{new}} \to \hat{\boldsymbol{v}}_{\text{rand}}$ if $\lambda \to 0$. The relationship between λ and the weights ω_{rand} and ω_{field}, stated by Ko et al. (2014), is graphically shown in Fig. 4.12. Nevertheless, λ is not a fixed parameter, it is adaptively adjusted according to the progress of the motion planning, such that λ decreases if difficulties are found in growing along the vector-field directions and vice versa.

In the work of Ko et al. (2014), λ is initialized to a predefined λ_{init} and it is updated every N iterations as follows. First, let $\boldsymbol{q}$ be a configuration candidate to be added to the configuration tree and a rectilinear segment in $\mathcal{C}$ be called *motion*. Then, $\boldsymbol{q}$ is considered as being *efficient* if:

a) m, the motion reaching $\boldsymbol{q}$, is collision-free, and

b) δ, the distance between $\boldsymbol{q}$ and its closest configuration in the tree, is greater than a predefined $\delta_{\text{ineff}} \in [0, \epsilon]$, where ϵ is the standard RRT step-size parameter.

Let $E_{\text{ineff}} \in [0, 1]$ be the rate of inefficient nodes found in the last N iterations, and E^*_{ineff} be a reference value for E_{ineff}. Then, λ is updated as $\lambda' = \lambda\,(1 + E^*_{\text{ineff}} - E_{\text{ineff}})$. Thereby, λ grows if $E_{\text{ineff}} < E^*_{\text{ineff}}$ and vice versa. It is important to highlight that λ_{init}, k, δ_{ineff} and E_{ineff} are user-defined and that some suitable values may be difficult to find. In addition, the parameter configuration is highly problem-dependant.

For these reasons, a new method to update λ is proposed here. First, λ is initialized to λ_{max}, i.e. a high enough value so that the tree follows the vector field at the first iteration. As opposed to the original procedure, λ is updated at each iteration. Hence a smooth continuous growth of the tree is ensured. The new value λ' is computed as

$$\lambda' = \begin{cases} \lambda\, e^{-1} & \text{if } m \text{ is in collision} \\ \lambda\, e^{1-2\left(1-\frac{\delta}{\epsilon}\right)^{0.3}} & \text{otherwise} \end{cases} \tag{4.12}$$

Finally, λ is clamped to the range $[\lambda_{\text{min}}, \lambda_{\text{max}}]$ to prevent it from underflowing or growing too much unnecessarily. λ_{min} and λ_{max} are set to 10^{-3} and 10^5, respectively, since no significant changes are observed in $\hat{\boldsymbol{v}}_{\text{new}}$ when λ varies from 0 to λ_{min} or from λ_{max} to ∞. Note that in this way, the user does not need to define any parameter and λ still decreases if it is difficult to grow the tree following the vector field (i.e. m implies collision or $\delta \to 0$).

To test the performance of the proposed modifications of the VF-RRT planner, a set of 2D benchmarking problems have been set up. They consist of a mobile robot navigating in an obstacle-free square of side length 1 m, where four different vector fields have been established (see Fig. 4.13). The upstream criterion $\mathcal{U}$, proposed by Ko et al. (2014), is used as a quality metric in the comparison of the solution paths obtained with the original and the proposed VF-RRT. $\mathcal{U}$ measures the extent to which a path $\mathcal{P}$ goes against a vector field $\boldsymbol{f}(\boldsymbol{q})$ and is computed as

$$\mathcal{U} = \int_{\mathcal{P}} \|\boldsymbol{f}(\boldsymbol{q})\| - \frac{\boldsymbol{f}(\boldsymbol{q}) \cdot \dot{\boldsymbol{q}}}{\|\dot{\boldsymbol{q}}\|} \, \mathrm{d}\boldsymbol{q} \tag{4.13}$$

Thereby a path with lower $\mathcal{U}$ is less deviated with respect to the directions of the vector field and, therefore, is considered to be better. For instance, in case the vector field is computed with human movements, the lower $\mathcal{U}$ a path obtains, the better the human movements are mimicked. A maximum planning time of 5 s is considered in the experimentation. If a path is not obtained within this time, the execution is marked as a failure. Table 4.8 shows the resulting average

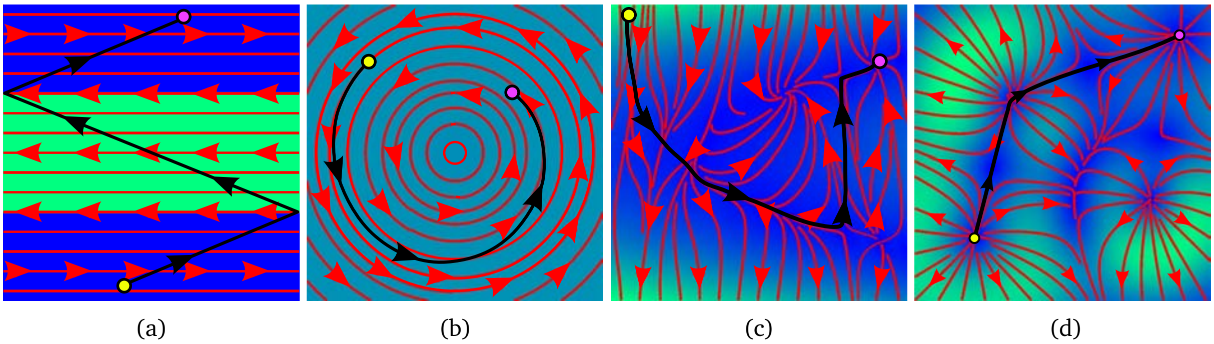


Figure 4.13: Benchmarking of the modified VF-RRT. A mobile robot travels from the start to the goal position through the unit square, where no obstacles are present and different vector fields $\boldsymbol{f}$ are defined in each case. The directions of $\boldsymbol{f}$ are depicted by red stream lines and the magnitude of $\boldsymbol{f}$ is denoted by the background color (changing from blue to green as $\|\boldsymbol{f}\|$ grows). The black lines show the paths with minimum upstream criterion.

Table 4.8: Average results of the motion planning of the benchmarking problems.

$\boldsymbol{f}$	VF-RRT version	Success rate [%]	Planning time [s]	# Planning iterations	Upstream criterion	Solution length [rad]
a)	Modified	100	0.2665	1266	0.421	1.753
	Original	90.1	0.6222	3135	0.529	1.679
b)	Modified	100	0.1863	672	0.146	1.870
	Original	96.5	0.4629	2540	0.469	1.614
c)	Modified	100	0.1641	541	0.675	1.495
	Original	93.1	0.4000	1956	1.140	1.403
d)	Modified	100	0.2169	894	1.053	1.556
	Original	92.0	0.5363	2764	1.087	1.566

values of the success rate, the planning time, the number of iterations, the path upstream criterion $\mathcal{U}$ and the solution length. Note that for each vector field, 100 executions have been run for the modified VF-RRT and that 10,000 different parameter configurations were used for the original planner. In order to consider all the possible instances of the original VF-RRT, the value of its parameters have been uniformly chosen at random from the corresponding intervals and a high enough maximum value for k has been considered. The approach proposed in this work has been implemented within The Kautham Project (Rosell et al., 2014) and run in a 3.40-GHz Intel i7-3770, 4-GB RAM PC. From the simulation results shown in Table 4.8 it can be appreciated that the modified planner VF-RRT outperforms the original planner in all aspects: it obtains better solution paths (i.e. with a lower $\mathcal{U}$) and in less time. In fact, the original VF-RRT is not able to find a solution within the time restriction for some executions.

Paths with a low upstream criterion (i.e. highly aligned with the vector field) are obtained with the method used by the planner VF-RRT, see Eq. (4.11). Nevertheless, the VF-RRT planner uses a nearest-neighbor structure to select the node that must be extended, as any RRT-based planner does. This produces a considerable computational load and, furthermore, it makes the planning algorithm vulnerable to the problems that may appear if the Euclidean distance is not a good metric for the configuration space (Palmieri and Arras, 2015). Besides, these problems are exacerbated in scenarios with a high number of DOFs, like, for instance, the case of planning dual-arm motions. On the other side, the KPIECE planner, Kinodynamic planning by interior-exterior cell (Suçan and Kavraki, 2012), settles all these problems by using another method, which does not rely on nearest-neighbors, to bias the growth of the configuration tree towards the unexplored areas of $\mathcal{C}$. This planner guides the exploration of $\mathcal{C}$ using a projection of the tree configurations into a discretized space: the configuration tree is mostly extended from configurations lying in the boundary of this discretization. Hence, the algorithm here proposed to plan dual-arm motions following the synergy-based vector-field introduced in the previous subsection is a KPIECE-variant planner. More specifically, the proposed planning algorithm is called FOS-BKPIECE, from *First-Order Synergies Bidirectional KPIECE*, and explained below, has a basic structure similar to the bidirectional implementation of the KPIECE planner provided by the *Open Motion Planning Library* (Suçan et al., 2012). Additionally, the proposed FOS-BKPIECE planner uses the cell discretization of the lower-dimensional space B_k, introduced in the previous subsection, to estimate the coverage of $\mathcal{C}$, and the here introduced modifications of the VF-RRT planner to guide the tree towards the synergy directions. Thus, solution paths that mimic the human movements are obtained.

The proposed planner is described in Algorithm 9 and has the following main features:

- Two trees, one rooted at the start configuration $\boldsymbol{q}_{\text{start}}$ and the other at the goal configuration $\boldsymbol{q}_{\text{goal}}$ (Lines 1-2), are steered towards each other while exploring the configuration space $\mathcal{C}$ (the trees are stored in the structures $\mathcal{G}_{\text{A}}$ and $\mathcal{G}_{\text{B}}$ explained below). The function ADDSEGMENT($\mathcal{G}$, $\boldsymbol{q}_{\text{init}}$, $\boldsymbol{q}_{\text{new}}$, d) inserts the motion from $\boldsymbol{q}_{\text{init}}$ to $\boldsymbol{q}_{\text{new}}$ in the tree, so that $\boldsymbol{q}_{\text{init}}$ becomes the parent node of $\boldsymbol{q}_{\text{new}}$. d is a measure of the closeness of $\boldsymbol{q}_{\text{new}}$ to the other tree and it is used in the extension of the tree to figure out how easy it may be to connect the trees through the computed $\boldsymbol{q}_{\text{new}}$ (see Fig. 4.14). Hence d can be an estimation of the

Algorithm 9: FOS-BKPIECE

```
    Input  : Start configuration $\boldsymbol{q}_{\text{start}} \in \mathcal{C}$ and goal configuration $\boldsymbol{q}_{\text{goal}} \in \mathcal{C}$
    Output: Collision-free path $\mathcal{P}$ from $\boldsymbol{q}_{\text{start}}$ to $\boldsymbol{q}_{\text{goal}}$
 1: AddSegment($\mathcal{G}_{\text{A}}, \boldsymbol{q}_{\text{start}}, \boldsymbol{q}_{\text{start}}, \|\boldsymbol{q}_{\text{goal}} - \boldsymbol{q}_{\text{start}}\|$)
 2: AddSegment($\mathcal{G}_{\text{B}}, \boldsymbol{q}_{\text{goal}}, \boldsymbol{q}_{\text{goal}}, \|\boldsymbol{q}_{\text{start}} - \boldsymbol{q}_{\text{goal}}\|$)
 3: while not StopCriteria($\mathcal{G}_{\text{A}}, \mathcal{G}_{\text{B}}$) do
 4:     $\boldsymbol{q}_{\text{init}} \leftarrow$ RandConf($\mathcal{G}_{\text{A}}$.SelectCell( ))
 5:     $\boldsymbol{q}_{\text{bias}} \leftarrow$ RandConf($\mathcal{G}_{\text{B}}$.TopExteriorCell( ))
 6:     $\boldsymbol{q}_{\text{new}} \leftarrow$ Extend($\mathcal{G}_{\text{A}}, \boldsymbol{q}_{\text{init}}, \boldsymbol{q}_{\text{bias}}$)
 7:     if $\boldsymbol{q}_{\text{new}} \neq \emptyset$ then
 8:         $\boldsymbol{q}_{\text{bridge}} \leftarrow$ RandConf($\mathcal{G}_{\text{B}}$.CellContaining($\boldsymbol{q}_{\text{new}}$))
 9:         if $\boldsymbol{q}_{\text{bridge}} \neq \emptyset$ and CollisionFree($\boldsymbol{q}_{\text{bridge}}, \boldsymbol{q}_{\text{new}}$) then
10:             AddSegment($\mathcal{G}_{\text{B}}, \boldsymbol{q}_{\text{bridge}}, \boldsymbol{q}_{\text{new}}, 0$)
11:             return Path($\mathcal{G}_{\text{A}}, \mathcal{G}_{\text{B}}$)
12:     Swap($\mathcal{G}_{\text{A}}, \mathcal{G}_{\text{B}}$)
13: return $\emptyset$
```

actual minimum distance between $\boldsymbol{q}_{\text{new}}$ and any configuration in the other tree.

- The trees are projected into B_k to guess the explored areas of $\mathcal{C}$, or at least their projections into B_k (see Fig. 4.11). B_k has been divided into *tree cells* of a predefined cell size. The cells where each tree lie, referred as *grids* $\mathcal{G}_{\text{A}}$ and $\mathcal{G}_{\text{B}}$, are classified as *interior cells* if all their neighbor cells contain tree configurations; or as *exterior cells* otherwise (see Fig. 4.14). The tree cells must not be confused with the synergy cells in which B is divided, and that contain first-order synergies (see Subsection 4.4.1). Moreover, the cells in $\mathcal{G}_{\text{A}}$ and $\mathcal{G}_{\text{B}}$ are sorted by an heuristic-based score. Preference is given to the tree cells that: a) are exterior; b) have fewer neighbors; c) have been more recently populated; d) contain fewer tree configurations; and e) their nodes are close to the other tree and have been fewer times used to grow the tree (Suçan and Kavraki, 2012).
- At each iteration, the tree in $\mathcal{G}_{\text{A}}$ grows from a configuration $\boldsymbol{q}_{\text{init}}$, randomly selected from the tree configurations in the top-scored cell of $\mathcal{G}_{\text{A}}$ (Line 4). The configuration $\boldsymbol{q}_{\text{init}}$ is extended towards a new configuration $\boldsymbol{q}_{\text{new}}$ (Line 6), by an increment step ϵ, steered by the first-order synergies and $\boldsymbol{q}_{\text{bias}}$ (Line 5), a configuration randomly selected from the top-scored tree cell on the boundary of $\mathcal{G}_{\text{B}}$, the other grid (see Fig. 4.14). Note that both the functions SelectCell( ) and TopExteriorCell( ) return the cell with the highest

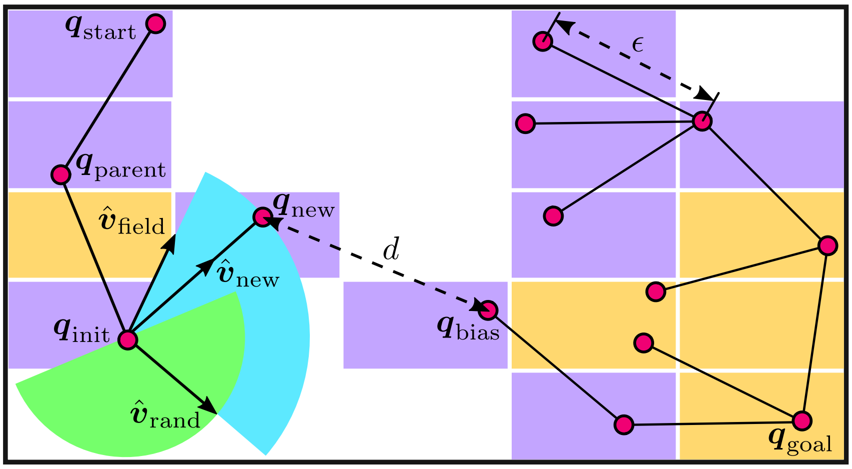


Figure 4.14: Hypothetical representation of the planning procedure: Two configuration trees, rooted at the configurations $\boldsymbol{q}_{\text{start}}$ and $\boldsymbol{q}_{\text{goal}}$, explore $\mathcal{C}$ pursuing their connections. To estimate the coverage of $\mathcal{C}$, the trees are projected to the subspace $B_k \subseteq \mathcal{C}$, where a cell-based discretization is established. The cells of B_k containing tree configurations are classified, based on their number of neighbors, as interior or exterior (filled in orange and purple, respectively). At each iteration, a tree is expanded from a configuration $\boldsymbol{q}_{\text{init}}$ to a new configuration $\boldsymbol{q}_{\text{new}}$. The advance direction $\hat{\boldsymbol{v}}_{\text{new}}$ lies in the blue region and is a combination of a random direction $\hat{\boldsymbol{v}}_{\text{rand}}$, lying in the green region, and the direction $\hat{\boldsymbol{v}}_{\text{field}}$ of the first-order synergies. In this example, $\boldsymbol{q}_{\text{new}}$ does not lie in any cell of the tree rooted at $\boldsymbol{q}_{\text{goal}}$ and therefore no configuration $\boldsymbol{q}_{\text{bridge}}$ exists and the connection of the trees cannot yet be attempted.

score (following the list of preferences introduced above), but while the former searches for the best cell all over the grid the latter narrows the search to the exterior cells. The function EXTEND is detailed in Algorithm 10 and explained below.

