



SAPIENZA
UNIVERSITÀ DI ROMA

DEPARTMENT OF COMPUTER, CONTROL, AND MANAGEMENT
“A. RUBERTI”

PHD PROGRAM IN ENGINEERING IN COMPUTER SCIENCE

**Contact aware robust semi-autonomous
teleoperation of mobile manipulators**
PhD thesis

Supervisor

Prof. Fiora Pirri
pirri@diag.uniroma1.it

PhD Candidate

Manuel A. Ruiz Garcia
ruiz@diag.uniroma1.it

A. A. 2017-2018

Contents

1	Introduction	iii
2	Mobile manipulators	1
2.1	Kinematics modeling	2
2.1.1	Mobile manipulator Jacobian	3
2.1.2	Redundancy resolution	4
2.1.3	Manipulability measure	14
2.1.4	Remote center of motion (RCM) constraint	18
2.2	Dynamics modeling and estimation	20
2.2.1	Euler-Lagrange formulation	20
2.2.2	Dynamic model of mobile manipulators	24
2.2.3	Identification of dynamic parameters	25
2.3	Incipient collision detection	28
2.4	Reactive control of mobile manipulators	30
2.4.1	Control of manipulators	31
2.4.2	Control of mobile robots	34
2.4.3	On-line planning with artificial potential fields	39
3	Semiautonomous teleoperation of redundant mobile manipulators	43
3.1	Problem statement	45
3.2	Proposed approach	47
3.2.1	RCM constraint in <i>orbit mode</i> control	47
3.2.2	Mapping operator intentions	49
3.2.3	Reactive redundancy resolution	50
3.2.4	Manipulability maximization	50
3.2.5	Proposed control scheme	51
3.3	Experimental evaluation	52
3.4	Conclusions and future work	55

4	Robust collision detection in articulated tracked vehicles	67
4.1	Problem statement	69
4.2	Statistical learning framework	71
4.2.1	Wavelet packed decomposition (WPD)	73
4.2.2	Classification	75
4.3	Experimental evaluation	78
4.4	Conclusions and future work	85
5	Coplanarity constrained monocular visual odometry	87
5.1	Introduction	87
5.2	Stereo geometry revisited	89
5.2.1	Preliminaries	89
5.2.2	Tangent space of the essential manifold \mathcal{E}	91
5.2.3	Essential manifold parameterization	92
5.2.4	Optimization on the essential manifold	93
5.3	Coplanarity constraint	95
5.3.1	Mixed algebraic matrix constraints	95
5.3.2	Algebraic insights	97
5.4	Constrained geometric optimization	99
5.4.1	Three-view soft-constrained optimization	99
5.4.2	Experimental results	103
5.4.3	Three-view hard-constrained optimization	106
5.4.4	Experimental results	108
5.5	Conclusions and future work	112
	References	114

Chapter 1

Introduction

*Teleoperators of today
are the autonomous
robots of tomorrow*

Murray, Li and Sastry

With the advent of autonomous robotics, collaborative robotics, minimally invasive robotics surgery, rehabilitation and health care robotics, etc., the classical paradigm of robotics *safety by segregation* has become obsolete. Even in applications involving unpredictable or hazardous environments for human operators, like space operations, underwater inspections, nuclear or toxic waste management, explosives disarming, military surveillance and search & rescue on disaster scenarios, robots must safeguard their own integrity. Nowadays many research efforts, including design strategies, control methods, planning techniques and perceptual awareness, are focused in the concept of *intrinsic safety*, that is, an inner property of the robotic system and not a set of rules imposed to its behaviour. In such context, two main issues need to be addressed [48]: the protection of the dynamic environment –specially in the presence of humans– and the protection of the robotic structure. The first and most critical issue, in terms of situational awareness, is prediction and prevention of risky circumstances. The second, in terms of self-sensing capabilities, is proprioceptive awareness and prompt compliant reactions after unexpected risky interactions.

The use of mobile manipulators is widespread on applications involving unpredictable or hazardous environments for human operators [59, 69], applications where the manipulator’s motion is controlled autonomously or remotely by specialized operators. For example, articulated tracked vehicles are widely used in contexts where terrain conditions are difficult and unpredictable. For better trac-

tion on harsh terrains these robotic platforms can either increase or decrease the tracks contact area with the ground by actuating their active sub-tracks, namely *flippers*, placed at the end of each track. Flippers allow climbing stairs and similar obstacles by ensuring a better contact with the ground. Also, redundant mobile manipulators are common in context where a high dexterity is required for safe handling and manipulation.

Depending on the application, the remote control of a robot can be identified as teleoperation or telemanipulation. The former identifier refers to the case when the remote operator’s intentions are defined at a task-level. The latter, to the case where operator’s intentions are defined at an object-level. Regardless the type, there are three distinct levels of control schemes. In the most basic level, the operator *directly* controls the robot motion. At the highest level, the operator *supervises* actions computed by the robot in total autonomy. In the middle, there’s a wide variety of *shared* architectures, where semi-autonomous actions are guided by mixing the “human intelligence” and the “machine intelligence”. Here we will focus on shared control architectures for task-oriented applications.

Teleoperation of manipulators is not a straightforward task, and in many practical cases represent a common source of failure [12]. Common issues during the remote control of manipulators are: increasing control complexity with respect to the mechanical degrees of freedom; inadequate or incomplete feedback to the user; and motion directives for task execution may be incompatible with constraints or obstacles imposed by the environment. In the latter case, part of the manipulator may get trapped or blocked by some obstacle in the environment, failure that cannot be easily detected, isolated nor counteracted remotely. Today, an increasing number of approaches is focusing on semi-autonomous remote control (see for example [52, 80, 102, 78, 38, 42]), this is still an open problem and the above mentioned approaches provide case studies and not yet feasible solutions.

It is worth noticing that, among the wide variety of risky or hazardous circumstances, the most dangerous risk specific to robots are the unexpected collisions or unwanted force exertions between the machine and the environment [89]. Avoiding unexpected force exertions implies foreseeing dangerous situations, and thus relies on sensing, situational awareness, planning and decision making capabilities. When an unexpected exertion occurs, impact forces may be eased through lightweight design and compliant mechanisms and control. Prompt reactions after incipient exertions requires real-time fault diagnosis, fast reflexes and intelligent compensatory behaviours to recover safe operative conditions. In the context of teleoperation where full knowledge of the environment is not always

available, autonomous mechanism for collision detection, reaction and isolation are fundamental to reduce the operator’s efforts required to restore safe operative conditions after unexpected –unhandled– events occurs. Here, reaction implies a prompt definition and execution of a strategy minimizing the effects of unexpected perturbations, while avoiding compromising the task execution. The ability of the robotic system to satisfy multiple goals or constraints simultaneously is defined in terms of the kinematic *redundancy* of the robot. Redundancy allows self-motions in the mechanical structure of the robot without disturbing the constraints of the primary task. The scope of this thesis is to explore a combination of a reactive control scheme and a proprioceptive mechanism for robust semi-autonomous teleoperation of mobile manipulators when unexpected collisions occurs. The goal is to autonomously handle the stabilization of the robot toward safe operative conditions after an unexpected collision with the environment has been detected.

Our aim is to develop a novel approach for teleoperation, taking advantage of the robot’s awareness of the surrounding environment to translate the operator intentions (or commands) into a safe and reliable motion during the task execution. In particular, we are interested in the automatic detection and autonomous reaction to collisions or contact events between the manipulator and the environment, while minimizing the operator efforts to conclude the task after some unexpected interaction of this kind occurs. Since the collisions between the environment and the manipulator can involve any arbitrary region of the robotic arm’s shape, we impose as a research objective, that our detection framework should not rely on any tactile sensory technology, since a sensor based approach implies that the whole manipulator’s surface should be uniformly covered by arrays of sensors, which may imply expensive efforts either in terms of cost, design, interfacing or signal processing.