- If the extension of the tree in $\mathcal{G}_{\text{A}}$ succeeds, i.e. EXTEND( ) does not return $\emptyset$, the connection of the trees is attempted through a motion between $\boldsymbol{q}_{\text{new}}$ and $\boldsymbol{q}_{\text{bridge}}$, a configuration randomly selected from the configurations in the tree cell of $\mathcal{G}_{\text{B}}$ to which $\boldsymbol{q}_{\text{new}}$ would be projected (Line 9), if the tree cell exists. The function CELLCONTAINING($\boldsymbol{q}_{\text{new}}$) computes the projection of $\boldsymbol{q}_{\text{new}}$ into B_k and, without adding it to the grid, searches for the cell where the projection lies. If the motion connecting $\boldsymbol{q}_{\text{new}}$ and $\boldsymbol{q}_{\text{bridge}}$ is collision-free, it is added to $\mathcal{G}_{\text{B}}$ and the solution path from $\boldsymbol{q}_{\text{start}}$ to $\boldsymbol{q}_{\text{goal}}$ is returned (Lines 9-11). Otherwise, the tree roles are exchanged (Line 12).

The proposed planner uses the function EXTEND, outlined in Algorithm 10, to grow a tree from a given $\boldsymbol{q}_{\text{init}}$ to some $\boldsymbol{q}_{\text{new}}$ following a given vector field. It proceeds as follows:

- If $\boldsymbol{q}_{\text{init}}$ is in the neighborhood of $\boldsymbol{q}_{\text{bias}}$, i.e. the trees are closer than a distance ϵ, no synergy bias is applied and $\boldsymbol{q}_{\text{new}}$ is $\boldsymbol{q}_{\text{bias}}$ itself (Lines 1-2). Otherwise, $\hat{\boldsymbol{v}}_{\text{new}}$ is steered by the direction $\hat{\boldsymbol{v}}_{\text{field}}$ of the first-order synergies and a random direction $\hat{\boldsymbol{v}}_{\text{rand}}$ (Line 12).

```
Algorithm 10: EXTEND
    Input  : Grid $\mathcal{G}$ and configurations $\boldsymbol{q}_{\text{init}} \in \mathcal{G}$, $\boldsymbol{q}_{\text{bias}} \in \mathcal{C}$
    Output: Configuration $\boldsymbol{q}_{\text{new}}$
1:  if $\|\boldsymbol{q}_{\text{bias}} - \boldsymbol{q}_{\text{init}}\| \leq \epsilon$ then
2:      $\boldsymbol{q}_{\text{new}} \leftarrow \boldsymbol{q}_{\text{bias}}$
3:  else
4:      $\hat{\boldsymbol{v}}_{\text{field}} \leftarrow \text{FOS}(\boldsymbol{q}_{\text{init}})$
5:      if ROOTEDATGOAL($\mathcal{G}$) then
6:          $\hat{\boldsymbol{v}}_{\text{field}} \leftarrow -\hat{\boldsymbol{v}}_{\text{field}}$
7:      if RAND01( ) $< P_{\text{bias}}$ then
8:          $\hat{\boldsymbol{v}}_{\text{rand}} \leftarrow$ UNITVECTOR($\boldsymbol{q}_{\text{bias}} - \boldsymbol{q}_{\text{init}}$)
9:      else
10:         $\boldsymbol{q}_{\text{parent}} \leftarrow \mathcal{G}$.PARENT($\boldsymbol{q}_{\text{init}}$)
11:         $\hat{\boldsymbol{v}}_{\text{rand}} \leftarrow$ RANDDIR($\boldsymbol{q}_{\text{init}} - \boldsymbol{q}_{\text{parent}}$)
12:     $\hat{\boldsymbol{v}}_{\text{new}} \leftarrow$ NEWDIR($\hat{\boldsymbol{v}}_{\text{rand}}, \hat{\boldsymbol{v}}_{\text{field}}$)
13:     $\boldsymbol{q}_{\text{new}} \leftarrow \boldsymbol{q}_{\text{init}} + \epsilon\, \hat{\boldsymbol{v}}_{\text{new}}$
14: if COLLISIONFREE($\boldsymbol{q}_{\text{near}}, \boldsymbol{q}_{\text{new}}$) then
15:     ADDSEGMENT($\mathcal{G}, \boldsymbol{q}_{\text{init}}, \boldsymbol{q}_{\text{new}}, \|\boldsymbol{q}_{\text{bias}} - \boldsymbol{q}_{\text{new}}\|$)
16:     return $\boldsymbol{q}_{\text{new}}$
17: else
18:     return $\emptyset$
```

- $\hat{\boldsymbol{v}}_{\text{field}}$ is the velocity of the vector field associated with $\boldsymbol{q}_{\text{init}}$ (Line 4) and it is computed, as described above, with the first-order synergies and the synergy cells. Notice that when the tree being extended is rooted at $\boldsymbol{q}_{\text{goal}}$, i.e. ROOTEDATGOAL($\mathcal{G}$) returns `true`, the tree grows backwards, i.e. from $\boldsymbol{q}_{\text{start}}$ towards $\boldsymbol{q}_{\text{goal}}$, and then $\hat{\boldsymbol{v}}_{\text{field}}$ must be reversed (Lines 5-6).
- $\hat{\boldsymbol{v}}_{\text{rand}}$ points towards $\boldsymbol{q}_{\text{bias}}$ with a probability P_{bias} (Lines 7-8). Otherwise, $\hat{\boldsymbol{v}}_{\text{rand}}$ is some random unit vector that satisfies $\hat{\boldsymbol{v}}_{\text{rand}} \cdot (\boldsymbol{q}_{\text{init}} - \boldsymbol{q}_{\text{parent}}) \geq 0$ (see Fig. 4.14), where $\boldsymbol{q}_{\text{parent}}$ is the parent node of $\boldsymbol{q}_{\text{init}}$ (Lines 10-11). The function UNITVECTOR($\boldsymbol{v}$) returns $\frac{\boldsymbol{v}}{\|\boldsymbol{v}\|}$ if $\|\boldsymbol{v}\| \neq 0$ and it returns $\boldsymbol{v}$ otherwise.
- $\hat{\boldsymbol{v}}_{\text{new}}$ is a combination of $\hat{\boldsymbol{v}}_{\text{rand}}$ and $\hat{\boldsymbol{v}}_{\text{field}}$ (Line 12), see Fig. 4.14 and Eq. (4.11). Finally, $\boldsymbol{q}_{\text{new}}$ is the configuration at distance ϵ from $\boldsymbol{q}_{\text{init}}$ in the direction of $\hat{\boldsymbol{v}}_{\text{new}}$ (Line 13).
- If the motion from $\boldsymbol{q}_{\text{init}}$ to $\boldsymbol{q}_{\text{new}}$ is collision-free (Line 14), it is added to the grid $\mathcal{G}$, using the distance between $\boldsymbol{q}_{\text{bias}}$ and $\boldsymbol{q}_{\text{new}}$ as an overestimation of the measure d (Line 15). In this case, $\boldsymbol{q}_{\text{new}}$ is returned; otherwise, $\boldsymbol{q}_{\text{new}}$ is not added to the grid and $\emptyset$ is returned.

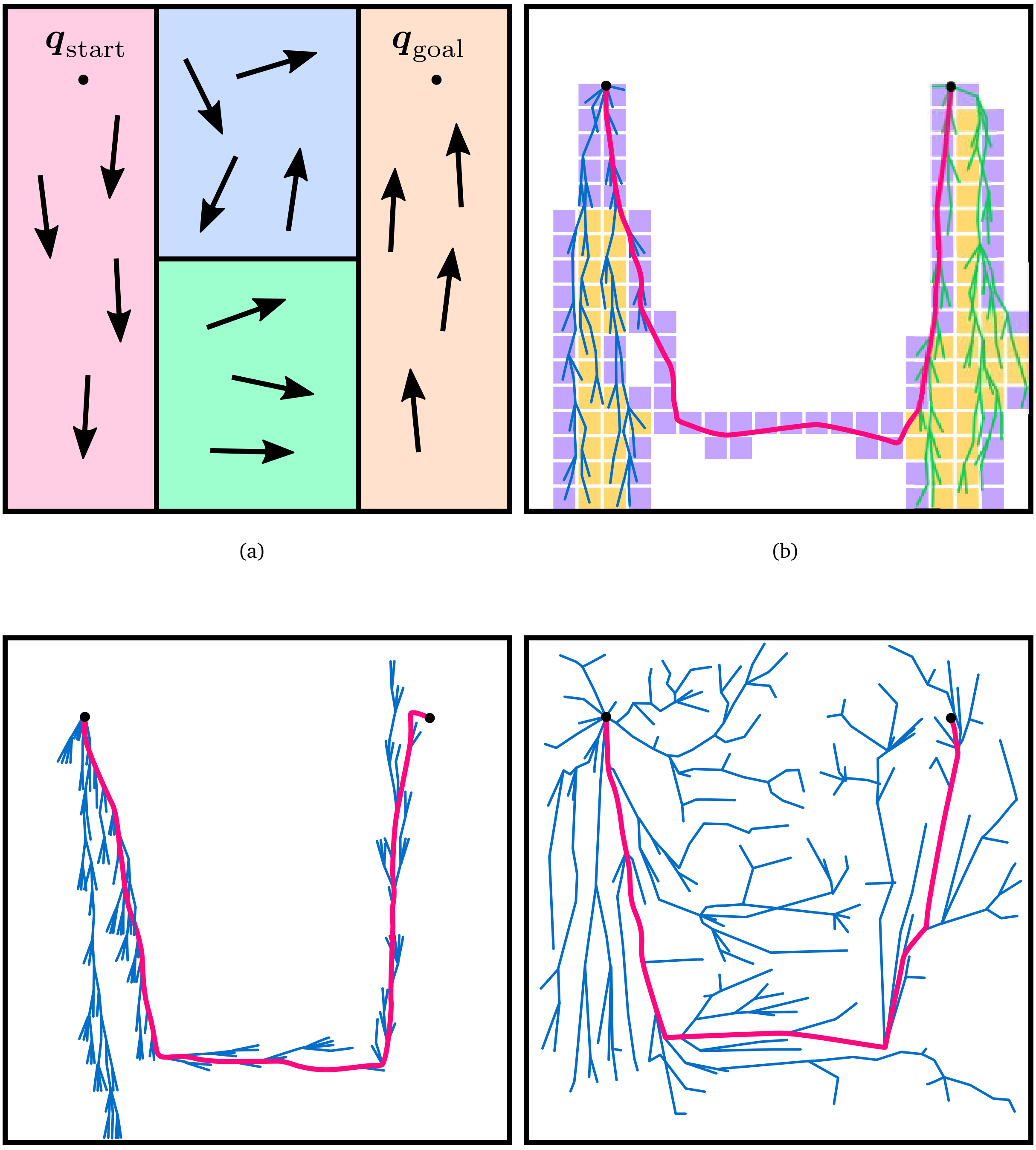
Good results are obtained setting P_{bias} to 0.05, being ϵ problem-dependant. Notice that the dimension k of B_k is chosen such that the variance related to the first k zero-order synergies surpasses the 95% of the total variance. The same criterion is followed with the number p of first-order synergies used to compute $\hat{\boldsymbol{v}}_{\text{field}}$. As suggested by Suçan and Kavraki (2012), the size of the tree cells is chosen so that 20 cells fit along each axis of B_k.

Note that each configuration in the configuration trees has a non-zero probability to be used to expand the trees and that the trees can grow in any direction. Therefore, the introduced planner is probabilistically complete. Nevertheless, the FOS-BKPIECE is not asymptotically optimal. The proposed planning algorithm has a complexity similar to the one of the KPIECE planner and it ensures that a solution path avoiding obstacles and self-collisions is found (if one exists) due to its probabilistic completeness, even if the synergies try to guide the trees towards invalid solutions. In addition, the speedup that can be achieved using multiple processing cores could make FOS-BKPIECE fast and accurate enough to be applicable in real-time motion planning of complex robotic systems.

It must be remarked that, in Section 4.2, the zero-order synergies have been used to select a lower-dimensional subspace in $\mathcal{C}$ and narrow the search of a solution trajectory to this subspace. In contrast, in this section the motion planning is performed in the whole configuration space and this lower-dimensional subspace B_k is only used to estimate the coverage of $\mathcal{C}$. Besides, in that previous section, the configuration tree has been grown uniformly in any direction (lying in B_k) while here the growth of the configuration tree is guided by first-order synergies (in the whole $\mathcal{C}$).

4.4.3 Approach validation

For illustrative purposes, a simple example has been set up consisting of a mobile robot moving in an obstacle-free square of side length 1 m. The start configuration $\boldsymbol{q}_{\text{start}}$ is at the top-left corner and the goal configuration $\boldsymbol{q}_{\text{goal}}$ is at the top-right corner. Four regions with different first-order synergies have been artificially defined (see Fig. 4.15-a): the first-order synergies point downwards in the left region, rightwards in the middle-bottom region, and upwards in the right region. However the synergies do not establish a clear direction of movement in the middle-top region. In this simple example, the robot has 2 DOFs, B_k is $\mathcal{C}$ itself and all the first-



(a) (b)

(c) (d)

Figure 4.15: Conceptual example: Four regions has been artificially imposed over $\mathcal{C}$, each one with different synergy directions (denoted by arrows). A mobile robot must go from $\boldsymbol{q}_{\text{start}}$ to $\boldsymbol{q}_{\text{goal}}$, preferably along the directions of the first-order synergies (a). Trees and solutions paths obtained with the FOS-BKPIECE (b), VF-RRT (c) and RRT*(d) planners are shown. The paths are depicted by red lines. Interior and exterior tree cells are filled in orange and purple, respectively.

Table 4.9: Average results of the motion planning of the conceptual example.

Planner	Success rate [%]	Planning time [s]	# Planning iterations	Upstream criterion	Solution length [m]
FOS-BKPIECE	100	0.125	452	0.513	3.831
FOS-KPIECE	100	0.298	775	0.494	3.880
VF-RRT	100	0.406	812	0.491	3.914
RRT*	100	5	992	0.505	3.517

order synergies associated to each synergy cell are used to generate the vector-field of desired velocities (i.e. $n = k = p = 2$). The problem has been solved with the FOS-BKPIECE and with a non-bidirectional version of the proposed planner, called FOS-KPIECE. In addition, the original VF-RRT and the RRT* have been also used. In this case, the parameters of the VF-RRT have been empirically set and the RRT* has been modified to minimise $\mathcal{U}$, see Eq. (4.13).

Some of the obtained solution paths are shown in Fig. 4.15. It can be appreciated how the configuration trees grow with a greater pace in the sense defined by the first-order synergies of each region, thus encountering a good quality solution. Note that the configuration trees of the FOS-BKPIECE and of the VF-RRT remain close to the solution path. However with the RRT* the configuration trees spread completely over $\mathcal{C}$, with the consequent waste of time. Table 4.9 shows the average results after 100 executions of the mentioned algorithms, with the planning time limited to 5 s. All the planners obtain similar upstream criterion values, being $\mathcal{U}$ a little bit higher with the FOS-BKPIECE due to its greedy attempt to connect the trees; but it is the fastest planner thanks to its bidirectionality (3 and 40 times faster than the VF-RRT and the RRT* planners, respectively). However, the non-bidirectional proposed algorithm FOS-KPIECE is still faster than the others. Notice that the planner RRT* always uses the maximum allowed time, even if it succeeded, because when a valid solution is found the planner still tries to find a better one until the time is over. Besides, it is important to highlight that, with a greater time limit, the planner RRT* would obtain the best $\mathcal{U}$ values.

The planning of the movements of the MADAR dual-arm robot, described in Subsection 1.3.1, is used for a real example of the proposed planning procedure. Hence, for the planning of the dual-arm movements the configuration space $\mathcal{C}$ has dimension $n = 12$ (for all the experiments presented below). From the captured data of the tasks described in Subsection 4.1, the zero- and first-order synergies are computed and B is found, i.e. the region of $\mathcal{C}$ containing the zero-order synergies. Then, based on synergy differences, B is split into synergy cells (as those predefined in the conceptual example) and then, using the computed synergies, a vector field of desired velocities is obtained. Thereby, for the *Assembly* task, B_k has dimension 2 (i.e. $k = 2$, see Subsection 4.2.1), the corresponding B is split into 21 synergy cells and the generated vector field is called $\boldsymbol{f}_{\mathrm{A}}$. For the *Free-movement* task, B_k has dimension 7, the corresponding B is split into 64 synergy cells and the generated vector field is called $\boldsymbol{f}_{\mathrm{F}}$. Note that for both tasks, the number p of first-order synergies used to generate the synergy-base vector-field depends on the synergy cell and it is the the minimum number of first-order synergies in a given synergy cell, taken in order, to cover at least the 95% of the sample variance.

The motions have been planned for the next cases (see task descriptions in Subsection 4.1):

1) *Assembly* task following the directions of $\boldsymbol{f}_{\mathrm{A}}$.
2) *Assembly* task following the directions of $\boldsymbol{f}_{\mathrm{F}}$.
3) *Bottle* task following the directions of $\boldsymbol{f}_{\mathrm{F}}$.

Notice that, except in the first case, the motion planning has been solved following the movement directions of another task. Thereby, it is tested the utility of $\boldsymbol{f}_{\mathrm{F}}$ as a general-purpose vector field in case a task-specific vector field is not available (i.e. when the task has not been demonstrated). As in the conceptual example, the planners that have been used are the proposed FOS-BKPIECE and FOS-KPIECE, a fine-tuned version of the original VF-RRT, and an RRT* modified to minimise the upstream criterion $\mathcal{U}$. For the first two cases, representative solution paths obtained with the FOS-BKPIECE planner are shown in Fig. 4.16, both in the simulation environment and with the real dual-arm robot. Videos of the paths shown in Fig. 4.16 are available in sir.upc.edu/projects/fos/index.html. For the three studied cases, Table 4.10 shows the average results of the motion planning after 100 executions and with a time limit of 100 s. Note that the solution length is measured in $\mathcal{C}$, as the sum of joint motions, and that the valid motion rate is also collected, i.e. the proportion of iterations in which no collisions occur and the tree actually grows. It can be appreciated that the paths that best follow the human movements, i.e. the ones with the lowest $\mathcal{U}$, are obtained with the RRT* but at the expense of a prohibitive planning time. The fastest planner is the proposed FOS-BKPIECE, without significantly impairing the upstream criterion. The planner FOS-KPIECE is still faster than the VF-RRT and RRT*, and obtains similar $\mathcal{U}$ values. For the Assembly task, lower $\mathcal{U}$ values are obtained with the task-specific vector field

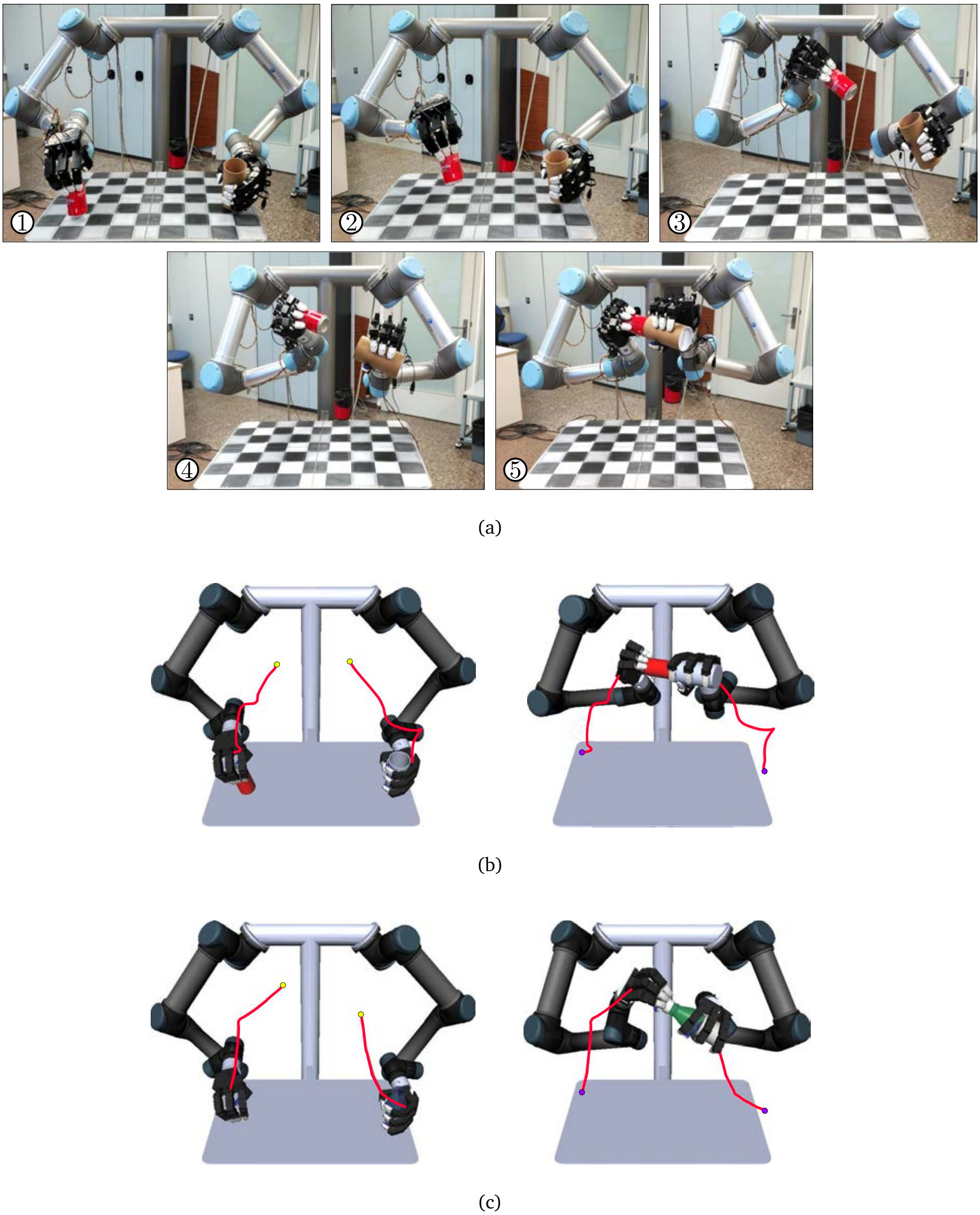


Figure 4.16: Solution paths obtained with the FOS-BKPIECE: Snapshots of the path execution with the real dual-arm robot for the *Assembly* task following $\boldsymbol{f}_{\mathrm{F}}$ (a); and start and goal configurations in the simulation environment, showing the translational planned path, for the *Assembly* task following $\boldsymbol{f}_{\mathrm{A}}$ (b), and the *Bottle* task following $\boldsymbol{f}_{\mathrm{F}}$ (c).

Table 4.10: Average results of the motion planning of the application problems (1-3) using the planners FOS-BKPIECE (a), FOS-KPIECE (b), VF-RRT (c) and RRT* (d).

Problem/ Planner		Success rate [%]	Planning time [s]	# Planning iterations	Upstream criterion	Solution length [m]	Valid motion rate [%]
1)	a)	100	0.489	154	3.234	3.862	91.69
	b)	100	0.665	115	3.197	4.135	93.25
	c)	100	3.688	290	3.174	4.090	72.09
	d)	100	100	14036	3.042	3.818	55.29
2)	a)	100	0.367	88	5.047	4.279	90.64
	b)	100	0.886	216	4.954	4.404	92.30
	c)	100	2.970	232	4.746	4.368	74.45
	d)	100	100	14423	4.462	4.509	54.77
3)	a)	100	0.406	94	4.912	4.244	85.60
	b)	100	0.829	224	4.735	4.378	94.33
	c)	100	2.158	206	4.707	4.105	79.71
	d)	100	100	13737	4.690	4.795	54.55

than with $\boldsymbol{f}_\mathrm{F}$. However, the paths of both tasks solved with the vector field $\boldsymbol{f}_\mathrm{F}$ maintain the human appearance (see Fig. 4.16).

5

Dealing with mobile-base and hand motions

The versatility of the robotic manipulators, either with one or two arms, is augmented significantly when they are mounted on a mobile platform. Such combined systems are commonly known as mobile manipulators and are able to perform dexterous manipulation tasks in larger and more cluttered workspaces than a fixed-base manipulator due to its redundancy and translating DOFs. However, to fully use the advantages offered by a mobile manipulator, it is necessary to understand how to properly and effectively plan its motions.

This chapter focuses on the motions of a mobile manipulator previous to a bimanual manipulation task (i.e. the motions of the robot while, first, moving to the place where the object to be manipulated stands, and then grasping the object). In the same fashion humans do, the robotic arms and the mobile base are coordinated in the approaching phase (with minimum hand motions or no movements at all). Afterwards, once the robot is positioned, the hand and the arm must coordinate themselves to properly grasp the object in the final phase (with minimum motions of the mobile base or no movements at all). In order to plan these robot motions, an extension of the approach proposed in the previous chapter is introduced in the following sections, to consider additionally:

(a) The movements of both arms coordinated with the movements of the mobile base, performing approaching motions (Section 5.1).

(b) The movements of the arm coordinated with the movements of the hand, performing grasping motions (Section 5.2).

5.1 Motion planning using synergies of approaching motions

The human arm-movements have been studied in hand-pointing motions (Chen et al., 2016) and hand-reaching motions (Sreenivasa et al., 2012; Albrecht et al., 2011). Nevertheless, the combined movements of the arms and the translational DOFs have not been deeply investigated yet in dual-arm tasks neither used in the motion planning of bimanual tasks using a mobile base. In order to fill this gap in the literature, this section studies how the position and the orientation of the robot base affects the synergies of a mobile anthropomorphic dual-arm robot. The final goal is to find a human-like coordination of the robot translational movements and the arms movements while approaching a table to perform a dual-arm manipulation task. This coordination can be then exploited in the motion-planning process, e.g. to improve its performance or to obtain human-like motions.

The main steps of the proposed approach are the following:

(a) Human movements are captured, and then mapped to the robotic system, while a human operator walks towards a table and solves manipulation tasks in front of it (see Subsection 5.1.1). It is important to remark that this work is not focused on biped movements, but only on the upper-limb motions.

(b) The captured movements are then analyzed to extract the existing correlations between the robot position, its orientation and the configuration of the arms. The variations of these correlations are studied and the upper-body synergies are computed from the mapped robot configurations (see Subsection 5.1.2). In addition, the Cartesian space is then discretized into different regions based on the changes in the computed synergies and the observed correlations.

(c) A motion-planning algorithm is finally introduced. This planner takes profit of the synergies in the different regions of the Cartesian space, such that the coordinated movements of the arms are similar to the ones of a human being (see Subsection 5.1.3). The proposed planning algorithm is then tested with a real mobile anthropomorphic dual-arm robotic system (see Subsection 5.1.4).

5.1.1 Motion capture

The focus is set on the approaching movements previous to the execution of bimanual manipulation tasks. For this purpose, the movements of a human operator walking towards a table and grasping two cylinders placed on pedestals have been captured, since the knowledge extracted from these movements can later be used to plan, for instance, the motion of the manipulator going to grasp a glass and a soda can from a table and to serve a beverage or any other approaching movements previous a bimanual task. Then, the human operator starts the experiment standing in a position and orientation parametrized by the distance ρ, the azimuth angle ϕ and the angle of the torso joint θ_t; and the positions of the cylinders are determined by the heights h_l and h_r, the constant distance d, and the angle ψ (see Fig. 5.1). The final position and orientation of the human operator is not fixed, it is only required to face the table and grasp both cylinders at the same time. Since it is not computationally feasible to cover the whole parameter space, the following illustrative values have been independently used in the experiments: $\rho \in \{2,\ 3\}$ m, $\phi \in \{-\frac{\pi}{6},\ 0,\ \frac{\pi}{6}\}$ rad, $\theta_t \in \{-\frac{\pi}{4},\ 0,\ \frac{\pi}{4}\}$ rad, $\psi \in \{-\frac{\pi}{6},\ 0,\ \frac{\pi}{6}\}$ rad, and $h_l, h_r \in \{1,\ 1.5\}$ m.

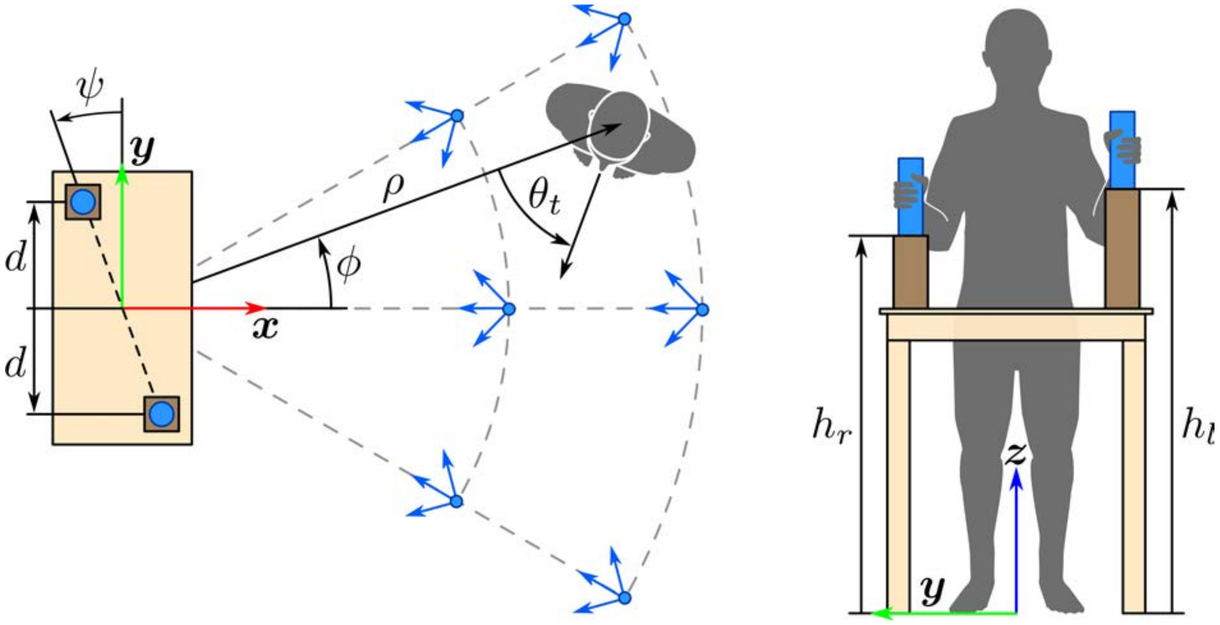


Figure 5.1: Experiment performed to capture the human movements: Top view at the start time (left) and front view at the end time (right). The layout of the start positions and orientations is also shown.

Notice that the combination of these values produces a total of 216 different experiment in which the 3D position of the shoulders, elbows and wrists of the human operator have been recorded using optical markers, as described in Subsection 1.3.2. Besides, two more markers have been attached to each palm to capture the wrist orientations.

Once the human movements have been captured, they are mapped to the robotic system applying the mapping proposed in Subsection 3.1.4. Thereby, each captured sample (i.e. the position of the shoulder, the elbow and the wrist of each arm plus both wrist orientations w.r.t. the table frame) is mapped to a robot configuration $\boldsymbol{q}$. This robot configuration contains the robot position $\boldsymbol{\chi} = [x, y]$, relative to the table frame, and the robot upper-body configuration $\boldsymbol{\theta}$, defined as the concatenation of the value of the torso joint θ_t and the left and right arm configurations $\boldsymbol{\theta}_\mathrm{l}$ and $\boldsymbol{\theta}_\mathrm{r}$, respectively. Note that the torso joint θ_t is not a real joint but this rotation is achieved with movements of the mobile base.

5.1.2 Motion analysis

The upper-body synergies, i.e. the observed correlations between the joints of the arms and the torso, are computed running a Principal Component Analysis (PCA) over the set of mapped upper-body configurations $\boldsymbol{\theta}$, which returns a new basis of the upper-body configuration space whose axes represent a *synergy*. It is observed that the upper-body synergies depend on the robot position $\boldsymbol{\chi}$. Therefore, the Cartesian space is recursively partitioned into sectors of annuli centered on the table to take into account these variations. The splitting radii ρ and angles ϕ are chosen such that the upper-body synergies associated to each annular sector are significantly different to the ones from the neighboring sectors (detailed below). Thereby, given the parent set Q of robot configuration samples $\boldsymbol{q}$ lying within a given annular sector to be partitioned, let:

- $Q_\rho^- = \{\boldsymbol{q} \in Q \mid x^2 + y^2 \leq \rho^2\}$ and $Q_\rho^+ = \{\boldsymbol{q} \in Q \mid x^2 + y^2 > \rho^2\}$ be the descendant sets of Q (if splitting the sector by ρ) in which the robot is respectively closer/farther to the table than a given distance ρ.

- $Q_\phi^- = \{\boldsymbol{q} \in Q \mid y \leq x \tan\phi\}$ and $Q_\phi^+ = \{\boldsymbol{q} \in Q \mid y > x \tan\phi\}$ be the descendant sets of Q (if splitting the sector by ϕ) in which the robot is seen from the table more on the right/left than a given azimuth angle ϕ, respectively.