As stated, our problem falls in the context of fault detection and isolation (FDI) [7]. There exist two main approaches for fault detection: signal based and model based. Signal based approaches searches for faulty patterns over sensor readings. Instead, model based approaches compute a *residual signal*, measuring the discrepancy between sensor readings and predicted (nominal) values. In the case of dynamical systems, pure signal based approaches lead to inaccurate results since the dynamics of the system induce deviations from the nominal fault-free operation that cannot be predicted without proper dynamics modeling [28]. Moreover, in the case of incipient collision detection, the use of specific external sensors (vision, strain gauges, load cells, etc.) increases the cost and complexity of the robotics system [22]. Besides, model based approaches are subject to an accu-

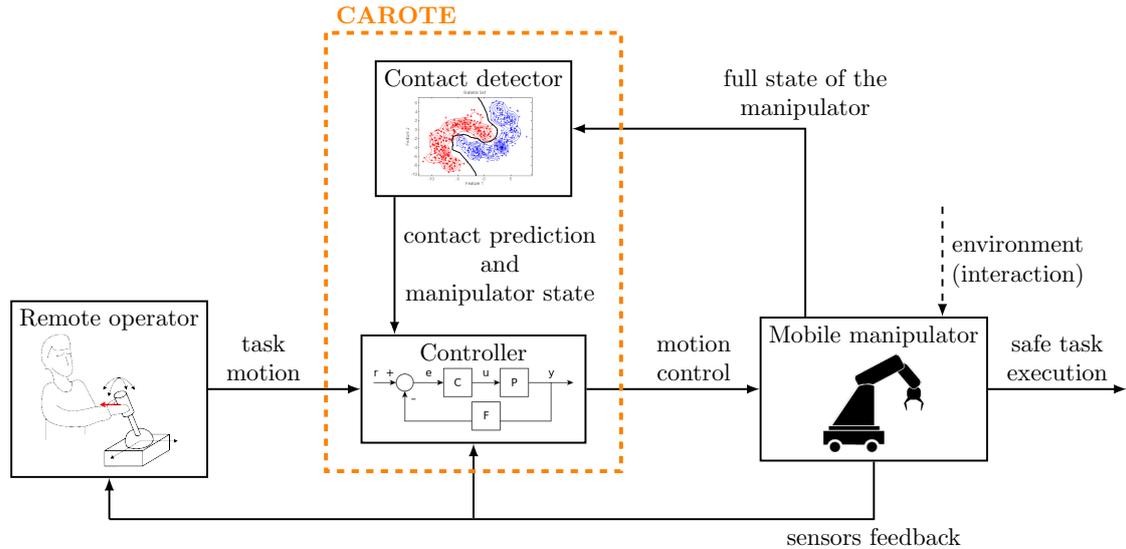


Figure 1.1: CAROTE framework.

rate identification of the dynamics coefficients of the model [39] and may require full-state feedback, including accelerations which, in practice, are inherently noisy due numerical estimation [28].

With this idea in mind, our methodology consists of the identification of contact events with the environment during teleoperation by exploitation of a model based proprioceptive measurement of the robot dynamics [22, 48], allowing the computation of suitable control laws exploiting the available mechanical redundancy [25, 72, 36, 37] to *locally* correct the robot’s pose without human intervention nor disturbing the task execution. More precisely, the proposed *Contact Aware Robust semi-autonomous Teleoperation* (CAROTE) framework consist in two different building blocks as shown on Figure 1.1. One, for detecting unexpected interactions with the environment (perceptive block). The other, for intelligent and autonomous reaction after the stimulus (control block).

The first or perceptive block, responsible of the contact event identification [43]. In short, this approach has proved the claim that a collision detection method for robot manipulators, can be extended to the field of articulated tracked vehicles, by embedding it within a statistical learning framework that takes into account the dynamic model uncertainties and locomotion disturbances. Formally, since the manipulator (a tree-shaped open kinematic chain) is rigidly attached to a mobile base, the core idea is to employ the sensorless FDI approach of [21, 22], where collisions are associated to unexpected transient perturbations of a residual signal, obtained through a non-linear observer of the manipulator generalized momentum. The extension of such FDI approach requires an accurate knowledge of the mobile

platform dynamics, and thus, an accurate model of the terrain. In FDI terms, our system presents an intermittent fault induced by the locomotion dynamics of the mobile base. To avoid the complex modeling of the robot locomotion, we propose the hypothesis that non modeled locomotion dynamics induce residual patterns that can be discriminated by those generated by the unexpected collisions, and so we can apply a classification method to recover any contact event. In other words, we model the residual dynamics as a linear combination of two disturbance sources, whose evolution patterns can be independently identified within a statistical learning framework. That is a mixed signal and model based approach for fault detection and isolation.

The second or control block, deals with the intelligent or autonomous reaction after a contact or impact event with the environment occurs. Therefore, we need a variable controller that produces different control laws based on the contact state. The controller implements a classical Jacobian null-space kinematic control scheme [36]. The primary task is defined by a direct Cartesian control law for intuitive teleoperation [56] based on the orbit object control mode [9], implemented through the *remote center of motion* (RCM) constraint [1, 18], widely used in the field of minimally invasive surgery (MIS). The secondary task exploits the kinematic redundancy between the manipulator and the mobile base, by inducing self-motions (i.e. projected in the null-space of the primary task), driven by an artificial potential field maximizing a variable cost function. When no unexpected interactions occur, this cost function is fixed, given by the arm manipulability [110, 101]. Instead, after the identification of one or more unexpected collisions, the potential field is defined by a linear combination between the manipulability maximizing potential field and a set of potential fields maximizing the clearness of the robot during motion.

The rest of the thesis is organized as follows. Chapter 2 provides a general overview of the modeling and control techniques of mobile manipulators underlying the CAROTE framework. In particular, we focus on the local redundancy resolution methods, dynamic model identification, sensorless incipient collision detection and kinematic control schemes. Chapter 3 described the proposed control mechanism for semi-autonomous teleoperation of redundant mobile manipulators and its empirical demonstration with a redundant holonomic mobile manipulator. The perceptive block of CAROTE is described in Chapter 4 together with the outcomes obtained in a simulated environment.

Chapter 5 of the manuscript contains two geometric optimization algorithms, developed by the authors, for on-manifold estimation of the pair-wise stereo ge-

ometry between two calibrated views, applied to the problem of monocular visual odometry, using as consistency measure the coplanarity constraint between three views. Although this work doesn't belong to the CAROTE framework, it represents an original research effort done by the authors during the last years of his PhD, efforts that are proudly presented together with the main research achievements concerning the CAROTE framework.

Chapter 2

Mobile manipulators

From a mechanical perspective, a robot manipulator is a kinematic chain of rigid bodies (links), connected by means of kinematic pairs (joints) of one, two or three degrees of freedom. One end of the chain is firmly attached to an (inertial) reference frame or base, while an end-effector is mounted on the other end.

Relative position and orientation (pose) between two consecutive links are conveniently described in terms of an elementary homogeneous transformation matrix, which encode the links geometry and the kinematic relation defined by the common joint variable. The end-effector pose is then defined by the composition of all elementary homogeneous transformation matrices from the base link to the end-effector. In the same way, relative motions are formally described in terms of the Jacobian matrices and the velocities of the joint variables and, as a consequence, the motion of the end-effector is obtained by composition of all elementary motions acting on it. End-effector motions and joint efforts¹ can be related through the dynamic model of the manipulator. Although the analysis of manipulator dynamics provides great analytical and practical benefits (e.g., motion simulation, control algorithms synthesis, structural properties analysis during design, etc.), the computation of the model requires a precise knowledge of the dynamic parameters which in practice implies an *identification* step.

A mobile manipulator is a robot manipulator whose base correspond to a mobile platform. Therefore, the kinematics of a mobile manipulator depend on poses and motions of both the chain and the base. Often, depending on the task, the degrees of freedom introduced by the mobile base make the mobile manipulator *redundant*. Redundancy can be exploited to achieve subsidiary goals during the task execution without compromising nor disturbing it. In general, modeling the dynamics of mobile manipulators is a complex task, since it depends on the base

¹Or *generalized forces*, than can be either forces or torques.

link locomotion, which in turns is defined by the contact interaction between the robot and the ground.

In general, the motion control of a mobile manipulator implies the determination of the time history of the joint efforts required to guarantee the execution of a given task, satisfying all temporal and spatial requirements (or *constraints*) provided. In the rest of the manuscript, by task we indicate a trajectory in position and orientation of the end-effector. The computation of such predefined motions correspond to the problem of motion planning. In the following we will refer to the *controller* as the unit implementing the control mechanism and the *planner* as the unit generating the input references for the controller.

The present chapter is devoted to introducing the aforementioned mechanical aspects of mobile manipulators and to provide a review of some motion control and on-line planning techniques. That is, all analytical and modeling tools underlying the CAROTE system. In particular, Section 2.1 introduces some tools for the analysis of the mobile manipulators kinematics. The dynamic model and experimental identification of the dynamic coefficients of robot manipulators are described in Section 2.2. Section 2.3 introduces the generalized momenta dynamic observer for sensorless collision detection and reaction. Finally, a brief review of motion control techniques for mobile manipulator is presented in Section 2.4.

In the next paragraphs all poses and motions of the end-effector (or any other generic reference frame in the workspace) will be defined in terms of the *world frame* and unless otherwise stated all computation will be derived with respect to it. In the particular case of robot manipulators such *world frame* will always coincides with the robot's *base frame*. Most of the contents of this chapter follow the textbooks [14, 93, 90].

2.1 Kinematics modeling

The present paragraph aims to briefly describe some specific topics of the differential kinematics of mobile manipulators. In particular, we will focus on the concept of redundancy exploitation and manipulability maximization, to provide a consistent theoretical framework for the design of robust velocity based control schemes for redundant mobile manipulators (see Chapter 3). It is assumed that the reader is familiar with the concepts of rigid body motion, forward/inverse kinematics of manipulators and kinematic models of classical wheeled mobile robots.

The rest of the section is organized as follows. Section 2.1.1 briefly introduces the mobile manipulator Jacobian. Local resolution methods for redundancy

exploitation are derived in Section 2.1.2. In Section 2.1.3 the concept of manipulability and a general manipulability maximization scheme is defined. Finally, in Section 2.1.4 the remote center constraint RCM is discussed.

2.1.1 Mobile manipulator Jacobian

Suppose that a mobile manipulator is modeled by a n -dimensional vector \mathbf{q} of generalized coordinates defined inside a configuration space $\mathcal{Q} \subset \mathbb{R}^n$, whose end-effector can reach any pose in the workspace $\mathcal{W} \subset \mathbb{R}^m$. The end-effector pose can be computed through the direct kinematics map $\mathbf{q}(t) \mapsto f(\mathbf{q}(t))$, $f: \mathcal{Q} \rightarrow \mathcal{W}$. Any configuration \mathbf{q} represents the concatenation of the joint variables \mathbf{q}_a of the arm together with the generalized coordinates \mathbf{q}_b of the mobile base

$$\mathbf{q} = \begin{pmatrix} \mathbf{q}_a \\ \mathbf{q}_b \end{pmatrix} \quad (2.1)$$

Suppose that a *feasible* time-varying task $\mathbf{r}(t) \in \mathcal{W}_r$, with $\mathcal{W}_r \subset \mathbb{R}^r$ the task space² and $r \leq m$, is assigned to the end-effector. Our goal is to determine the differential relation between the end-effector motion (i.e., workspace velocities) $\dot{\mathbf{r}}(t)$ and the velocities $\dot{\mathbf{q}}(t)$ of the generalized coordinates. Removing the explicit time dependency for better readability, such relation corresponds to

$$\dot{\mathbf{r}} = \frac{d}{dt} f(\mathbf{q}) = \frac{\partial f(\mathbf{q})}{\partial \mathbf{q}} \dot{\mathbf{q}} = \begin{pmatrix} \frac{\partial f(\mathbf{q})}{\partial \mathbf{q}_a} & \frac{\partial f(\mathbf{q})}{\partial \mathbf{q}_b} \end{pmatrix} \begin{pmatrix} \dot{\mathbf{q}}_a \\ \dot{\mathbf{q}}_b \end{pmatrix} = \begin{pmatrix} \mathbf{J}_a(\mathbf{q}) & \mathbf{J}_b(\mathbf{q}) \end{pmatrix} \begin{pmatrix} \dot{\mathbf{q}}_a \\ \dot{\mathbf{q}}_b \end{pmatrix}$$

Let's assume now that the real velocity commands $\dot{\hat{\mathbf{q}}}_b(t)$ of the mobile platform are related to the generalized velocities $\dot{\mathbf{q}}_b(t)$ through a differential map of the form

$$\dot{\mathbf{q}}_b(t) = \mathbf{G}(\mathbf{q}_b(t)) \dot{\hat{\mathbf{q}}}_b(t) \quad (2.2)$$

with $\mathbf{G}(\mathbf{q}_b(t))$ representing any locomotion constraint of the mobile base (either holonomic or nonholonomic). Then, dropping the explicit time dependency for better readability, we have that

$$\dot{\mathbf{r}} = \begin{pmatrix} \mathbf{J}_a(\mathbf{q}) & \mathbf{J}_b(\mathbf{q}) \end{pmatrix} \begin{pmatrix} \dot{\mathbf{q}}_a \\ \mathbf{G}(\mathbf{q}_b) \dot{\hat{\mathbf{q}}}_b \end{pmatrix} = \begin{pmatrix} \mathbf{J}_a(\mathbf{q}) & \mathbf{J}_b(\mathbf{q})\mathbf{G}(\mathbf{q}_b) \end{pmatrix} \begin{pmatrix} \dot{\mathbf{q}}_a \\ \dot{\hat{\mathbf{q}}}_b \end{pmatrix} = \mathbf{J}(\mathbf{q}) \dot{\hat{\mathbf{q}}} \quad (2.3)$$

²A fully *reachable* task space \mathcal{W}_r is entirely contained in the workspace of the robot, that is, $\mathcal{W}_r \subset \mathcal{W}$. When the task space is partially reachable, i.e., $\mathcal{W}_r \not\subset \mathcal{W}$, a *feasible* task $\mathbf{r}(t)$ is entirely contained in the non-empty intersection $\mathcal{W}_r \cap \mathcal{W}$ for all $t > \mathbb{R}^+$.

where the matrix $J(\mathbf{q})$ represents the Jacobian of the mobile manipulator. It is worth noticing that in the case of holonomic constraints, based on the implicit function theorem, it is possible to solve the constraints and define an unconstrained configuration space of lower dimension. Instead, in the case of nonholonomic constraints, we have that the columns of $G(\mathbf{q}_b(t))$ in (2.2) span the null space of the matrix $A^\top(\mathbf{q})$ defining the set of nonholonomic Pfaffian constraint of the mobile platform

$$A^\top(\mathbf{q}) \dot{\mathbf{q}}_b = 0 \quad (2.4)$$

2.1.2 Redundancy resolution

From a kinematic viewpoint, a robot is said to be redundant if, for a given feasible task $\mathbf{r}(t) \in \mathcal{W}_r$, the dimension n of the configuration space \mathcal{Q} is greater than the dimension p of the task space \mathcal{W}_r . It is worthy of remark that redundancy is a relative concept, depending entirely on the task definition. For example, a 6-DoF anthropomorphic manipulator is redundant for a task constraining the position of the end-effector, but not for a task constraining its entire pose. Redundancy implies that for any reachable task configuration $\mathbf{r} \in \mathcal{W}_r$, there exists at least one continuous set of configurations $\mathcal{Q}_r \subset \mathcal{Q}^3$ such that

$$\mathbf{r} = f(\mathbf{q}), \quad \forall \mathbf{q} \in \mathcal{Q}_r$$

which means that any motion $\mathbf{q}(t) \in \mathcal{Q}_r$, will reproduce the same task configuration \mathbf{r} for all $t \in \mathbb{R}^+$. In other words, there exists a set of *internal* displacements or *self*-motions modifying the robot posture without disturbing the constraints imposed by the task. Such *unobservable* displacements at the task level can be algebraically described in terms of the range and null spaces of the Jacobian matrix.