Similar to what has been considered in the Subsection 4.4.1, let a partition dividing a sector be valid if each descendant sector contains at least 5000 configuration samples of Q and has an aspect ratio less than 1:5, i.e. the longer side of the resulting sector measures in the $\rho - \phi$ space at most five times the shorter side. These values have been empirically chosen. Nevertheless, as in the cases in previous chapter, the sensibility of the procedure with respect to these parameters is not high, thus their values are not a critical issue. Besides, the task-likeness introduced in Subsection 3.2.1 is used in this section to measure how similar are two given sets Q_A and Q_B of configuration samples, referring this measure as $\mathcal{L}(Q_A, Q_B)$. However, it is important to remark that, in this case, only the upper-body part $\boldsymbol{\theta}$ of the robot configurations $\boldsymbol{q}$ is taken into account to compute $\mathcal{L}(Q_A, Q_B)$. Then, the best position to divide a sector is the one that minimizes the objective function f defined as the maximum likeness between the parent set Q and its descendant sets, i.e. $f = \max(\mathcal{L}(Q, Q_\rho^-), \mathcal{L}(Q, Q_\rho^+))$ if splitting by ρ, or $f = \max(\mathcal{L}(Q, Q_\phi^-), \mathcal{L}(Q, Q_\phi^+))$ if splitting by ϕ. Hence, a given partition is better than others if the others are non-valid or they have a greater value of f.

The Cartesian-space discretization is stored into a *synergy tree*, which is a k-d tree $\mathcal{T}_{KD}$ with:

a) Non-leaf nodes, with the partition location (i.e. whether the sector is split by ρ or by ϕ and at which value) and the synergy subtrees before and after the partition.

b) Leaf nodes, composed of the upper-body synergies of the mapped movements lying in the corresponding sector.

The synergy tree is built by recursively applying the following procedure, outlined in Algorithm 11. First, the best partition distance ρ^* and angle ϕ^* are computed (Lines 1-2). Then, the sector is split up by the best valid partition, among the ones defined by ρ^* (Lines 3-5) and ϕ^* (Lines 6-8), if a valid partition exists. This procedure is then recursively self-invoked (Line 11), until no valid partitions are found (Lines 9-10).

The introduced partition procedure has been applied to the set of configurations containing the data captured as described in the previous section plus the same data reflected in the *y*-axis of the table frame (see Fig. 5.1) in order to artificially increase the number of samples. Thereby, the partition of the Cartesian-space shown in Fig. 5.2 is obtained, which results to be formed by 10 symmetrically-distributed annular sectors. Notice that the synergies are similar in regions far from the table independently of ϕ (i.e. one big sector exists in which the arms are mostly at

Algorithm 11: SYNERGYTREE

```
Input  : Set of configurations $Q$
Output: Synergy tree $\mathcal{T}_{KD}$
$\rho^* \leftarrow \underset{\rho \in (r_{\min}, r_{\max})}{\arg\min} \max\left(\mathcal{L}(Q, Q_\rho^-), \mathcal{L}(Q, Q_\rho^+)\right)$
$\phi^* \leftarrow \underset{\phi \in (\phi_{\min}, \phi_{\max})}{\arg\min} \max\left(\mathcal{L}(Q, Q_\phi^-), \mathcal{L}(Q, Q_\phi^+)\right)$
if partitioning by $\rho^*$ is valid and better than partitioning by $\phi^*$ then
    $\mathcal{T}_{KD}$.PARTITION $\leftarrow \rho^*$
    $\mathcal{T}_{KD}$.SUBTREES $\leftarrow \left(\text{SYNERGYTREE}(Q_{\rho^*}^-), \text{SYNERGYTREE}(Q_{\rho^*}^+)\right)$
else if partitioning by $\phi^*$ is valid and better than partitioning by $\rho^*$ then
    $\mathcal{T}_{KD}$.PARTITION $\leftarrow \phi^*$
    $\mathcal{T}_{KD}$.SUBTREES $\leftarrow \left(\text{SYNERGYTREE}(Q_{\phi^*}^-), \text{SYNERGYTREE}(Q_{\phi^*}^+)\right)$
else
    $\mathcal{T}_{KD}$.SYNERGIES $\leftarrow$ SYNERGIES$(Q)$
return $\mathcal{T}_{KD}$
```

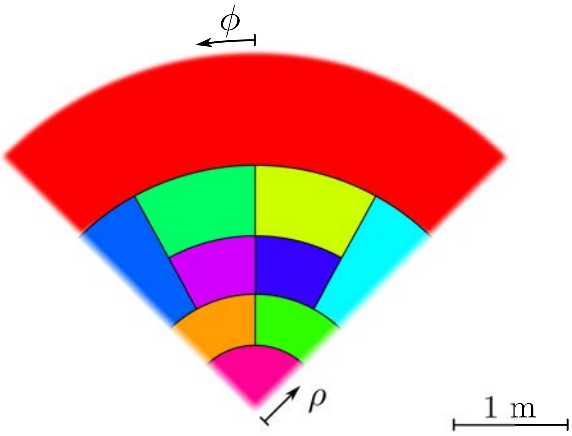


Figure 5.2: Resulting Cartesian-space partitions based on the differences of the upper-body synergies computed from the captured human movements.

rest). While getting closer to the table, the upper-body synergies differ and they are grouped into different sectors, remarking the gradual transition from the free-walk to the grasping phase (i.e. the arms get ready to reach the goal pose). Finally, when the robot is in front of the table grasping the cylinders, the synergies converge to a unique set of synergies.

5.1.3 Motion planning

As explained in Subsection 3.1.3, the upper-body synergies define a n-dimensional box B in the configuration space, where n is the number of upper-body DOFs of the robot (i.e. without considering the translational DOFs of the robot base). Here, $n = 13$ due to the used hardware, i.e. the MADAR robot described in Subsection 1.3.1. The box dimension can be decreased by using only $k < n$ synergies (picking them in decreasing order of the associated sample variance). Despite this simplification, the resulting lower-dimensional box B_k still represents accurately the mapped upper-body configurations. Note that since the Cartesian space is partitioned in sectors based on the synergies, each sector may have a different B_k. Thereby, if the planning of the upper-body motions is performed in the corresponding B_k, the planning complexity is reduced and the obtained motions are similar to the movements mapped from the human operator.

In order to integrate the obtained upper-body synergies and the Cartesian-space discretization into the motion planning, a modification of the RRT-Connect (Kuffner and LaValle, 2000), see Subsection 4.2.2, is proposed. Here, the standard function used to extend a configuration tree $\mathcal{T}$ towards a given target configuration $\boldsymbol{q}_{\text{target}}$, i.e. Algorithm 3, is replaced by the function EXTEND described in Algorithm 12 and depicted in Fig. 5.3. In this function, the tree also grows from the configuration in the tree closest to $\boldsymbol{q}_{\text{target}}$, $\boldsymbol{q}_{\text{near}}$ (Line 1), but the robot position $\boldsymbol{\chi}$ and the upper-body configuration $\boldsymbol{\theta}$ are treated differently, as detailed below. A step, with a maximum length ϵ_{χ}, is taken from the robot position $\boldsymbol{\chi}_{\text{near}}$ towards $\boldsymbol{\chi}_{\text{rand}}$, reaching a new robot

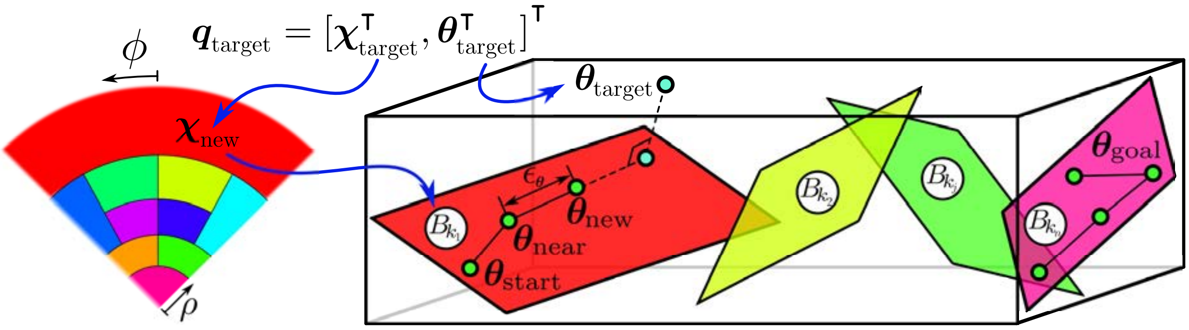


Figure 5.3: A sample tree is steered towards a given robot configuration $\boldsymbol{q}_{\text{target}} = [\boldsymbol{\chi}^{\mathsf{T}}_{\text{target}}, \boldsymbol{\theta}^{\mathsf{T}}_{\text{target}}]^{\mathsf{T}}$. The robot position and upper-body configuration are treated differently. The sample tree reaches a new configuration $\boldsymbol{q}_{\text{new}}$ composed of a robot position $\boldsymbol{\chi}_{\text{new}}$ and a new upper-body configuration $\boldsymbol{\theta}_{\text{new}}$, forced to lie close to the box B_k associated to $\boldsymbol{\chi}_{\text{new}}$.

```
Algorithm 12: EXTEND
    Input : Tree of configurations $\mathcal{T}$ and configuration $\boldsymbol{q}_{\text{target}} = [\boldsymbol{\chi}_{\text{target}}^{\mathsf{T}}, \boldsymbol{\theta}_{\text{target}}^{\mathsf{T}}]^{\mathsf{T}}$
    Output: Configuration $\boldsymbol{q}_{\text{new}} = [\boldsymbol{\chi}_{\text{new}}^{\mathsf{T}}, \boldsymbol{\theta}_{\text{new}}^{\mathsf{T}}]^{\mathsf{T}}$
 1: $\boldsymbol{q}_{\text{near}} = [\boldsymbol{\chi}_{\text{near}}^{\mathsf{T}}, \boldsymbol{\theta}_{\text{near}}^{\mathsf{T}}]^{\mathsf{T}} \leftarrow \textsc{NearestConf}(\mathcal{T}, \boldsymbol{q}_{\text{target}})$
 2: $\boldsymbol{\chi}_{\text{new}} \leftarrow \boldsymbol{\chi}_{\text{near}} + \min(\epsilon_{\boldsymbol{\chi}}, \|\boldsymbol{\chi}_{\text{target}} - \boldsymbol{\chi}_{\text{near}}\|)\,(\boldsymbol{\chi}_{\text{target}} - \boldsymbol{\chi}_{\text{near}})$
 3: $\boldsymbol{\Delta\theta} \leftarrow \textsc{Project}\big(\boldsymbol{\theta}_{\text{target}}, B_k(\boldsymbol{\chi}_{\text{new}})\big) - \boldsymbol{\theta}_{\text{near}}$
 4: $\boldsymbol{\theta}_{\text{new}} \leftarrow \boldsymbol{\theta}_{\text{near}} + \min(\epsilon_{\boldsymbol{\theta}}, \|\boldsymbol{\Delta\theta}\|)\,\boldsymbol{\Delta\theta}$
 5: $\boldsymbol{q}_{\text{new}} \leftarrow [\boldsymbol{\chi}_{\text{new}}^{\mathsf{T}}, \boldsymbol{\theta}_{\text{new}}^{\mathsf{T}}]^{\mathsf{T}}$
 6: if $\textsc{CollisionFree}(\boldsymbol{q}_{\text{near}}, \boldsymbol{q}_{\text{new}})$ then
 7:     $\textsc{AddSegment}(\mathcal{T}, \boldsymbol{q}_{\text{near}}, \boldsymbol{q}_{\text{new}})$
 8:     return $\boldsymbol{q}_{\text{new}}$
 9: else
10:     return $\emptyset$
```

position $\boldsymbol{\chi}_{\text{new}}$ (Line 2). Then, a step, with a maximum length $\epsilon_{\boldsymbol{\theta}}$, is taken from the upper-body configuration $\boldsymbol{\theta}_{\text{near}}$ towards the projection of $\boldsymbol{\theta}_{\text{rand}}$ onto $B_k(\boldsymbol{\chi}_{\text{new}})$, i.e. the lower-dimensional box B_k spanned by the synergies of the Cartesian-space region containing $\boldsymbol{\chi}_{\text{new}}$, instead of moving towards $\boldsymbol{\theta}_{\text{rand}}$ (as it would be done in the standard procedure). Hence, a new upper-body configuration $\boldsymbol{\theta}_{\text{new}}$ is reached (Lines 3-4). The computed $\boldsymbol{\chi}_{\text{new}}$ and $\boldsymbol{\theta}_{\text{new}}$ are joined to compose $\boldsymbol{q}_{\text{new}}$ (Line 5). Finally, as in the standard procedure, only if the rectilinear segment connecting $\boldsymbol{q}_{\text{near}}$ and $\boldsymbol{q}_{\text{new}}$ is collision-free (Line 6), the segment is added to the tree and $\boldsymbol{q}_{\text{new}}$ is returned (Lines 7-8). Otherwise, $\emptyset$ is returned (Line 10).

5.1.4 Approach validation

For illustrative purposes, the motions of the MADAR robot have been planned using the synergies computed from the captured human movements and the planner introduced above. The robot must go from a start configuration in the neighborhood of a table to a goal configuration in which the robot is ready to grasp two cylinders lying on the table, see Fig. 5.4. The robot must avoid collisions and perform human-like motions, which do not have to be exactly the movements used to compute the synergies. The robot motions have been planned with: (a) the proposed planner, and, (b) the standard RRT-Connect, both implemented within The

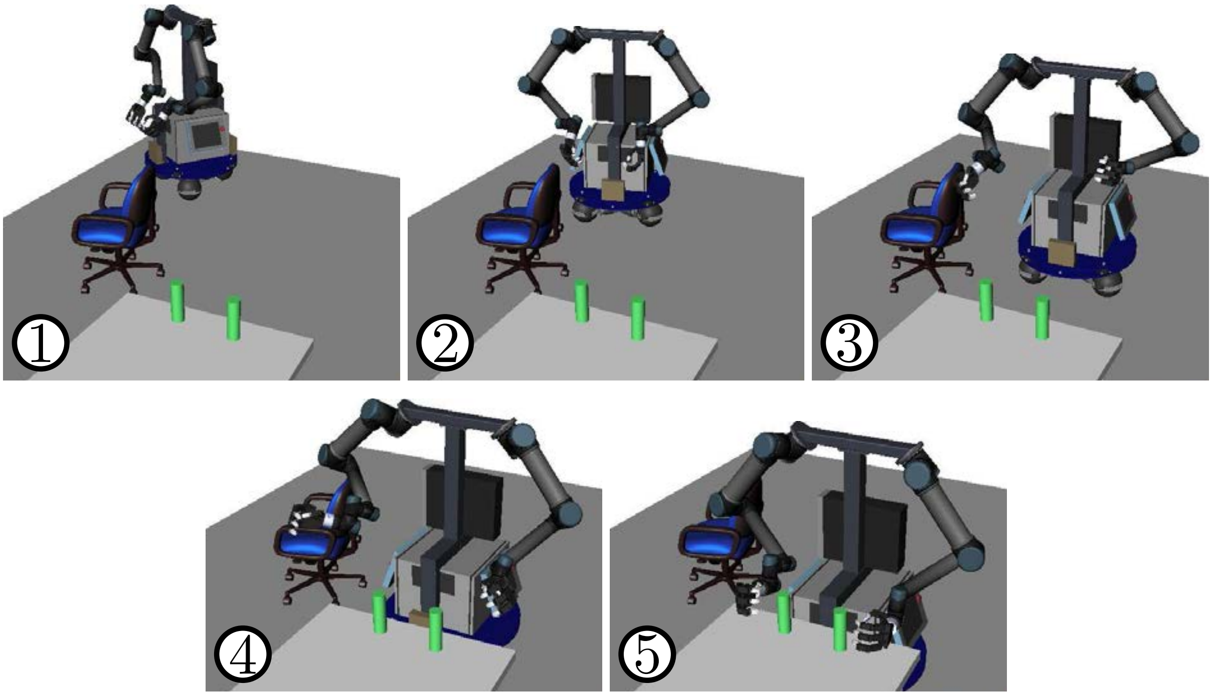


Figure 5.4: Chronologically ordered snapshots of a planned path for the mobile anthropomorphic dual-arm robot.

Table 5.1: Average results of the motion planning using the proposed approach (a) and the standard RRT-Connect (b).

Case	Success rate	Planning time	# Planning iterations	# Collision checks	Valid motion rate	Solution length
a)	100%	2.923 s	290	2156	74.09%	4.378 rad
b)	100%	11.378 s	1940	6532	63.32%	4.731 rad

Kautham Project (described in Subsection 1.4.1) and run in a 2.13-GHz Intel 2, 4-GB RAM PC. A maximum planning time of 30 s was allowed and if a path was not found within this time, the execution was marked as a failure. After 100 executions, Table 5.1 shows the average values of the success rate, the planning time, the number of iterations and collision checks, the rate of valid motions (i.e. the ratio of iterations in which the trees actually grow) and the path length (measured in $\mathcal{C}$ space as the weighted sum of accumulated rotated angles of the wheels and the arm joints along the path). Fig. 5.4 illustrates a representative solution path obtained with the proposed motion-planning algorithm. A video of a planned path can be found at sir.upc.edu/projects/kautham/videos/SIMPAR2018.mp4.