To this end, let's us denote with $J(\mathbf{q}(t))$ the Jacobian matrix of our mobile manipulator with $\mathbf{q}(t) \in \mathcal{Q}$ and $t \in \mathbb{R}^+$. Workspace velocities are linearly related to the velocities of the configuration space through the relation

$$\dot{\mathbf{r}}(t) = J(\mathbf{q}(t)) \dot{\mathbf{q}}(t) \quad (2.5)$$

Since $p \leq n$, then $J(\mathbf{q})$ has a non-empty⁴ null space $\mathcal{N}(J(\mathbf{q}))$. Dropping the explicit

³Without redundancy –at most– a countable set of isolated configurations may satisfy the constraints imposed by the task. For example, when for a given task configuration the inverse kinematics has multiple isolated solutions.

⁴ $J(\mathbf{q})$ is a $(p \times n)$ matrix with $n > p$, then the dimension of its range space $\mathcal{R}(J(\mathbf{q}))$ is at most p . By the rank-nullity theorem the dimension of the null space $\mathcal{N}(J(\mathbf{q}))$ is at least $n - p$.

time dependency for better readability and denoting with $\mathbf{N}(\mathbf{q})$ a $(n \times n)$ matrix whose columns spans the null space $\mathcal{N}(\mathbf{J}(\mathbf{q}))$, it follows that for any velocity $\dot{\mathbf{q}}_0$

$$\dot{\mathbf{r}} = \mathbf{J}(\mathbf{q})(\dot{\mathbf{q}} + \mathbf{N}(\mathbf{q})\dot{\mathbf{q}}_0) = \mathbf{J}(\mathbf{q})\dot{\mathbf{q}} + \mathbf{J}(\mathbf{q})\mathbf{N}(\mathbf{q})\dot{\mathbf{q}}_0 = \mathbf{J}(\mathbf{q})\dot{\mathbf{q}}$$

since $\mathbf{J}(\mathbf{q})\mathbf{N}(\mathbf{q})\dot{\mathbf{q}}_0$ is always zero regardless the value of $\dot{\mathbf{q}}_0$. Clearly, for a given task $\mathbf{r}(t) \in \mathcal{W}_{\mathbf{r}}$ we can identify infinitely many solutions $\mathbf{q}(t) \in \mathcal{Q}$ satisfying the constraint (2.5). Then, a particular solution can be chosen so as to improve some subsidiary goal(s) like, for example, obstacle clearance maximization, manipulability maximization in predefined directions, energy minimization, etc. That is, we can define a constrained optimization problem for the local or global redundancy resolution.

On the one hand, the global redundancy resolution problem can be stated as: for a given task $\mathbf{r}(t)$, a cost functional $H(\mathbf{q}(t), \dot{\mathbf{q}}(t))$, a task execution time t_f and the initial robot configuration $\mathbf{q}(t_0)$ such that $\mathbf{r}(t_0) = f(\mathbf{q}(t_0))$, determine the optimal velocity $\dot{\mathbf{q}}(t)$ for $t \in [t_0, t_f]$. The optimization problem correspond a non-linear two-point boundary value (TPBV) problem, whose computational complexity may require off-line computation of the optimal solution. Therefore, depending on the application, the optimal solution can represent either a standalone open-loop control, or a feed-forward law if disturbances were taken into account during optimization, or the input reference for a feedback control loop.

On the other hand, the local redundancy resolution problem can be stated as: for a given task $\mathbf{r}(t)$, a cost function $H(\mathbf{q}(t), \dot{\mathbf{q}}(t))$, the initial robot configuration $\mathbf{q}(t_0)$ such that $\mathbf{r}(t_0) = f(\mathbf{q}(t_0))$ and the sampling time of the discretization $T > 0$, determine the optimal velocity $\dot{\mathbf{q}}(t)$ at $t = kT$ for each $k \in \mathbb{N}^+$. Each computed optimal value $\dot{\mathbf{q}}^*(kT)$ represents the current control command of the robot and, as soon as the new state $\mathbf{q}((k+1)T)$ is determined either by means of sensors feedback or numerical integration, the whole process is repeated at each time instant until task completion. This optimization problem correspond to a particular case of quadratic-programing (QP) that is solved on-line as part of the control loop during the task execution. The rest of the paragraph focuses on classical local redundancy resolution methods. Since the redundancy resolution methods described in the following sections are applied at each discrete time instant $t = kT$, we will remove further explicit time dependencies for better readability. Also, for simplicity of notation, in the following we will drop the explicit dependency of the Jacobian matrix $\mathbf{J}(\mathbf{q})$ on the robot configuration \mathbf{q} .

Least norm solution Let's assume that we're seeking for the optimal velocity $\dot{\mathbf{q}}^*$, satisfying the task constraint (2.5) and minimizing the following cost function

$$H(\mathbf{q}, \dot{\mathbf{q}}) = H(\dot{\mathbf{q}}) = \frac{1}{2} \|\dot{\mathbf{q}}\|^2 = \frac{1}{2} \dot{\mathbf{q}}^\top \dot{\mathbf{q}} \quad (2.6)$$

that is, $\dot{\mathbf{q}}^*$ represents the only velocity vector satisfying the task constraint having a minimal norm (i.e., the control effort). The present QP problem can be solved using the Lagrange multipliers method. To this end, let's define the Lagrangian

$$\mathcal{L}(\dot{\mathbf{q}}, \lambda) = H(\dot{\mathbf{q}}) + \lambda^\top (\dot{\mathbf{r}} - \mathbf{J}\dot{\mathbf{q}}) \quad (2.7)$$

For $\mathcal{L}(\dot{\mathbf{q}}, \lambda)$ to have a local minimum at $(\dot{\mathbf{q}}^*, \lambda^*)$ the following necessary conditions must be verified

$$\left. \frac{\partial \mathcal{L}(\dot{\mathbf{q}}, \lambda)}{\partial \dot{\mathbf{q}}} \right|_{(\dot{\mathbf{q}}^*, \lambda^*)} = \dot{\mathbf{q}}^* - \mathbf{J}^\top \lambda^* = 0 \implies \dot{\mathbf{q}}^* = \mathbf{J}^\top \lambda^* \quad (2.8)$$

$$\left. \frac{\partial \mathcal{L}(\dot{\mathbf{q}}, \lambda)}{\partial \lambda} \right|_{(\dot{\mathbf{q}}^*, \lambda^*)} = \dot{\mathbf{r}} - \mathbf{J}\dot{\mathbf{q}}^* = 0 \implies \dot{\mathbf{r}} = \mathbf{J}\dot{\mathbf{q}}^* \quad (2.9)$$

$$\left. \frac{\partial^2 \mathcal{L}(\dot{\mathbf{q}}, \lambda)}{\partial \dot{\mathbf{q}}^2} \right|_{(\dot{\mathbf{q}}^*, \lambda^*)} = \mathbf{I} > 0 \quad (2.10)$$

substituting the equation (2.8) on the equation (2.9), it follows that

$$\dot{\mathbf{r}} = \mathbf{J}\mathbf{J}^\top \lambda^* \implies \lambda^* = (\mathbf{J}\mathbf{J}^\top)^{-1} \dot{\mathbf{r}} \quad (2.11)$$

the matrix inversions is possible only under the assumption that \mathbf{J} has full row rank. Now substituting (2.11) inside (2.8) we obtain the optimal solution

$$\dot{\mathbf{q}}^* = \mathbf{J}^\top (\mathbf{J}\mathbf{J}^\top)^{-1} \dot{\mathbf{r}} = \mathbf{J}^\dagger \dot{\mathbf{r}} \quad (2.12)$$