5.2 Motion planning using synergies of hand-arm grasping motions

This section aims to plan the motions of a hand-arm robotic system trying to mimic the hand-arm movements that a human does to pick an object up. For this, it is proposed a characterization of the synergies existing in the human grasping motions (Ficuciello, 2019), considering different grasp types (Cutkosky, 1989; Dai et al., 2013). A method to identify the different phases in the grasping motions is presented and used to obtain an all-purpose pre-grasp set of synergies as well as a set of grasping synergies for each grasp type. The approach pursues both efficiency and human-likeness in the obtaining of the hand-arm movements. Thus, it introduces a new sampling-based motion planner that considers different potential grasps at the same time and steers the path towards the direction of the synergies. The main key points of the proposed approach are the following:

a) The motions of a human operator performing different types of grasp on several objects are captured and then mapped to the robot whose motions are aimed to be planned (Subsection 5.2.1).
b) From the captured information, the different grasping phases in the demonstrations are identified and the set of hand synergies existing in the human grasping motions are computed (one per grasp type). Thereby, the complexity of the different DOF-couplings in the human motions is accurately explained in a simple manner (Subsection 5.2.2).
c) A bidirectional sampling-based planner is designed to bias the tree growth towards the directions of the computed synergies and to reduce the dimension of the search space, being this dimension-reduction process dependent on the synergies of the grasp type to be performed. Hence, human-like movements are obtained efficiently (Subsection 5.2.3). The proposed planning algorithm is tested with a real anthropomorphic hand-arm robotic system (Subsection 5.2.4).

5.2.1 Motion capture

As stated above, human motions are used as a reference to obtain human-like movements of a hand-arm robotic system picking a given object up. Many types of human grasps are gathered

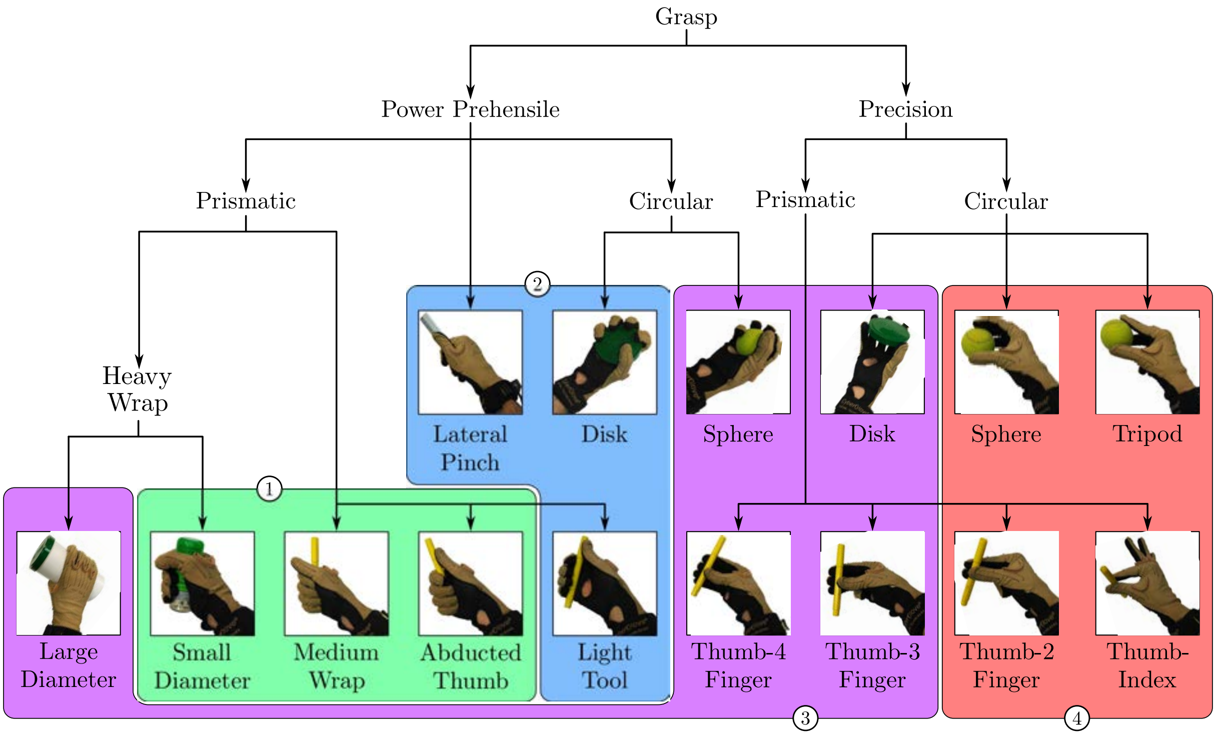


Figure 5.5: The 15 force-closure grasps whose movements have been captured, classified in a tree structure, adapting the grasp taxonomy of Cutkosky (1989), and grouped into grasp families, 1 to 4, according to the grouping of Dai et al. (2013).

in the grasp taxonomy of Cutkosky (1989), which classifies the grasps depending on the object size and on the task to be performed. Although this classification is not complete, and there exist more extensive grasp classifications (Feix et al., 2016), it is detailed enough for the purposes considered in this work. Besides, Dai et al. (2013) updated the taxonomy of Cutkosky and analyzed, from a different perspective, the entire grasping trajectory of the fingers and not only the grasping configuration (i.e. the final configuration), proving that the grasp types can be grouped naturally into a set of four consistent grasp families (see Fig. 5.5). This family-grouping is used here to adapt the planning process according to the grasp being performed, also considering several potential grasp types simultaneously.

Thereby, using a *Cyberglove*, the sensorized glove with a 50 Hz sampling frequency described in Subsection 1.3.2, the motions of a human operator are recorded performing 15 different grasp types (see Fig. 5.5) on 9 objects (see Fig. 5.6), with 12 repetitions per grasp type and starting off from a comfortable position in front of the object. This implies 180 demonstrations and more

Figure 5.6: Set of objects used in the experiments.

than 15,000 configuration samples (where each sample contains 22 measurements read from the glove that describe completely the positions of the finger joints). Once the samples have been captured, they are mapped to the robotic hand. This mapping depends on the kinematic structure and particularities of the used robotic hand. Nevertheless, slight differences in the mapping strategy do not affect excessively the overall performance of the proposed approach. Here, a robotic hand-arm system composed of a 6-DOF *UR5* robotic arm equipped with a 16-DOF *Allegro Hand* is used (see Subsection 1.3.1). For this robotic hand-arm, a simple mapping have been used, consisting on applying a joint-to-joint mapping for those robotic joints which are present in the human hand and a fingertip-position mapping for the joints that are different from the human ones (see a more detailed description of the mapping in Subsection 3.1.4).

5.2.2 Motion analysis

Two phases are observed in the mapped grasping motions (see Fig. 5.7). During the first phase, called *pre-grasp phase*, the trajectories of the hand joints are common motions opening the hand similarly in all the executions, regardless of the grasp type performed. Then, there is a moment in which the demonstrated trajectories begin to differ and specialize according to the type of grasp being carried out. This is the *grasp phase* itself. Nevertheless, the transition from one phase to the other is diffuse and does not occur at the same time for all the demonstrations. Hence, the transition time is obtained as follows.

Let Q be the set of hand configurations mapped from a given grasping demonstration, and,

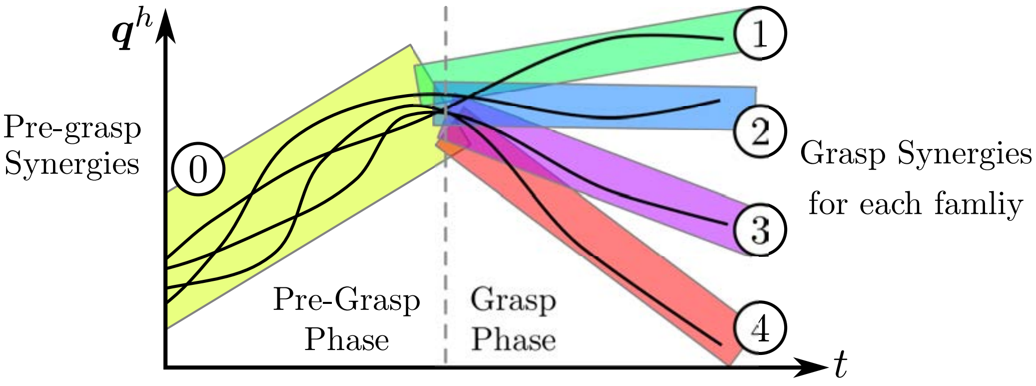


Figure 5.7: Hypothetical mapped trajectories on the hand-configuration space, divided into pre-grasp and grasp phases, to obtain the common pre-grasp synergies (0) and the grasp synergies of each family (1 to 4).

for a given time instant t, let Q_t^- and Q_t^+ be the sets of configurations in Q captured before and after t, respectively. A measure $\mathcal{L}(Q_t^-, Q_t^+)$ of how similar are these two sets Q_t^- and Q_t^+ can be obtained using task-likeness index presented in Subsection 3.2.1 (i.e. $\mathcal{L}(Q_t^-, Q_t^+)$ accounts for the overlapping of the distributions of the configurations in the two sets). Then, the time t marking the transition between the two phases is defined as the one minimizing $\mathcal{L}(Q_t^-, Q_t^+)$, i.e. the time t makes the two sets Q_t^- and Q_t^+ as different as possible. Thereby, the pre-grasp and grasp phases have been identified in the 180 mapped trajectories.

All the pre-grasp phases have been grouped and used to compute the *pre-grasp synergies*. On the other hand, the grasp phases have been grouped according the grasp family which each demonstrated grasp belongs to, then, a set of *grasp synergies* has been computed for each grasp family shown in Fig. 5.5. In this way, the pre-grasp synergies explain the hand motions in the pre-grasp phase in all the grasps, and each set of grasp synergies models the hand motions in the corresponding grasp family. Table 5.2 shows the accumulated sample variance for the obtained set of synergies.

Remember that as explained in Section 3.1, the hand synergies are obtained running a Principal Component Analysis (PCA) over the set of mapped hand configurations and each axis of the new basis of the hand configuration space returned by the PCA represents a *synergy*. For a robotic hand with n DOFs (here $n = 16$), the hand synergies define an n-dimensional box B centered at the barycenter of the configurations used to obtain the synergies and with each side aligned with a synergy, see Subsection 3.1.3. The dimension of the box B can be reduced by

Table 5.2: Accumulated sample variance as a function of the number k of chosen synergies, for the common pre-grasp phase and the grasp phase of each of the demonstrated grasp families. Bold values indicate the minimum number of synergies, taken in order, to cover at least the 95% of the sample variance.

k	Pre-Grasp	Grasp Family			
		1	2	3	4
1	65.575%	79.474%	64.234%	63.280%	88.568%
2	77.795%	86.125%	81.877%	84.238%	91.955%
3	84.586%	91.442%	88.091%	91.428%	94.921%
4	90.316%	94.015%	92.225%	94.377%	**96.606%**
5	93.260%	**96.229%**	**95.108%**	**96.394%**	97.676%
6	**95.996%**	97.665%	96.781%	97.664%	98.685%
7	97.262%	98.315%	97.850%	98.569%	99.160%
8	98.165%	98.802%	98.834%	99.027%	99.449%
9	98.901%	99.218%	99.241%	99.467%	99.624%
10	99.318%	99.515%	99.562%	99.656%	99.769%
11	99.593%	99.741%	99.749%	99.803%	99.882%
12	99.817%	99.886%	99.882%	99.896%	99.931%
13	99.933%	99.955%	99.964%	99.963%	99.967%
14	100%	100%	100%	100%	100%
15	100%	100%	100%	100%	100%
16	100%	100%	100%	100%	100%

using only $k<n$ synergies (picking them in decreasing order) such that k is the minimum value making the accumulated variance be above a confidence level of 95%. Thus, the dimension k of the resulting lower-dimensional boxes, called B_k, is 4 or 5 for the grasp phase, depending on the grasp family (see bold values in Table 5.2). For the pre-grasp phase, 6 synergies are needed (a little bit greater, as it was expected, since the movements of all the grasp families are included). Despite the simplification (reducing from dimension 16 to 6 or less), the boxes B_k still represent accurately the mapped hand motions. Thereby, if the hand motions are planned in the corresponding B_k, the planning complexity is reduced and the obtained motions are still similar to the movements mapped from the human.

5.2.3 Motion planning

Let $\mathcal{C}$ be the configuration space of the whole robot, let $\boldsymbol{q} = \left[\boldsymbol{q}^{a\mathsf{T}},\ \boldsymbol{q}^{h\mathsf{T}}\right]^{\mathsf{T}} \in \mathcal{C}$ be a robot configuration defined as an arm configuration $\boldsymbol{q}^a$ concatenated with a hand configuration $\boldsymbol{q}^h$, and let $G = (\boldsymbol{q}^h, \chi_o^h)$ be a grasp composed of the hand configuration $\boldsymbol{q}^h$ and the hand pose χ_o^h relative

Algorithm 13: EXTEND

```
Input  : Graph of configurations $(E, V)$ and configuration $\boldsymbol{q}_{\text{target}}$
Output: Configuration $\boldsymbol{q}_{\text{new}}$
1:  $\boldsymbol{q}_{\text{near}} \leftarrow$ NEARESTCONF$(V, \boldsymbol{q}_{\text{target}})$
2:  if $\|\boldsymbol{q}_{\text{target}} - \boldsymbol{q}_{\text{near}}\| \leq \epsilon$ then
3:      $\boldsymbol{q}_{\text{new}} \leftarrow \boldsymbol{q}_{\text{target}}$
4:  else
5:      $\boldsymbol{q}_{\text{proj}} \leftarrow$ PROJECT$\big(\boldsymbol{q}_{\text{target}}, (E, V)\big)$
6:      $\boldsymbol{q}_{\text{new}} \leftarrow \boldsymbol{q}_{\text{near}} + \min(\epsilon, \|\boldsymbol{q}_{\text{proj}} - \boldsymbol{q}_{\text{near}}\|) \frac{\boldsymbol{q}_{\text{proj}} - \boldsymbol{q}_{\text{near}}}{\|\boldsymbol{q}_{\text{proj}} - \boldsymbol{q}_{\text{near}}\|}$
7:  if COLLISIONFREE$(\boldsymbol{q}_{\text{near}}, \boldsymbol{q}_{\text{new}})$ then
8:      ADDSEGMENT$\big((E, V), \boldsymbol{q}_{\text{near}}, \boldsymbol{q}_{\text{new}}\big)$
9:      return $\boldsymbol{q}_{\text{new}}$
10: else
11:     return $\emptyset$
```

to the object at the grasping time. The proposed planner is supplied with a collision-free start configuration $\boldsymbol{q}_{\text{start}}$ of the whole robot, the object pose χ_o^r relative to the robot, and a set $\{G_i\}$ of grasps. The introduced algorithm is based on the RRT-Connect (Kuffner and LaValle, 2000), see Subsection 4.2.2. However, it has been modified here to:

a) *Extend the trees following the synergies:* In order to integrate the synergies into the motion planning, the standard EXTEND function extending the tree in RRT-based planners (see Algorithm 3 in Subsection 4.2.2) is replaced here by the function EXTEND described in Algorithm 13. As in the classic method, a single step is performed from $\boldsymbol{q}_{\text{near}}$, the configuration in the graph closest to the desired target configuration $\boldsymbol{q}_{\text{target}}$ (Line 1), reaching a new configuration $\boldsymbol{q}_{\text{new}}$. If the segment connecting $\boldsymbol{q}_{\text{near}}$ and $\boldsymbol{q}_{\text{new}}$ is collision-free, $\boldsymbol{q}_{\text{new}}$ is returned (Line 8); otherwise, $\emptyset$ is returned (Line 10). However, here, $\boldsymbol{q}_{\text{new}}$ is computed differently, using the synergies. A step, with a maximum length ϵ, is not oriented towards the desired $\boldsymbol{q}_{\text{target}}$ (as it would be done in the standard procedure) but towards its projection $\boldsymbol{q}_{\text{proj}}$ onto the lower-dimensional box spanned by the synergies (Lines 5-6), as shown in Fig. 5.8. It should be remarked that, in the computation of $\boldsymbol{q}_{\text{proj}}$, the arm component of $\boldsymbol{q}_{\text{target}}$ remains the same. Nevertheless, the hand component of $\boldsymbol{q}_{\text{target}}$ is projected onto the lower-dimensional box B_k of synergies associated with the root of