The $(n \times p)$ matrix \mathbf{J}^\dagger in (2.12) is known as the Moore-Penrose inverse of \mathbf{J} . This matrix always exists and satisfy the *pseudo-inverse* properties

$$\mathbf{J}\mathbf{J}^\dagger \mathbf{J} = \mathbf{J} \quad \mathbf{J}^\dagger \mathbf{J}\mathbf{J}^\dagger = \mathbf{J}^\dagger \quad (\mathbf{J}\mathbf{J}^\dagger)^\top = \mathbf{J}\mathbf{J}^\dagger \quad (\mathbf{J}^\dagger \mathbf{J})^\top = \mathbf{J}^\dagger \mathbf{J} \quad (2.13)$$

also \mathbf{J}^\dagger represents a *right* inverse of \mathbf{J} since $\mathbf{J}\mathbf{J}^\dagger = \mathbf{I}$. Moreover, following simple algebraic computations it is possible to show that if the Jacobian matrix admits a singular value decomposition of the form $\mathbf{J} = \mathbf{U}\mathbf{S}\mathbf{V}^\top$, then the Moore-Penrose inverse

if given by

$$\mathbf{J} = \mathbf{U}\mathbf{S}\mathbf{V}^\top = \mathbf{U} \begin{pmatrix} \text{diag}\{\sigma_i\} & 0 \\ 0 & 0 \end{pmatrix} \mathbf{V}^\top \implies \mathbf{J}^\dagger = \mathbf{V}\mathbf{S}^\dagger\mathbf{U}^\top = \mathbf{V} \begin{pmatrix} \text{diag}\{\frac{1}{\sigma_i}\} & 0 \\ 0 & 0 \end{pmatrix} \mathbf{U}^\top \quad (2.14)$$

note that expression (2.14) holds even when \mathbf{J} is not full row rank⁵.

Weighted least norm solution In complete analogy to the previous optimization problem, let's assume that the cost function to minimize is given by

$$H(\mathbf{q}, \dot{\mathbf{q}}) = H(\dot{\mathbf{q}}) = \frac{1}{2} \|\dot{\mathbf{q}}\|_{\mathbf{Q}}^2 = \frac{1}{2} \dot{\mathbf{q}}^\top \mathbf{Q} \dot{\mathbf{q}}$$

in this case $\dot{\mathbf{q}}^*$ represents the only velocity vector satisfying the task constraint having a minimal norm, in terms of the metric induced by the weighting matrix \mathbf{Q} . \mathbf{Q} is assumed to be symmetric and positive definite, and can thus be decomposed as $\mathbf{Q} = \mathbf{W}^\top \mathbf{W}$ with \mathbf{W} an upper triangular matrix with positive diagonal entries. To compute the optimal solution it suffices to set $\dot{\mathbf{q}} = \mathbf{W} \dot{\mathbf{q}}$ and $\tilde{\mathbf{J}} = \mathbf{J}\mathbf{W}^{-1}$, then the least norm solution is given by

$$\dot{\mathbf{q}}^* = \tilde{\mathbf{J}}^\dagger \dot{\mathbf{r}} \implies \dot{\mathbf{q}}^* = \mathbf{W}^{-1} \tilde{\mathbf{J}}^\dagger \dot{\mathbf{r}} = \mathbf{W}^{-1} (\mathbf{J}\mathbf{W}^{-1})^\dagger \dot{\mathbf{r}} = \mathbf{J}_\mathbf{Q}^\dagger \dot{\mathbf{r}} \quad (2.15)$$

If \mathbf{J} is full row rank then (2.15) assumes the particular form

$$\mathbf{J}_\mathbf{Q}^\dagger = \mathbf{Q}^{-1} \mathbf{J}^\top (\mathbf{J} \mathbf{Q}^{-1} \mathbf{J}^\top)^{-1} \quad (2.16)$$

An example application of the weighted least norm solution is given by the classical approach of Chan and Dubey [13] for avoiding joint limits through redundancy resolution.

Generalized least norm solution As before, let's assume that we're seeking for the optimal velocity $\dot{\mathbf{q}}^*$, satisfying the task constraint (2.5) and minimizing the following cost function

$$H(\mathbf{q}, \dot{\mathbf{q}}) = H(\dot{\mathbf{q}}) = \frac{1}{2} \|\dot{\mathbf{q}} - \dot{\mathbf{q}}_0\|^2 = \frac{1}{2} (\dot{\mathbf{q}} - \dot{\mathbf{q}}_0)^\top (\dot{\mathbf{q}} - \dot{\mathbf{q}}_0)$$

⁵Actually, for an inverse problem of the form $\mathbf{y} = \mathbf{A}\mathbf{x}$, the Moore-Penrose inverse \mathbf{A}^\dagger provides the least norm least-squares solution for underdetermined systems (\mathbf{A} has more columns than rows) and the least squares solution for overdetermined ones (\mathbf{A} has more rows than columns), regardless the rank of \mathbf{A} . When \mathbf{A} is *onto*, i.e., has full row rank, the Moore-Penrose inverse matrix assumes the particular form $\mathbf{A}^\dagger = \mathbf{A}^\top (\mathbf{A}\mathbf{A}^\top)^{-1}$. Instead, when \mathbf{A} is *one-to-one*, i.e., has full column rank, such matrix has the special form $\mathbf{A}^\dagger = (\mathbf{A}^\top \mathbf{A})^{-1} \mathbf{A}^\top$.

In words, the objective of the QP problem is to satisfy the task constraint and to keep minimally bounded the norm of the optimal velocity $\dot{\mathbf{q}}^*$ while staying as close as possible to the reference velocity $\dot{\mathbf{q}}_0$. Since the associated Lagrangian has the same form of (2.7), equations (2.8) and (2.9) holds in this case, while the first necessary condition for a minimum at $(\dot{\mathbf{q}}^*, \lambda^*)$ becomes

$$\left. \frac{\partial \mathcal{L}(\dot{\mathbf{q}}, \lambda)}{\partial \dot{\mathbf{q}}} \right|_{(\dot{\mathbf{q}}^*, \lambda^*)} = \dot{\mathbf{q}}^* - \dot{\mathbf{q}}_0 - \mathbf{J}^\top \lambda^* = 0 \implies \dot{\mathbf{q}}^* = \mathbf{J}^\top \lambda^* + \dot{\mathbf{q}}_0 \quad (2.17)$$

substituting the equation (2.17) on the equation (2.9), it follows that

$$\dot{\mathbf{r}} = \mathbf{J}\mathbf{J}^\top \lambda^* + \mathbf{J}\dot{\mathbf{q}}_0 \implies \lambda^* = (\mathbf{J}\mathbf{J}^\top)^{-1}(\dot{\mathbf{r}} - \mathbf{J}\dot{\mathbf{q}}_0) \quad (2.18)$$

the matrix inversions is possible only under the assumption that \mathbf{J} has full row rank. Now substituting (2.18) inside (2.17) we obtain the optimal solution

$$\dot{\mathbf{q}}^* = \mathbf{J}^\dagger(\dot{\mathbf{r}} - \mathbf{J}\dot{\mathbf{q}}_0) + \dot{\mathbf{q}}_0 = \mathbf{J}^\dagger \dot{\mathbf{r}} + (\mathbf{I} - \mathbf{J}^\dagger \mathbf{J})\dot{\mathbf{q}}_0 \quad (2.19)$$

As can be seen, the optimal solution (2.19) is composed by two additive terms. The first, is the least norm solution previously computed. The second, trying to satisfy the additional constraint imposed by the reference $\dot{\mathbf{q}}_0$. By recalling the pseudo-inverse properties (2.13) it is possible to note that the matrix $\mathbf{J}^\dagger \mathbf{J}$ represents an orthogonal projector in the range space $\mathcal{R}(\mathbf{J}^\top)$ of the Jacobian transpose \mathbf{J}^\top . Therefore, the matrix $\mathbf{I} - \mathbf{J}^\dagger \mathbf{J}$ corresponds to an orthogonal projector on the null space $\mathcal{N}(\mathbf{J})$ of the Jacobian. This implies that any reference velocity $\dot{\mathbf{q}}_0$ in (2.19) can be used to induce self-motions of the robot without disturbing the task constraint, indeed, any velocity vector $\dot{\mathbf{q}} = (\mathbf{I} - \mathbf{J}^\dagger \mathbf{J})\dot{\mathbf{q}}_0$ satisfy the homogeneous matrix equation $\mathbf{J}\dot{\mathbf{q}}=0$.