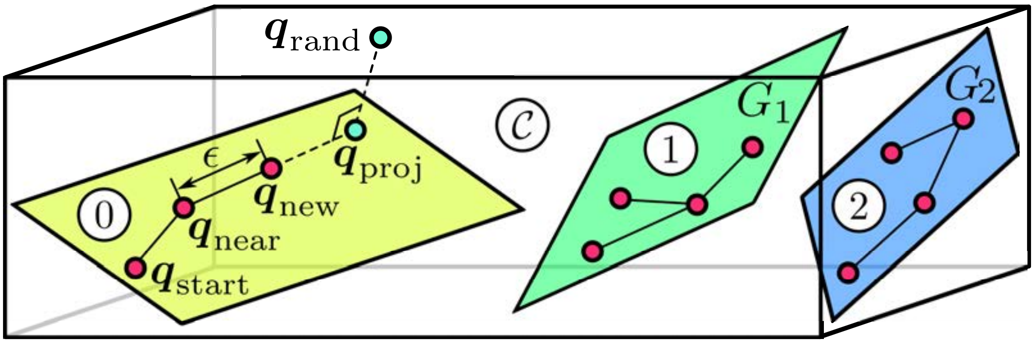


Figure 5.8: Motion planning representation in $\mathcal{C}$-space: sample trees rooted at the start configuration $\boldsymbol{q}_{\text{start}}$ and the grasps G_1 and G_2, growing close to the associated synergy lower-dimensional boxes (0 to 2), while steering a given configuration $\boldsymbol{q}_{\text{near}}$ towards a random $\boldsymbol{q}_{\text{rand}}$ and reaching $\boldsymbol{q}_{\text{new}}$.

the tree containing $\boldsymbol{q}_{\text{near}}$. Thereby, if $\boldsymbol{q}_{\text{near}}$ belongs to the tree rooted at $\boldsymbol{q}_{\text{start}}$, $\boldsymbol{q}_{\text{target}}$ is projected onto the box of pre-grasp synergies, otherwise, $\boldsymbol{q}_{\text{near}}$ belongs to a tree rooted at a certain G_i and, hence, $\boldsymbol{q}_{\text{target}}$ is projected onto the box of synergies associated with G_i. However, if $\boldsymbol{q}_{\text{target}}$ is close to $\boldsymbol{q}_{\text{near}}$, $\boldsymbol{q}_{\text{new}}$ is replaced by $\boldsymbol{q}_{\text{target}}$ (Line 3), so that if the two trees are close to be connected, the guideline to follow the synergies may be relaxed.

b) *Deal with multigoal queries:* The proposed planner, outlined in Algorithm 14, maintains two graphs, each one denoted by a pair formed by a set of edges E and a set of vertices V. One of this graphs represents a tree rooted at the start configuration $\boldsymbol{q}_{\text{start}}$ (Line 1), and the other contains the trees rooted each one at a grasp configuration (Lines 2-6), as shown in Fig. 5.8. These grasp configurations are computed by, first, solving the arm inverse kinematics (IK) given the grasps G_i to be performed and the object pose χ_o^r (Line 4) and, then, rejecting those cases that do not have an IK-solution or that imply collisions (Line 5). Thus, in each iteration, one of the graphs is steered towards a random configuration $\boldsymbol{q}_{\text{rand}}$ (uniformly sampled in $\mathcal{C}$), reaching a configuration $\boldsymbol{q}_{\text{new}}$ (Lines 8-10). Then, the connection between the graphs is attempted. Note that the EXTEND method explained above returns $\emptyset$ if a collision is found, and consequently the graph is not extended.

c) *Connect the trees in a less greedy fashion:* In the classic RRT-Connect, the trees are connected greedily by extending one of trees directly until the other tree is reached or a collision occurs. However, here, in order to obtain an smoother connection, both graphs are, in alternation and successively, extended towards the last added configuration in the other

Algorithm 14: HAND-ARM SYNERGY-BASED MOTION PLANNER

```
    Input  : Start configuration $\boldsymbol{q}_{\text{start}} \in \mathcal{C}$, object pose $\boldsymbol{\chi}_o^r$, and set of grasps $\{G_i\}$
    Output: Collision-free path $\mathcal{P}$ connecting $\boldsymbol{q}_{\text{start}}$ and one $G_i$
 1: $(E_a, V_a) \leftarrow (\emptyset, \boldsymbol{q}_{\text{start}})$
 2: $(E_b, V_b) \leftarrow (\emptyset, \emptyset)$
 3: forall $G_i$ do
 4:     $\boldsymbol{q}_{\text{goal}}^i \leftarrow$ InvKin$(G_i, \boldsymbol{\chi}_o^r)$
 5:     if $\boldsymbol{q}_{\text{goal}}^i \neq \emptyset$ and CollisionFree$(\boldsymbol{q}_{\text{goal}}^i)$ then
 6:         $V_b \leftarrow V_b \cup \boldsymbol{q}_{\text{goal}}^i$
 7: while not StopCriteria$\big((E_a, V_a), (E_b, V_b)\big)$ do
 8:     $\boldsymbol{q}_{\text{rand}} \leftarrow$ RandConf( )
 9:     $\boldsymbol{q}_{\text{new}} \leftarrow$ Extend$\big((E_a, V_a), \boldsymbol{q}_{\text{rand}}\big)$
10:     $(E_a, V_a) \leftarrow \big(E_a \cup (\boldsymbol{q}_{\text{near}}, \boldsymbol{q}_{\text{new}}), V_a \cup \boldsymbol{q}_{\text{new}}\big)$
11:     while $\boldsymbol{q}_{\text{new}} \neq \emptyset$ do
12:         if $V_a \cap V_b \neq \emptyset$ then
13:             return Path$\big((E_a, V_a), (E_b, V_b)\big)$
14:         else
15:             Swap$\big((E_a, V_a), (E_b, V_b)\big)$
16:             $\boldsymbol{q}_{\text{new}} \leftarrow$ Extend$\big((E_a, V_a), \boldsymbol{q}_{\text{new}}\big)$
17:             $(E_a, V_a) \leftarrow \big(E_a \cup (\boldsymbol{q}_{\text{near}}, \boldsymbol{q}_{\text{new}}), V_a \cup \boldsymbol{q}_{\text{new}}\big)$
18:     Swap$\big((E_a, V_a), (E_b, V_b)\big)$
19: return $\emptyset$
```

graph (Lines 15-17), until the graphs are connected and the solution path is returned (Line 13). In case the steering process fails (Line 11), the graphs swap their roles (Line 18) and the whole process is repeated until a solution is found or some termination condition is satisfied (Line 7), like for instance surpassing a maximum planning time, number of iterations or memory allocation. In this case, an empty path $\emptyset$ is returned (Line 19).

5.2.4 Approach validation

The motions of the MADAR robot, see Subsection 1.3.1, have been planned for illustrative purposes. The robot is located in front of a bookshelf and, starting off from a natural standing pose, it must grasp a cylinder standing on one of the shelves (see Fig. 5.9). Besides, the robot

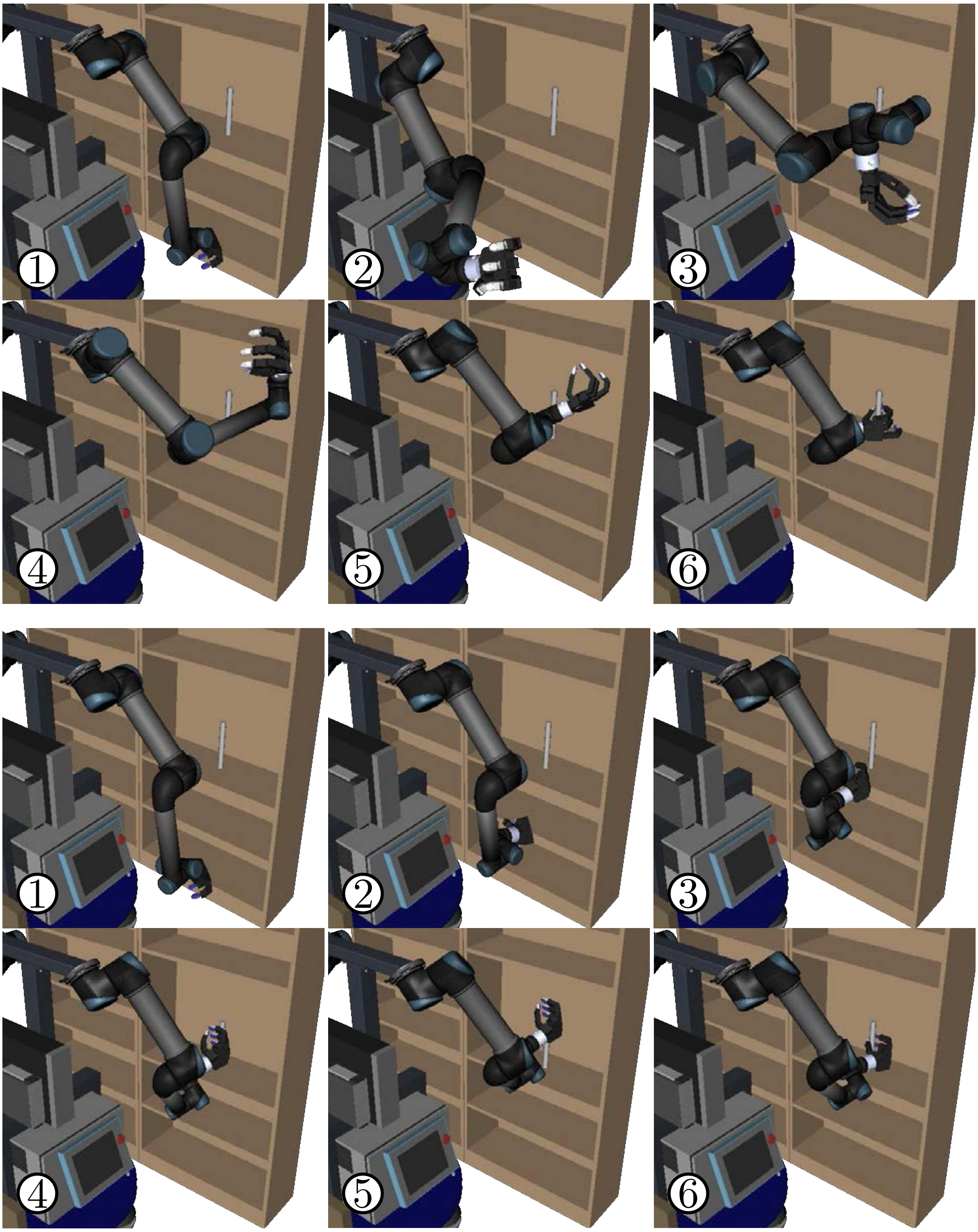


Figure 5.9: Chronologically ordered snapshots of paths obtained with an standard RRT-Connect (top) and with the proposed procedure using the proper synergies for each grasp type (bottom).

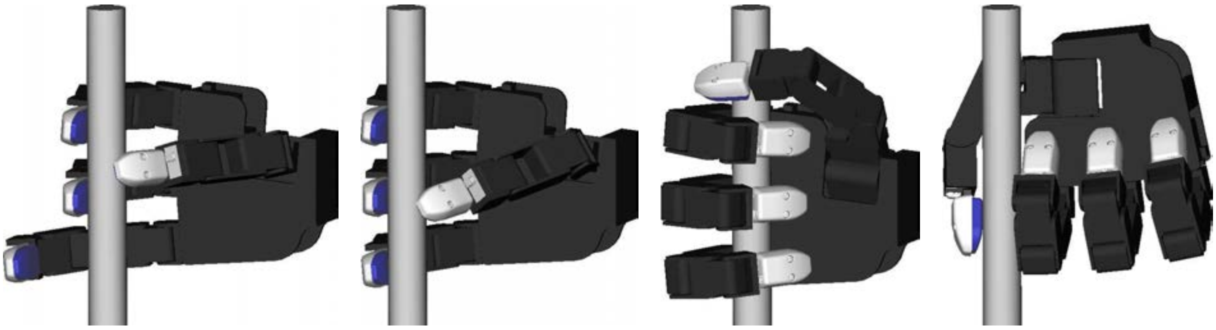


Figure 5.10: Examples of different grasp types used in the motion planning, each one from a different grasp family: *Thumb-2 Finger*, *Thumb-3 Finger*, *Medium Wrap* and *Lateral Pinch* (from left to right, respectively).

must perform human-like motions while avoiding self-collisions and collisions with the bookshelf and the cylinder. For this, the planning algorithm is provided with the exact position of the cylinder and with a set of 8 force-closure grasps G_i (from different grasp types and grasp families, see Fig. 5.10). This information can be obtained, for instance, from the vision system on the robot and a grasp generator, respectively. In order to evaluate and compare the performance of the proposed approach, three planners have been benchmarked:

a) A standard RRT-Connect, modified to tackle multi-goal queries, planning without using synergies.
b) The proposed approach, planning using the proper grasp synergies related to the grasp type to be performed.
c) The proposed approach, planning with the grasp synergies and the grasp families intentionally mismatched (i.e. each grasp family has been randomly associated with the synergies of another grasp family).

At each execution, each planner is provided with a single start configuration (obtained by lightly modifying at random the pose of the robot base) and a set of different goal configurations (one per each of the 8 different grasps G_i).

The experiments introduced above have been implemented within The Kautham Project (see Subsection 1.4.1) and run in a 2.13-GHz Intel 2, 4-GB RAM PC. A maximum planning time of 100 s is considered for each planning instance. Thereby, if a path is not found within this time, the execution is marked as a failure. Table 5.3 shows the average values, after 100 executions, of the success rate, the planning time, the number of iterations and collision checks, the rate of

Table 5.3: Average results of the motion planning when running the classic RRT-Connect (a) and the proposed approach with the proper (b) and with mismatched grasp synergies (c).

Case	Success rate	Planning time	#Planning iterations	# Collision checks	Valid motion rate	Solution length	Human-Likeness $\mathcal{H}$
a)	97%	51.80 s	1834	32231	68.3%	14.18 rad	73.6%
b)	100%	6.21 s	274	10649	80.0%	7.79 rad	83.1%
c)	100%	11.79 s	484	13667	75.3%	8.35 rad	81.9%

valid motions (i.e. the ratio of iterations in which the trees actually grow), the path length (measured in $\mathcal{C}$ as the weighted sum of accumulated joint movements along the path), and the path human-likeness (see Subsection 3.2.2). The human-likeness index computes the misalignment of a path with respect to some given reference human movements (García et al., 2017a). Here, natural free-movements of the operator while moving freely the fingers in an unconstrained way (i.e. without performing any specific task), trying to cover the whole hand workspace, are used as a reference.

On the one hand, it can be noticed from the simulation results that effectively the proposed planning approach is up to an order of magnitude faster than the standard RRT-Connect algorithm. In fact, the motion planner succeeds within the time constraints for the 100% of the executions only when the proposed approach is used, either when the grasp synergies are properly associated with the selected grasps or when they are mismatched. It can be stated that the use of synergies clearly reduces the planning time, this is because the solution is enforced to lie close to the lower-dimensional boxes B_k, focusing the search efforts close to the demonstrated movements (which belong to a set of demonstrated feasible solutions), thus accelerating the connection of the trees and, thereby, reducing the number of iterations and collision checks needed to find a solution. In addition, since the grasp synergies are obtained from feasible movements, the probability of obtaining collision-free robot configurations increases when using synergies (see *Valid motion rate* in Table 5.3), reducing greatly the computational time. The results also show that even when no correct grasping synergies were used, i.e. case (c), the benefits of using synergies are still evident. In this case, the planning time is slightly penalized, however, it is still a better option than not using synergies at all.

On the other hand, the proposed planning procedure produces movements of the robotic system that look more natural and human-like (see *Human-Likeness* in Table 5.3), since the grasp synergies are obtained from human demonstrations and the human-likeness is preserved within the planning process. Besides the numerical results, Fig. 5.9 shows representative solution paths for cases (a) and (b), and the higher human-likeness of the solution obtained with the proposed approach can be noticed.

6

Conclusions

This chapter summarizes the main contributions of this doctoral thesis and discusses the future work. Finally, the publications generated from this work are listed, including the relation between each publication and the sections of this document.

6.1 Contributions

This doctoral thesis has addressed the problem of planning the movements of anthropomorphic dual-arm robots, equipped with dexterous mechanical hands and mounted on a mobile base. In order to simultaneously obtain robot motions in an efficient way and with human-like appearance, the existence of synergies in the human dual-arm movements has been investigated and used in novel sampling-based algorithms to reduce the computational cost of the planning process and derive human-like robot movements. This approach has been considered in the motion planning of the coordinated movements of:

(a) Only the arms in bimanual tasks, either using synergies of zero- and first-order and synergies over potential fields.

(b) The arms and the base in approaching motions, using upper-body synergies.

(c) An arm and its hand in grasping motions, using hand-arm synergies.