Generalized weighted least norm solution In complete analogy to the previous optimization problem, let's assume that the cost function to minimize is given by

$$H(\mathbf{q}, \dot{\mathbf{q}}) = H(\dot{\mathbf{q}}) = \frac{1}{2} \|\dot{\mathbf{q}} - \dot{\mathbf{q}}_0\|_{\mathbf{Q}}^2 = \frac{1}{2} (\dot{\mathbf{q}} - \dot{\mathbf{q}}_0)^\top \mathbf{Q} (\dot{\mathbf{q}} - \dot{\mathbf{q}}_0)$$

in this case $\dot{\mathbf{q}}^*$ represents the only velocity vector satisfying the task constraint and, in terms of the metric induced by the weighting matrix \mathbf{Q} , having a minimal norm and staying as close as possible to the reference velocity $\dot{\mathbf{q}}_0$. The weighting matrix \mathbf{Q} is assumed to be symmetric and positive definite, thus it can be decomposed

as $\mathbf{Q}=\mathbf{W}^\top\mathbf{W}$ with \mathbf{W} an upper triangular matrix with positive diagonal entries. To compute the optimal solution it suffices to set $\dot{\tilde{\mathbf{q}}}=\mathbf{W}\dot{\mathbf{q}}$, $\tilde{\mathbf{J}}=\mathbf{J}\mathbf{W}^{-1}$ and $\dot{\tilde{\mathbf{q}}}_0=\mathbf{W}\dot{\mathbf{q}}_0$, then the least norm solutions is given by

$$\dot{\tilde{\mathbf{q}}}^* = \tilde{\mathbf{J}}^\dagger \dot{\tilde{\mathbf{r}}} + (\mathbf{I} - \tilde{\mathbf{J}}^\dagger \tilde{\mathbf{J}}) \dot{\tilde{\mathbf{q}}}_0$$

which implies

$$\dot{\mathbf{q}}^* = \mathbf{W}^{-1}(\mathbf{J}\mathbf{W}^{-1})^\dagger \dot{\tilde{\mathbf{r}}} + \left(\mathbf{W}^{-1} - \mathbf{W}^{-1}(\mathbf{J}\mathbf{W}^{-1})^\dagger \mathbf{J}\mathbf{W}^{-1}\right) \mathbf{W}\dot{\mathbf{q}}_0 = \mathbf{J}_q^\dagger \dot{\tilde{\mathbf{r}}} + (\mathbf{I} - \mathbf{J}_q^\dagger \mathbf{J}) \dot{\mathbf{q}}_0 \quad (2.20)$$

Again, if \mathbf{J} is full row rank, then \mathbf{J}_q^\dagger assumes the form (2.16).

Self-motion generation Since we're dealing with local or on-line redundancy resolution methods, it seems natural to choose the reference velocity $\dot{\mathbf{q}}_0$ proportional to the gradient of a scalar differentiable objective function $H_0(\mathbf{q})$ of the robot configuration \mathbf{q} , to realize one step of a constrained optimization algorithm at each iteration of the control loop

$$\dot{\mathbf{q}}_0 = \pm\eta \left(\frac{\partial H_0(\mathbf{q})}{\partial \mathbf{q}} \right)^\top$$

where $\eta>0$ and the sign depends on the optimization task (either maximization or minimization). Typical objective functions are the manipulability measure (discussed in Section 2.1.3), joint distance to mechanical limits, obstacle clearance, etc. A more efficient approach for the gradient and compound computation velocity is presented in [23]; its natural extension to the case of nonholonomic mobile manipulators is described in [25]. Flacco et al. [36] developed an efficient iterative algorithm for a generalized weighted least norm redundancy resolution method in the presence of hard bounds on the joint space motion.

Tikhonov regularization or damped least squares (DLS) When the robot configuration is close to a kinematics singularity, at least one (non-zero) singular value σ_i of the Jacobian matrix \mathbf{J} is close to become zero. As can be observed in equation (2.14), this may lead to algorithmic singularities in the computation of \mathbf{J}^\dagger , leading to unbounded velocities in the configuration space. It is possible to avoid such numerical instability by redefining the least norm constrained optimization problem as a regularized unconstrained one.

Let's define the following cost function, that takes into account both the task

constraint (2.5) and the velocity norm minimization cost (2.6)

$$H(\dot{\mathbf{q}}) = \frac{\mu^2}{2} \|\dot{\mathbf{q}}\|^2 + \frac{1}{2} \|\dot{\mathbf{r}} - \mathbf{J}\dot{\mathbf{q}}\|^2 = \frac{\mu^2}{2} \dot{\mathbf{q}}^\top \dot{\mathbf{q}} + \frac{1}{2} (\dot{\mathbf{r}} - \mathbf{J}\dot{\mathbf{q}})^\top (\dot{\mathbf{r}} - \mathbf{J}\dot{\mathbf{q}})$$

where μ is a small positive regularizing term. The optimal velocity $\dot{\mathbf{q}}^*$ minimizing the above cost has minimal norm and provides the least task error. The regularization parameter μ allows tuning the compromise between the constraints. For $H(\dot{\mathbf{q}})$ to have a local minimum at $\dot{\mathbf{q}}^*$ it must be verified that

$$\left. \frac{d}{d\dot{\mathbf{q}}} H(\dot{\mathbf{q}}) \right|_{\dot{\mathbf{q}}^*} = (\mathbf{J}^\top \mathbf{J} + \mu^2 \mathbf{I}) \dot{\mathbf{q}}^* - \mathbf{J}^\top \dot{\mathbf{r}} = 0 \quad \implies \quad \dot{\mathbf{q}}^* = (\mathbf{J}^\top \mathbf{J} + \mu^2 \mathbf{I})^{-1} \mathbf{J}^\top \dot{\mathbf{r}}$$

the optimal velocity $\dot{\mathbf{q}}^*$ represents a global minimum since the second derivative of $H(\dot{\mathbf{q}})$ with respect $\dot{\mathbf{q}}$ is always positive: $\mathbf{J}^\top \mathbf{J} + \mu^2 \mathbf{I} > 0$. It is fundamental to note that $\forall \mu \neq 0$ the following algebraic identity holds

$$\mathbf{J}^\top (\mathbf{J} \mathbf{J}^\top + \mu^2 \mathbf{I}_1)^{-1} = (\mathbf{J}^\top \mathbf{J} + \mu^2 \mathbf{I}_2)^{-1} \mathbf{J}^\top$$

with \mathbf{I}_1 and \mathbf{I}_2 identity matrices of proper dimensions; therefore depending on if the inverse problem is underdetermined or overdetermined, then one can choose one expression or the other, respectively, for faster computation. In fact, in our case the most convenient choice for the regularized solution is

$$\dot{\mathbf{q}}^* = \mathbf{J}^\top (\mathbf{J} \mathbf{J}^\top + \mu^2 \mathbf{I})^{-1} \dot{\mathbf{r}} = \mathbf{J}_{DLS}^\dagger \dot{\mathbf{r}} \quad (2.21)$$

Moreover, when \mathbf{J} is onto (i.e., full row rank), the regularized solution \mathbf{J}_{DLS}^\dagger converges to the least norm solution \mathbf{J}^\dagger as $\mu \rightarrow 0$.

In this basic version, the DLS solution always produces a task error because \mathbf{J}_{DLS}^\dagger is not a real pseudo-inverse of \mathbf{J} . This issue can be mitigated by introducing damping exclusively along non-feasible task directions (near singularities), using a distance measure based on the value of the smallest non-zero singular value of \mathbf{J} , technique known as numerical filtering [63].