The main contributions of this work are:

- The combined use of synergies of different order to obtain human-like robot motions has been introduced.
- An index of the likeness between two tasks, using the synergies of each of them, has been proposed (i.e. the similarity of the movements needed to solve two given tasks).
- A path-quality measure, based on first-order synergies, has been introduced and its use has been proposed to measure quantitatively the human-likeness of a given robot trajectory.
- A method for mapping human movements to mobile anthropomorphic dual-arm robots, with or without redundancy in the robotic arms, and considering the motions of the mechanical hand and the mobile base has been proposed.
- A method to compute robot synergies from real human movements has been introduced. Besides, a method to determine the lower-dimensional box spanned by the synergies has been presented, with the minimum dimension and the minimum size that guarantees that a given percentage of the sample distribution of the demonstrated movements is covered.
- An automatic partition method has been defined to optimally divide the relevant parts of the configuration space, determined by the zero-order synergies, into cells where the first-order synergies are significantly different.
- Novel motion-planning procedures have been proposed, and tested in simulation and with the real robot. The procedures were used to plan the coordinated movements of both arms in bimanual tasks using zero-order synergies, synergies over potential fields, and first-order synergies.
- A procedure was developed and tested both in simulation and with the real robot, to consider:
    (a) The coordinated movements of both arms and the mobile base in approaching motions, using upper-body synergies.
    (b) The coordinated movements of the arm and the hand in grasping motions, using hand-arm synergies.

- Several software libraries have been implemented. For instance, a low-level module, called Low Level Universal Robot Control (LLUR-Control), has been implemented to command the UR5 robotic arms in the MADAR robot using the API provided by the manufacturer. A controller for the Allegro Hands mounted on the MADAR robot has been also implemented and the library controlling the mobile platform of the MADAR robot have been improved. In addition, all the motion-planning algorithms presented in this doctoral thesis have been added to The Kautham Project to be publicly available. Finally, all these modules have been integrated using ROS.

6.2 Future work

The planning of the coordinated movements of mobile anthropomorphic dual-arm robots involves several levels of coordination (e.g. arm-arm, arm-hand, torso-base ...) and new features of practical relevance can be addressed. In this work, some novel solutions to the presented problem have been proposed, nevertheless, there are still several interesting topics that could be treated in future work:

- The extension of the proposed concepts to arms forming a closed kinematic chain, like, for instance, when moving an object simultaneously grasped by both hands.
- The extension of the proposed approach to the velocity space, to plan the motions of the mobile base coordinated with the motions of the arms, as well as to plan the motions of the hand coordinated with the motions of the arm.
- The extension of the proposed planning method to the kynodynamic motion planning, i.e. using synergies at force/torque level.
- The coordination of the robot base, arms and hands all at the same time, either using zero- or first-order synergies.
- The application of the proposal to a system composed of several collaborative robots.
- The optimization of the presented human-likeness index while solving the motion planning, in order to better mimic human task executions.

- A deep analysis of the typology of tasks that can be done with a dual-arm system, in such a way that a taxonomy or clusters of tasks could be generated using the task-likeness index based on synergies proposed in this work.
- The use of learning procedures to improve the corresponding set of synergies, each time the task is executed.

6.3 Derived publications

The following published papers were generated during the course of this thesis.

Journal publications

- García, N., Suárez, R., Rosell, J., June 2017. Task-Dependent Synergies for Motion Planning of an Anthropomorphic Dual-Arm System. IEEE Trans. Robotics 33 (3), 756–764 **(JCR 2017 JIF: 4.264 Q1)**.

  The paper contributions were described in Subsection 3.2.1 and Section 4.2.
- García, N., Rosell, J., Suárez, R., Oct. 2017. Motion Planning by Demonstration With Human-Likeness Evaluation for Dual-Arm Robots. IEEE Trans. Systems, Man, and Cybernetics: Systems, 10.1109/TSMC.2017.2756856 **(JCR 2017 JIF: 5.135 Q1)**.

  The paper contributions were described in Subsection 3.2.2 and Section 4.3.

Refereed conference publications

- Suárez, R., Rosell, J., García, N., May 2015. Using Synergies in Dual-Arm Manipulation Tasks. In: Proc. IEEE Int. Conf. Robotics and Automation (ICRA). pp. 5655–5661.

  The paper contributions were described in Sections 3.1 and 4.2.
- García, N., Rosell, J., Suárez, R., Sept. 2015. Motion planning using first-order synergies. In: Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS). pp. 2058–2063.

  The paper contributions were described in Sections 3.1 and 4.4.
- García, N., Suárez, R., Rosell, J., Sept. 2015. HG-RRT*: Human-Guided optimal random trees for motion planning. In: Proc. IEEE Int. Conf. Emerging Technologies and Factory

Automation (ETFA) **(Fumio Harashima Best Paper Award in Emerging Technologies)**.
The paper contributions were described in Section 4.3.

- García, N., Suárez, R., Rosell, J., July 2017. Application of RRT algorithms in the planning of optimal movements in robotics. In: Proc. Metaheuristics International Conference (MIC). pp. 953–962.
  The paper contributions were described in Section 2.1.

- García, N., Suárez, R., Rosell, J., July 2017. First-Order Synergies for Motion Planning of Anthropomorphic Dual-Arm Robots. IFAC-PapersOnLine 50 (1), 2247 – 2254, IFAC World Congress.
  The paper contributions were described in Section 4.4.

- García, N., Rosell, J., Suárez, R., May 2018. Modeling human-likeness in approaching motions of dual-arm autonomous robots. In: Proc. IEEE Int. Conf. Simulation, Modeling and Programming for Autonomous Robots (SIMPAR). pp. 43–48.
  The paper contributions were described in Section 5.1.

- García, N., Suárez, R., Rosell, J., Oct. 2018. Planning Hand-Arm Grasping Motions with Human-Like Appearance. In: Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS). pp. 3517–3522 **(Finalist IROS Best Application Paper Award)**.
  The paper contributions were described in Section 5.2.

Obtained Awards

- **Fumio Harashima Best Paper Award in Emerging Technologies:** García, N., Suárez, R., Rosell, J., Sept. 2015. HG-RRT*: Human-Guided optimal random trees for motion planning. In: Proc. IEEE Int. Conf. Emerging Technologies and Factory Automation (ETFA).

- **Finalist IROS Best Application Paper Award:** García, N., Suárez, R., Rosell, J., Oct. 2018. Planning Hand-Arm Grasping Motions with Human-Like Appearance. In: Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS). pp. 3517–3522.

Bibliography

Agravante, D., Pagès, J., Chaumette, F., May 2013. Visual servoing for the REEM humanoid robot's upper body. In: Proc. IEEE Int. Conf. Robotics and Automation (ICRA). pp. 5253–5258.

Akgun, B., Stilman, M., Sept. 2011. Sampling Heuristics for Optimal Motion Planning in High Dimensions. In: Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS). pp. 2640–2645.

Albrecht, S., Ramirez-Amaro, K., Ruiz-Ugalde, F., Weikersdorfer, D., Leibold, M., Ulbrich, M., Beetz, M., Oct. 2011. Imitating human reaching motions using physically inspired optimization principles. In: Proc. IEEE-RAS Int. Conf. Humanoid Robots (Humanoids). pp. 602–607.

Albu-Schöffer, A., Haddadin, S., Ott, C., Stemmer, A., Wimböck, T., Hirzinger, G., May 2007. The DLR lightweight robot: Design and control concepts for robots in human environments. Industrial Robot: An Int. J. 34 (5), 376–385.

Argall, B., Chernova, S., Veloso, M., Browning, B., May 2009. A survey of robot learning from demonstration. Robotics and Autnonomous Systems 57 (5), 469–483.

Arslan, O., Tsiotras, P., May 2013. Use of relaxation methods in sampling-based algorithms for optimal motion planning. In: Proc. IEEE Int. Conf. Robotics and Automation (ICRA). pp. 2421–2428.

Artemiadis, P., Katsiaris, P., Kyriakopoulos, K., Nov. 2010. A biomimetic approach to inverse kinematics for a redundant robot arm. Autonomous Robots 29 (3-4), 293–308.

Asfour, T., Gyarfas, F., Azad, P., Dillmann, R., Dec. 2006a. Imitation Learning of Dual-Arm Manipulation Tasks in Humanoid Robots. In: Proc. IEEE-RAS Int. Conf. Humanoid Robots (Humanoids). pp. 40–47.

Asfour, T., Regenstein, K., Azad, P., Schroder, J., Bierbaum, A., Vahrenkamp, N., Dillmann, R., Dec. 2006b. ARMAR-III: An Integrated Humanoid Platform for Sensory-Motor control. In: Proc. IEEE-RAS Int. Conf. Humanoid Robots (Humanoids). pp. 169–175.

Basañez, L., Suárez, R., 2009. Teleoperation. In: Nof, S. (Ed.), Springer Handbook of Automation. Springer-Verlag, pp. 449–468.

Bernstein, N., 1967. The coordination and regulation of movements. Oxford.

Bianchi, M., Salaris, P., Bicchi, A., Apr. 2013a. Synergy-based Hand Pose Sensing: Optimal Glove Design. Int. J. Robotics Research 32 (4), 407–424.

Bianchi, M., Salaris, P., Bicchi, A., Apr. 2013b. Synergy-based Hand Pose Sensing: Reconstruction Enhancement. Int. J. Robotics Research 32 (4), 396–406.

Bicchi, A., Gabiccini, M., Santello, M., 2011. Modeling Natural and Artificial Hands with Synergies. Philosophical Transactions of the Royal Society B 366, 3153–3161.

Bicho, E., Erlhagen, W., L., Louro, Costa e Silva, E., Silva, R., Hipolito, N., Dec. 2011a. A dynamic field approach to goal inference, error detection and anticipatory action selection in human-robot collaboration. New Frontiers in Human Robot Interaction 2, 135–164.

Bicho, E., Erlhagen, W., Louro, L., Costa e Silva, E., Oct. 2011b. Neuro-cognitive mechanisms of decision making in joint action: A human–robot interaction study. Human Movement Science 30 (5), 846–868.

Bohren, J., Rusu, R., Jones, E., Marder-Eppstein, E., Pantofaru, C., Wise, M., Mosenlechner, L., Meeussen, W., Holzer, S., May 2011. Towards autonomous robotic butlers: Lessons learned with the PR2. In: Proc. IEEE Int. Conf. Robotics and Automation (ICRA). pp. 5568–5575.

Calinon, S., D'halluin, F., Sauser, E., Caldwell, D., Billard, A., June 2010. Learning and Reproduction of Gestures by Imitation. IEEE Robotics and Automation Magazine 17 (2), 44–54.

Chen, W., Xiong, C., Yue, S., June 2015. Mechanical Implementation of Kinematic Synergy for Continual Grasping Generation of Anthropomorphic Hand. IEEE/ASME Trans. Mechatronics 20 (3), 1249–1263.

Chen, W., Xiong, C., Yue, S., Mar. 2016. On Configuration Trajectory Formation in Spatiotemporal Profile for Reproducing Human Hand Reaching Movement. IEEE Trans. Cybernetics 46 (3), 804–816.

Chitta, S., Cohen, B., Likhachev, M., May 2010. Planning for autonomous door opening with a mobile manipulator. In: Proc. IEEE Int. Conf. Robotics and Automation (ICRA). pp. 1799–1806.

Ciocarlie, M., Allen, P., July 2009. Hand Posture Subspaces for Dexterous Robotic Grasping. Int. J. Robotics Research 28 (7), 851–867.

Cobos, S., Ferre, M., Sánchez-Urán, M., Ortego, J., Aracil, R., 2010. Human Hand Descriptions and Gesture Recognition for Object Manipulation. Computer Methods in Biomechanical and Biomedical Engineering 13 (3), 307–317.

Cohen, B., Chitta, S., Likhachev, M., May 2012. Search-based planning for dual-arm manipulation with upright orientation constraints. In: Proc. IEEE Int. Conf. Robotics and Automation (ICRA). pp. 3784–3790.

Colasanto, L., Suárez, R., Rosell, J., 2013. Hybrid Mapping for the Assistance of Teleoperated Grasping Tasks. IEEE Trans. Systems, Man and Cybernetics 43 (2), 390–401.

Cutkosky, M. R., 1989. On grasp choice, grasp models, and the design of hands for manufacturing tasks. IEEE Trans. Robotics and Automation 5 (3), 269–279.

Dai, W., Sun, Y., Qian, X., Nov. 2013. Functional analysis of grasping motion. In: IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS). pp. 3507–3513.

d'Avella, A., Portone, A., Fernandez, L., Lacquaniti, F., July 2006. Control of fast-reaching movements by muscle synergy combinations. J. Neuroscience 96 (30), 7791–7810.

Diftler, M., Mehling, J., Abdallah, M., Radford, N., Bridgwater, L., Sanders, A., Askew, R., Linn, D., Yamokoski, J., Permenter, F., Hargrave, B., Piatt, R., Savely, R., Ambrose, R., May 2011. Robonaut 2 - The first humanoid robot in space. In: Proc. IEEE Int. Conf. Robotics and Automation (ICRA). pp. 2178–2183.

Duffy, B., Mar. 2003. Anthropomorphism and the social robot. Robotics and Autnonomous Systems 42 (3–4), 177–190, Socially Interactive Robots.

Edsinger, A., Kemp, C., 2008. Two arms are better than one: A behavior-based control system for assistive bimanual manipulation. Lecture Notes in Control and Information Sciences 370, 345–355.

Elbanhawi, M., Simic, M., Jan. 2014. Sampling-based robot motion planning: A review. IEEE Access 2, 56–77.

Englsberger, J., Werner, A., Ott, C., Henze, B., Roa, M., Garofalo, G., Burger, R., Beyer, A., Eiberger, O., Schmid, K., Albu-Schaffer, A., Nov. 2014. Overview of the torque-controlled humanoid robot TORO. In: Proc. IEEE-RAS Int. Conf. Humanoid Robots (Humanoids). pp. 916–923.

Engø, K., June 2001. On the BCH-formula in so(3). BIT Numerical Mathematics 41 (3), 629–632.

Feix, T., Romero, J., Schmiedmayer, H. B., Dollar, A. M., Kragic, D., Feb. 2016. The GRASP Taxonomy of Human Grasp Types. IEEE Trans. Human-Machine Systems 46 (1), 66–77.

Ficuciello, F., Jan. 2019. Hand-arm autonomous grasping: synergistic motions to enhance the learning process. Intelligent Service Robotics 12 (1), 17–25.

Ficuciello, F., Falco, P., Calinon, S., July 2018. A brief survey on the role of dimensionality reduction in manipulation learning and control. IEEE Robotics and Automation Letters 3 (3), 2608–2615.

Ficuciello, F., Palli, G., Melchiorri, C., Siciliano, B., Apr. 2014. Postural Synergies of the UB Hand IV for Human-like Grasping. Robotics and Autnonomous Systems 62 (4), 515–527.

Fiedler, M., Sept. 1971. Bounds for the determinant of the sum of hermitian matrices. Proc. American Mathematical Society 30, 27–31.

Flash, T., Hogans, N., July 1985. The Coordination of Arm Movements: An Experimentally Confirmed Mathematical Model. J. Neuroscience 5 (7), 1688–1703.

Fornberg, B., Oct. 1988. Generation of Finite Difference Formulas on Arbitrarily Spaced Grids. Math. of Computation 51, 699–706.

Fukuda, T., Michelini, R., Potkonjak, V., Tzafestas, S., Valavanis, K., Vukobratovic, M., Mar. 2001. How far away is "artificial man". IEEE Robotics and Automation Magazine 8 (1), 66–73.

Gabiccini, M., Bicchi, A., Prattichizzo, D., Malvezzi, M., Oct. 2011. On the Role of Hand Synergies in the Optimal Choice of Grasping Forces. Autonomous Robots 31 (2-3), 235–252.

García, N., Rosell, J., Suárez, R., Sept. 2015a. Motion planning using first-order synergies. In: Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS). pp. 2058–2063.

García, N., Rosell, J., Suárez, R., Oct. 2017a. Motion Planning by Demonstration With Human-Likeness Evaluation for Dual-Arm Robots. IEEE Trans. Systems, Man, and Cybernetics: Systems PP (99), 10.1109/TSMC.2017.2756856.

García, N., Rosell, J., Suárez, R., May 2018a. Modeling human-likeness in approaching motions of dual-arm autonomous robots. In: Proc. IEEE Int. Conf. Simulation, Modeling and Programming for Autonomous Robots (SIMPAR). pp. 43–48.

García, N., Suárez, R., Rosell, J., Sept. 2015b. HG-RRT*: Human-Guided optimal random trees for motion planning. In: Proc. IEEE Int. Conf. Emerging Technologies and Factory Automation (ETFA). pp. 1–7.

García, N., Suárez, R., Rosell, J., July 2017b. Application of RRT algorithms in the planning of optimal movements in robotics. In: Proc. Metaheuristics International Conference (MIC). pp. 953–962.

García, N., Suárez, R., Rosell, J., July 2017c. First-Order Synergies for Motion Planning of Anthropomorphic Dual-Arm Robots. IFAC-PapersOnLine 50 (1), 2247 – 2254, IFAC World Congress.