Weighted regularized solution In complete analogy to the previous optimization problem, let's assume now that the cost function is given by

$$H(\dot{\mathbf{q}}) = \frac{\mu^2}{2} \|\dot{\mathbf{q}}\|_{\mathbf{Q}}^2 + \frac{1}{2} \|\dot{\mathbf{r}} - \mathbf{J}\dot{\mathbf{q}}\|_{\mathbf{P}}^2 = \frac{\mu^2}{2} \dot{\mathbf{q}}^\top \mathbf{Q} \dot{\mathbf{q}} + \frac{1}{2} (\dot{\mathbf{r}} - \mathbf{J}\dot{\mathbf{q}})^\top \mathbf{P} (\dot{\mathbf{r}} - \mathbf{J}\dot{\mathbf{q}})$$

where μ is a small positive regularizing term, \mathbf{Q} and \mathbf{P} are two symmetric positive definite matrices that can be decomposed as $\mathbf{Q}=\mathbf{W}^\top\mathbf{W}$ and $\mathbf{P}=\mathbf{R}^\top\mathbf{R}$, with \mathbf{W} and \mathbf{R} upper triangular matrices with positive diagonal entries. The optimal velocity $\dot{\mathbf{q}}^*$ minimizing the above cost has minimal norm with respect the metric induced by \mathbf{Q} and provides the least task error in term of the metric induced by \mathbf{P} . The regularization parameter μ allows tuning the compromise between the constraints. To compute the optimal solution one only has to set $\dot{\tilde{\mathbf{q}}}=\mathbf{W}\dot{\mathbf{q}}$, $\tilde{\mathbf{J}}=\mathbf{R}\mathbf{J}\mathbf{W}^{-1}$ and $\dot{\tilde{\mathbf{r}}}=\mathbf{R}\dot{\mathbf{r}}$, then the DLS solutions is given by $\dot{\tilde{\mathbf{q}}}^*=\tilde{\mathbf{J}}_{DLS}^\dagger\dot{\tilde{\mathbf{r}}}$. In the original coordinates, we have

$$\begin{aligned}
\dot{\mathbf{q}}^* &= \mathbf{W}^{-1}(\mathbf{R}\mathbf{J}\mathbf{W}^{-1})^\top \left((\mathbf{R}\mathbf{J}\mathbf{W}^{-1})(\mathbf{R}\mathbf{J}\mathbf{W}^{-1})^\top + \mu^2\mathbf{I} \right)^{-1} \mathbf{R}\dot{\mathbf{r}} \\
&= \mathbf{W}^{-1}\mathbf{W}^{-\top}\mathbf{J}^\top\mathbf{R}^\top \left(\mathbf{R}\mathbf{J}\mathbf{W}^{-1}\mathbf{W}^{-\top}\mathbf{J}^\top\mathbf{R}^\top + \mu^2\mathbf{I} \right)^{-1} \mathbf{R}\dot{\mathbf{r}} \\
&= \mathbf{Q}^{-1}\mathbf{J}^\top\mathbf{R}^\top \left(\mathbf{R}\mathbf{J}\mathbf{Q}^{-1}\mathbf{J}^\top\mathbf{R}^\top + \mu^2\mathbf{I} \right)^{-1} \mathbf{R}\dot{\mathbf{r}} \\
&= \mathbf{Q}^{-1}\mathbf{J}^\top\mathbf{R}^\top \left(\mathbf{R}[\mathbf{J}\mathbf{Q}^{-1}\mathbf{J}^\top + \mu^2\mathbf{R}^{-1}\mathbf{R}^{-\top}]\mathbf{R}^\top \right)^{-1} \mathbf{R}\dot{\mathbf{r}} \\
&= \mathbf{Q}^{-1}\mathbf{J}^\top \left(\mathbf{J}\mathbf{Q}^{-1}\mathbf{J}^\top + \mu^2\mathbf{P}^{-1} \right)^{-1} \dot{\mathbf{r}} \\
&= \mathbf{J}_{W DLS}^\dagger \dot{\mathbf{r}}
\end{aligned} \tag{2.22}$$

All remarks previously stated for the DLS solution hold also in this case.

Generalized least squares solutions Generalized least norm solutions (2.19) and (2.20) are also vulnerable to algorithmic singularities when the robot configuration approaches a kinematic singularity. In such a case, bounded optimal velocities can be obtained by means of a regularized inversion \mathbf{J}_μ^\dagger of the Jacobian, computed either by means of DLS or WDLS

$$\dot{\mathbf{q}}^* = \mathbf{J}_\mu^\dagger \dot{\mathbf{r}} + (\mathbf{I} - \mathbf{J}_\mu^\dagger \mathbf{J}) \dot{\mathbf{q}}_0 \tag{2.23}$$

However, such bounded optimal solution introduces a task error near singularities due to both, the damping effect of the inversion at the task level, and the wrong projection of the subsidiary velocity $\dot{\mathbf{q}}_0$ to the null space of the Jacobian matrix. In fact, the matrix $(\mathbf{I} - \mathbf{J}_\mu^\dagger \mathbf{J})$ is not a real orthogonal projector. This can be easily shown in the case of DLS inversion. Let (2.14) be the singular value decomposition of the Jacobian matrix \mathbf{J} , then

$$\begin{aligned}
\mathbf{I} - \mathbf{J}_\mu^\dagger \mathbf{J} &= \mathbf{I} - \mathbf{J}^\top (\mathbf{J}\mathbf{J}^\top + \mu^2\mathbf{I})^{-1} \mathbf{J} \\
&= \mathbf{I} - \mathbf{V}\mathbf{S}^\top\mathbf{U}^\top \left(\mathbf{U}\mathbf{S}\mathbf{V}^\top\mathbf{V}\mathbf{S}^\top\mathbf{U}^\top + \mu^2\mathbf{I} \right)^{-1} \mathbf{U}\mathbf{S}\mathbf{V}^\top
\end{aligned}$$

$$\begin{aligned}
&= \mathbf{V} \left(\mathbf{I} - \mathbf{S}^\top (\mathbf{S}\mathbf{S}^\top + \mu^2 \mathbf{I})^{-1} \mathbf{S} \right) \mathbf{V}^\top \\
&= \mathbf{V} \begin{pmatrix} \text{diag} \left\{ \frac{\mu^2}{\sigma_i^2 + \mu^2} \right\} & 0 \\ 0 & (1 - \mu^2) \mathbf{I} \end{pmatrix} \mathbf{V}^\top
\end{aligned} \tag{2.24}$$

which turns out to be full rank for any positive value of μ , regardless the dimension of the null space of \mathbf{J} . This effect can be lessened by introducing damping only the critical –close to zero– singular values σ_i of \mathbf{J} , through dynamic weighting [84] or by considering a robust singularity handling [72].

Task augmentation and prioritizing Many robotics application requires not only the regulation of the end-effector pose but also another constrains along the kinematics chain [88, 1]. In such scenario, a set of r time-varying tasks $\{\mathbf{r}_i\}_{i=1}^r$, providing s_i constraints each, may be combined to define a single *augmented* task constraint $\mathbf{r}_\mathcal{A}$. It is worth noting that the total number of constraints s , imposed by the set of task $\{\mathbf{r}_i\}_i$, must be

$$s = \sum_{i=1}^r s_i \leq n$$

to avoid overdetermination of the problem. By denoting with \mathbf{J}_i the corresponding Jacobian of each task \mathbf{r}_i , we have

$$\mathbf{r}_\mathcal{A} = \begin{pmatrix} \mathbf{r}_1 \\ \vdots \\ \mathbf{r}_r \end{pmatrix} \implies \mathbf{J}_\mathcal{A} = \begin{pmatrix} \mathbf{J}_1 \\ \vdots \\ \mathbf{J}_r \end{pmatrix}$$

where each block \mathbf{J}_i is an $(s_i \times n)$ matrix and $\mathbf{J}_\mathcal{A}$ represents the augmented or extended Jacobian of task $\mathbf{r}_\mathcal{A}$. In principle any of the method already discussed for the redundancy resolution may be used to compute the optimal velocities \mathbf{q}^* minimizing some objective function $H(\dot{\mathbf{q}})$, subject to the constraint $\dot{\mathbf{r}}_\mathcal{A} = \mathbf{J}_\mathcal{A} \dot{\mathbf{q}}^*$. However, in general, the rank of $\mathbf{J}_\mathcal{A}$ is not always s due to the presence of algorithmic singularities. Such singularities arise when one or more constraints in the set $\{\mathbf{r}_i\}_{i=1}^r$ are linearly dependent. This imply that, for at least one pair of tasks \mathbf{r}_j and \mathbf{r}_k , $j \neq k$, we have

$$\mathcal{R}(\mathbf{J}_j^\top) \cap \mathcal{R}(\mathbf{J}_k^\top) \neq \emptyset \iff \text{rank} \begin{pmatrix} \mathbf{J}_j \\ \mathbf{J}_k \end{pmatrix} < \text{rank}(\mathbf{J}_j) + \text{rank}(\mathbf{J}_k) = s_j + s_k$$

To avoid algorithmic singularities it is therefore necessary to enforce empty intersections of the range spaces for each pair of tasks in $\{\mathbf{r}_i\}_{i=1}^r$

$$\mathcal{R}(\mathbf{J}_j^\top) \cap \mathcal{R}(\mathbf{J}_k^\top) = \emptyset \quad \forall j \in \{i\}_{\substack{i=1 \\ i \neq k}}^r, \quad \forall k \in \{i\}_{\substack{i=1 \\ i \neq j}}^r \quad (2.25)$$

condition difficult to meet along the complete motion of the robot while executing the augmented task.