García, N., Suárez, R., Rosell, J., June 2017d. Task-Dependent Synergies for Motion Planning of an Anthropomorphic Dual-Arm System. IEEE Trans. Robotics 33 (3), 756–764.

García, N., Suárez, R., Rosell, J., Oct. 2018b. Planning Hand-Arm Grasping Motions with Human-Like Appearance. In: Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS). pp. 3517–3522.

Geng, T., Lee, M., Hülse, M., 2011. Transferring human grasping synergies to a robot. Mechatronics 21 (1), 272–284.

Gharbi, M., Cortés, J., Siméon, T., July 2008. A sampling-based path planner for dual-arm manipulation. In: Proc. IEEE/ASME Int. Conf. Advanced Intelligent Mechatronics (AIM). pp. 383–388.

Gioioso, G., Salvietti, G., Malvezzia, M., Prattichizzo, D., Aug. 2013. Mapping Synergies From Human to Robotic Hands With Dissimilar Kinematics: An Approach in the Object Domain. IEEE Trans. Robotics 29 (4), 825–837.

Grinyagin, I., Biryukova, E., Maier, M., Oct. 2005. Kinematic and dynamic synergies of human precision-grip movements. J. Neurophysiology 94 (4), 2284–2294.

Harville, D., Sept. 1997. Matrix algebra from a statistician's perspective. Springer, New York.

Huang, Q., Sugano, S., Tanie, K., May 1998. Motion planning for a mobile manipulator considering stability and task constraints. In: Proc. IEEE Int. Conf. Robotics and Automation (ICRA). Vol. 3. pp. 2192–2198.

Huang, Y., Sun, Y., Sept. 2015. Generating manipulation trajectory using motion harmonics. In: Proc. IEEE/RSJ Int. Conf.Intelligent Robots and Systems (IROS). pp. 4949–4954.

Huynh, D., Oct. 2009. Metrics for 3D Rotations: Comparison and Analysis. J. Mathematical Imaging and Vision 35 (2), 155–164.

Infantino, I., Chella, A., Dindo, H., Macaluso, I., 2005. Cognitive Architecture for Robotic Hand Posture Learning. IEEE Trans. Systems, Man and Cybernetics C: Applications and Reviews 35 (1), 42–52.

Jaillet, L., Cortés, J., Siméon, T., Aug. 2010. Sampling-Based Path Planning on Configuration-Space Costmaps. IEEE Trans. Robotics 26 (4), 635–646.

Kalakrishnan, M., Chitta, S., Theodorou, E., Pastor, P., Schaal, S., May 2011. STOMP: Stochastic trajectory optimization for motion planning. In: Proc. IEEE Int. Conf. Robotics and Automation (ICRA). pp. 4569–4574.

Kaneko, K., Harada, K., Kanehiro, F., Miyamori, G., Akachi, K., Sept. 2008. Humanoid robot HRP-3. In: Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS). pp. 2471–2478.

Karaman, S., Frazzoli, E., June 2011. Sampling-based Algorithms for Optimal Motion Planning. Int. J. Robotics Research 30 (7), 846–894.

Kavraki, L., Svestka, P., Latombe, J.-C., Overmars, M., Aug. 1996. Probabilistic Roadmaps for Path Planning in High-Dimensional Configuration Spaces. IEEE Trans. Robotics and Automation 12 (4), 566–580.

Kim, S., Kim, C., Park, J., Oct. 2006. Human-like Arm Motion Generation for Humanoid Robots Using Motion capture Database. In: Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS). pp. 3486–3491.

Ko, I., Kim, B., Park, F., 2014. Randomized path planning on vector fields. Int. J. Robotics Research 33 (13), 1664–1682.

Kondo, T., Amagi, O., Nozawa, T., Oct. 2008. Proposal of Anticipatory Pattern Recognition for EMG Prosthetic Hand Control. In: Proc. IEEE Int. Conf. Systems, Man and Cybernetics. pp. 897–902.

Kuffner, J., LaValle, S., Apr. 2000. RRT-Connect: An efficient approach to single-query path planning. In: Proc. IEEE Int. Conf. Robotics and Automation (ICRA). pp. 995–1001.

Kurazume, R., Hasegawa, T., Oct. 2006. A new index of serial-link manipulator performance combining dynamic manipulability and manipulating force ellipsoids. IEEE Trans. Robotics 22 (5), 1022–1028.

Latombe, J.-C., 1991. Robot Motion Planning. Kluwer Academic Publishers, USA.

LaValle, S., 2006. Planning Algorithms. Cambridge University Press.

Liarokapis, M., Artemiadis, P., Bechlioulis, C., Kyriakopoulos, K., Sept. 2013. Directions, methods and metrics for mapping human to robot motion with functional anthropomorphism: A review. Tech. rep., School of Mechanical Engineering, National Technical University of Athens.

Maldonado, A., Alvarez, H., Beetz, M., Oct. 2012. Improving robot manipulation through fingertip perception. In: Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS). pp. 2947–2954.

Miners, B., Basir, O., Kamel, M., 2005. Understanding Hand Gestures Using Approximate Graph Matching. IEEE Trans. Systems, Man and Cybernetics A: Systems and Humans 35 (2), 239–248.

Neo, E., Sakaguchi, T., Yokoi, K., Kawai, Y., Maruyama, K., Dec. 2006. A Behavior Level Operation System for Humanoid Robots. In: Proc. IEEE-RAS Int. Conf. Humanoid Robots (Humanoids). pp. 327–332.

Oriolo, G., Mongillo, C., Apr. 2005. Motion Planning for Mobile Manipulators along Given End-effector Paths. In: Proc. IEEE Int. Conf. Robotics and Automation (ICRA). pp. 2154–2160.

Palli, G., Melchiorri, C., Vassura, G., Scarcia, U., Moriello, L., Berselli, G., Cavallo, A., Maria, G. D., Natale, C., Pirozzi, S., May, C., Ficuciello, F., Siciliano, B., Apr. 2014. DEXMART Hand: Mechatronic Design and Experimental Evaluation of Synergy-based Control for Human-like Grasping. Int. J. Robotics Research 33 (5), 799–824.

Palmieri, L., Arras, K., May 2015. Distance metric learning for RRT-based motion planning with constant-time inference. In: Proc. IEEE Int. Conf. Robotics and Automation (ICRA). pp. 637–643.

Pan, J., Chen, Z., Abbeel, P., 2014. Predicting initialization effectiveness for trajectory optimization. In: Proc. IEEE Int. Conf. Robotics and Automation (ICRA). pp. 5183–5190.

Plaku, E., Kavraki, L. E., Vardi, M. Y., June 2010. Motion planning with dynamics by a synergistic combination of layers of planning. IEEE Trans. Robotics 26 (3), 469–482.

Prattichizzo, D., Malvezzi, M., Gabiccini, M., Bicchi, A., Dec. 2013. On motion and force controllability of precision grasps with hands actuated by soft synergies. IEEE Trans. Robotics 29 (6), 1440–1456.

Quigley, M., Conley, K., P Gerkey, B., Faust, J., Foote, T., Leibs, J., Wheeler, R., Y Ng, A., May 2009. ROS: An open-source Robot Operating System. In: ICRA Workshop Open Source Software. pp. 1–6.

Quispe, A., Ben-Amor, H., Christensen, H., Sept. 2015. A Taxonomy of Benchmark Tasks for Bimanual Manipulators. In: Proc. Int. Symp. Robotics Research. pp. 1–10.

Ratliff, N., Zucker, M., Bagnell, J., Srinivasa, S., 2009. CHOMP: Gradient optimization techniques for efficient motion planning. In: Proc. of the IEEE Int. Conf. on Robotics and Automation. pp. 489–494.

Rodríguez, C., Montaño, A., Suárez, R., Aug. 2014. Planning manipulation movements of a dual-arm system considering obstacle removing. Robotics and Autonomous Systems 62 (12), 1816–1826.

Romero, J., Feix, T., Ek, C., Kjellström, H., Kragic, D., Dec. 2013. Extracting Postural Synergies for Robotic Grasping. IEEE Trans. Robotics 29 (6), 1342–1352.

Rosell, J., Pérez, A., Aliakbar, A., Muhayyuddin, Palomo, L., García, N., Sept. 2014. The Kautham Project: A teaching and research tool for robot motion planning. In: Proc. IEEE Int. Conf. Emerging Technologies and Factory Automation (ETFA). pp. 1–8.

Rosell, J., Suárez, R., Sept. 2014. cRRT*: Planning loosely-coupled motions for multiple mobile robots. In: Proc. IEEE Int. Conf. Emerging Technologies and Factory Automation (ETFA). pp. 1–6.

Rosell, J., Suárez, R., Pérez, A., July 2013. Path Planning for Grasping Operations Using an Adaptive PCA-based Sampling Method. Autonomous Robots 35 (1), 27–36.

Rosell, J., Suárez, R., Rosales, C., Pérez, A., July 2011. Autonomous motion planning of a hand-arm robotic system based on captured human-like hand postures. Autonomous Robots 31 (1), 87–102.

Rosenbaum, D., Loukopoulos, L., Vaughan, J., Meulenbroek, R., Engelbrecht, S., Jan. 1995. Planning reaches by evaluating stored postures. Psychological Review 102 (1), 28–67.

Safonova, A., Hodgins, J. K., Pollard, N. S., Aug. 2004. Synthesizing physically realistic human motion in low-dimensional, behavior-specific spaces. ACM Trans. Graphics 23 (3), 514–521.

Sakagami, Y., Watanabe, R., Aoyama, C., Matsunaga, S., Higaki, N., Fujimura, K., Oct. 2002. The intelligent ASIMO: System overview and integration. In: Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS). Vol. 3. pp. 2478–2483.

Santello, M., Flanders, M., Soechting, J., Feb. 2002. Patterns of Hand Motion during Grasping and the Influence of Sensory Guidance. J. Neuroscience 22 (4), 1426–1435.

Santello, M., Flanders, M., Soechting, J. F., Dec. 1998. Postural Hand Synergies for Tool Use. J. Neuroscience 18 (23), 10105–10115.

Schaal, S., July 2007. The New Robotics - Towards human-centered machines. Human Frontier Science Program Journal 1 (2), 115–126.

Schaal, S., Mohajerian, P., Ijspeert, A., Oct. 2007. Dynamics systems vs. optimal control - A unifying view. In: Computational Neuroscience: Theoretical Insights into Brain Function. Vol. 165 of Progress in Brain Research. Elsevier, pp. 425–445.

Schulman, J., Duan, Y., Ho, J., Lee, A., Awwal, I., Bradlow, H., Pan, J., Patil, S., Goldberg, K., Abbeel, P., Aug. 2014. Motion planning with sequential convex optimization and convex collision checking. Int. J. Robotics Research 33 (9), 1251–1270.

Shauri, R., Nonami, K., Aug. 2011. Assembly manipulation of small objects by dual-arm manipulator. Assembly Autom. 31 (3), 263–274.

Shin, D., Hamner, B., Singh, S., H., M., Oct. 2003. Motion planning for a mobile manipulator with imprecise locomotion. In: Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS). pp. 847–853.

Smith, C., Karayiannidis, Y., Nalpantidis, L., Gratal, X., Qi, P., Dimarogonas, D., Kragic, D., Oct. 2012. Dual arm manipulation - A survey. Robotics and Autononmous Systems 60 (10), 1340–1353.

Sreenivasa, M., Souères, P., Laumond, J., July 2012. Walking to Grasp: Modeling of Human Movements as Invariants and an Application to Humanoid Robotics. IEEE Trans. Systems, Man and Cybernetics A: Systems and Humans 42 (4), 880–893.

Srinivasa, S., Berenson, D., Cakmak, M., Collet, A., Dogar, M., Dragan, A., Knepper, R., Niemueller, T., Strabala, K., Vande Weghe, M., Ziegler, J., Aug. 2012. Herb 2.0: Lessons Learned From Developing a Mobile Manipulator for the Home. Proc. IEEE 100 (8), 2410–2428.

Stilman, M., June 2010. Global Manipulation Planning in Robot Joint Space With Task Constraints. IEEE Trans. Robotics 26 (3), 576–584.

Suárez, R., Palomo, L., Martinez, J., Clos, D., García, N., Aug. 2018. Development of a Dexterous Dual-Arm Omnidirectional Mobile Manipulator. IFAC-PapersOnLine 51 (22), 126 – 131, IFAC Symp. Robot Control (SYROCO).

Suárez, R., Rosell, J., García, N., May 2015. Using Synergies in Dual-Arm Manipulation Tasks. In: Proc. IEEE Int. Conf. Robotics and Automation (ICRA). pp. 5655–5661.

Suçan, I., Kavraki, L., 2012. A sampling-based tree planner for systems with complex dynamics. IEEE Trans. on Robotics 28 (1), 116–131.

Suçan, I., Moll, M., Kavraki, L., Dec. 2012. The Open Motion Planning Library. IEEE Robotics & Automation Magazine 19 (4), 72–82.

Sun, S., Rosales, C., Suárez, R., June 2010. Study of coordinated Motions of the Human Hand for Robotic Applications. In: Proc. IEEE Int. Conf. Information and Automation. pp. 776–781.

Tsoli, A., Jenkins, O., Nov. 2007. Robot grasping for prosthetic applications. In: Proc. Int. Symp. ISRR. pp. 1–12.

Uno, Y., Kawato, M., Suzuki, R., June 1989. Formation and control of optimal trajectory in human multijoint arm movement. Biological Cybernetics 61 (2), 89–101.

Vahrenkamp, N., Berenson, D., Asfour, T., Kuffner, J., Dillmann, R., Oct. 2009. Humanoid Motion Planning for Dual-Arm Manipulation and Re-Grasping Tasks. In: Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS). pp. 2464–2470.

Vinjamuri, R., Mao, Z.-H., Sclabassi, R., Sun, M., Aug. 2007. Time-Varying Synergies in Velocity Profiles of Finger Joints of the Hand during Reach and Grasp. In: Proc. IEEE Annual Int. Conf. Engineering in Medicine and Biology Society (EMBS). pp. 4846–4849.

Vinjamuri, R., Sun, M., Chang, C.-C., Lee, H.-N., Sclabassi, R., Mao, Z.-H., July 2010. Temporal Postural Synergies of the Hand in Rapid Grasping Tasks. IEEE Trans. Information Technology in Biomedicine 14 (4), 986–994.

Wachs, J., Stern, H., Edan, Y., 2005. Cluster Labeling and Parameter Estimation for the Automated Setup of a Hand-Gesture Recognition System. IEEE Trans. Systems, Man and Cybernetics A: Systems and Humans 35 (6), 932–944.

Wang, Y., Artemiadis, P., Dec. 2013. Closed-Form Inverse Kinematic Solution for Anthropomorphic Motion in Redundant Robot Arms. Advances in Robotics and Automation 2 (3), 1–7.

Wimböck, T., Jan, B., Hirzinger, G., Sept. 2011. Synergy-level impedance control for a multifingered hand. In: Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS). pp. 973–979.

Wojtara, T., Nonami, K., Sept. 2004. Hand Posture Detection by Neural Network and Grasp Mapping for a Master Slave Hand System. In: Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS). pp. 866–871.

Woodbury, M., 1950. Inverting Modified Matrices. No. 42 in Statistical Research Group Memorandum Reports. Princeton University, Princeton, NJ.

Yamamoto, Y., Yun, X., June 1994. Coordinating locomotion and manipulation of a mobile manipulator. IEEE Trans. Automatic Control 39 (6), 1326–1332.

Yang, Y., Brock, O., Apr. 2004. Adapting the sampling distribution in PRM Planners based on an Approximated Medial Axis. In: Proc. IEEE Int. Conf. Robotics and Automation (ICRA). pp. 4405–4411.

Yang, Y., Brock, O., Jan. 2010. Elastic Roadmaps-motion Generation for Autonomous Mobile Manipulation. Autonomous Robots 28 (1), 113–130.

Yershova, A., Jaille, L., Siméon, T., LaValle, S., Apr. 2005. Dynamic-Domain-RRTs: Efficient Exploration by Controlling the Sampling Domain. In: Proc. IEEE Int. Conf. Robotics and Automation (ICRA). pp. 3856–3861.

Zacharias, F., Borst, C., Beetz, M., Hirzinger, G., Sept. 2008. Positioning mobile manipulators to perform constrained linear trajectories. In: Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS). pp. 2578–2584.

Zhang, L., Manocha, D., May 2008. An Efficient Retraction-base RRT Planner. In: Proc. IEEE Int. Conf. Robotics and Automation (ICRA). pp. 3743–3750.

Zöllner, R., Asfour, T., Dillmann, R., Sept. 2004. Programming by demonstration: Dual-arm manipulation tasks for humanoid robots. In: Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS). pp. 479–484.

Zucker, M., Kuffner, J., Bagnell, J., 2008. Adaptive workspace biasing for sampling-based planners. In: Proc. IEEE Int. Conf. Robotics and Automation (ICRA). pp. 3757–3762.