Alternatively, one can leave unconstrained the intersection of the range spaces $\mathcal{R}(\mathbf{J}_i^\top)$ and introduce a prioritized scheme [88, 37] to map lower priority tasks constraints in the null space of higher priority tasks. Let's assume that a set of tasks $\{\mathbf{r}_i\}_{i=1}^r$ is given, with decreasing priority with respect i . Then, we can start by computing the a generalized optimal (possibly weighted) least norm solution, for the first task \mathbf{r}_1

$$\dot{\mathbf{q}}_1^* = \mathbf{J}_1^\dagger \dot{\mathbf{r}}_1 + (\mathbf{I} - \mathbf{J}_1^\dagger \mathbf{J}_1) \dot{\mathbf{q}}_2$$

now we can choose $\dot{\mathbf{q}}_2^*$ so as to satisfy, if possible, the second task. That is, determine the generalized (possibly weighted) least norm solution of $\dot{\mathbf{q}}_2^*$ subject to the task constraint $\dot{\mathbf{r}}_2 - \mathbf{J}_2 \dot{\mathbf{q}}_1$. Such constraint can be rewritten as

$$\dot{\mathbf{r}}_2 - \mathbf{J}_2 \dot{\mathbf{q}}_1 = \dot{\mathbf{r}}_2 - \mathbf{J}_2 \mathbf{J}_1^\dagger \dot{\mathbf{r}}_1 - \mathbf{J}_2 (\mathbf{I} - \mathbf{J}_1^\dagger \mathbf{J}_1) \dot{\mathbf{q}}_2 = \dot{\tilde{\mathbf{r}}}_2 - \tilde{\mathbf{J}}_2 \dot{\mathbf{q}}_2$$

where $\dot{\tilde{\mathbf{r}}}_2$ accounts for the contribution induced by the solution of the priority task \mathbf{r}_1 over \mathbf{r}_2 and $\tilde{\mathbf{J}}_2$ correspond to the projection of \mathbf{J}_2 into the null space $\mathcal{N}(\mathbf{J}_1)$. Therefore, only the rows of \mathbf{J}_2 belonging to $\mathcal{N}(\mathbf{J}_1)$ will be employed in the next optimization step. The generalized (possibly weighted) least norm solution of $\dot{\mathbf{q}}_2^*$ is thus given by

$$\dot{\mathbf{q}}_2^* = \tilde{\mathbf{J}}_2^\dagger \dot{\tilde{\mathbf{r}}}_2 + (\mathbf{I} - \tilde{\mathbf{J}}_2^\dagger \tilde{\mathbf{J}}_2) \dot{\mathbf{q}}_3$$

This process is then iteratively repeated up to the last task of the set. It is worth noticing that when computing the optimal velocity for the last task of the set, namely $\dot{\mathbf{q}}_r^*$, it is enough to compute a (possibly weighted) least norm solution, instead of a generalized one. The whole algorithm can be summarized as follows

$$\dot{\mathbf{q}}_i^* = \begin{cases} \mathbf{J}_i^\dagger \dot{\mathbf{r}}_i + (\mathbf{I} - \mathbf{J}_i^\dagger \mathbf{J}_i) \dot{\mathbf{q}}_{i+1} & \text{if } i=1, \\ \tilde{\mathbf{J}}_i^\dagger \dot{\tilde{\mathbf{r}}}_i + (\mathbf{I} - \tilde{\mathbf{J}}_i^\dagger \tilde{\mathbf{J}}_i) \dot{\mathbf{q}}_{i+1} & \text{if } 2 < i < r-1, \\ \tilde{\mathbf{J}}_i^\dagger \dot{\tilde{\mathbf{r}}}_i & \text{if } i=r. \end{cases}$$

Let us conclude the section with some remarks. First, in the case of algorithmic singularities at task \mathbf{r}_i , a regularized method should be used to compute an optimal bounded velocity \mathbf{q}_i^* . In such a case, only the current task is affected by the damping effect, but will accumulate errors induced by all velocities \mathbf{q}_j^* associated to lower priority tasks \mathbf{r}_j , with $j > i$, because of the improper definition of the orthogonal projector (2.24) into the null space $\mathcal{N}(\tilde{\mathbf{J}}_i)$. Second, the prioritized control scheme ensures that for any successive pair of tasks \mathbf{r}_i and \mathbf{r}_{i+1} , $i=1, \dots, r-1$, the following equality holds

$$\mathcal{R}(\tilde{\mathbf{J}}_{i+1}^\top) \subset \mathcal{R}(\tilde{\mathbf{J}}_i^\top)^\perp \equiv \mathcal{N}(\tilde{\mathbf{J}}_i)$$

which represents a stronger condition than (2.25). Finally, both augmentation and priority schemes can be combined to produce mixed control strategies, specially when multiple tasks share the same priority, like the remote center of motion (RCM) constraint discussed in Section 2.1.4.

2.1.3 Manipulability measure

Given a feasible task $\mathbf{r}(t) \in \mathcal{W}_r$, $\mathcal{W}_r \subset \mathbb{R}^p$, and a suitably redundant robot for its execution, then it is possible to identify a robot motion $\mathbf{q}(t) \in \mathcal{Q}$, $\mathcal{Q} \subset \mathbb{R}^n$, satisfying the task constraints $\mathbf{r}(t) = f(\mathbf{q}(t))$, at each time instant $t \in \mathbb{R}^+$, and assessing the best performance respect to some quality index or measure during the task execution. One such a measure is given by the *manipulability* [110], which quantifies the robot ability to freely maneuver or apply static generalized forces in workspace. The present paragraph is devoted to the formal introduction of the measure, including its notable properties in robotics control and a brief overview of key underlying mechanical concepts.

Statics and kineto-statics duality Let's assume that our robot is at rest at the given configuration \mathbf{q} . In such static equilibrium, it is possible to determine the relation between the vector $\boldsymbol{\tau}$ of joint generalized forces and the vector $\boldsymbol{\xi}$ of external generalized forces applied to the end-effector, through the application of the *principle of virtual work*⁶. To this end, let's assume that $\boldsymbol{\xi}$ has the form

$$\boldsymbol{\xi} = \begin{pmatrix} \xi_l \\ \xi_r \end{pmatrix}$$

⁶The principle states that the virtual work of the applied generalized forces is zero for all virtual displacements of the robot from static equilibrium.

where ξ_l represents the force contributions (linear forces) and ξ_r the moment of force contributions (rotational forces). Moreover, let's denote with $\delta\mathbf{p}$ and $\delta\theta$, respectively, the linear and angular virtual displacements of the end-effector. Since the mechanical structure of the robot is defined by a set of scleronomic holonomic constraints, the virtual configuration displacements $\delta\mathbf{q}$ coincide with the elementary displacement $d\mathbf{q}$. As consequence, the differential kinematics relation allows to map virtual displacements $\delta\mathbf{q}$ of the configuration space to virtual displacements in the workspace

$$\begin{pmatrix} \delta\mathbf{p} \\ \delta\theta \end{pmatrix} = \mathbf{J}(\mathbf{q})\delta\mathbf{q}$$

Then, by the principle of virtual work we have that

$$\delta W = \tau^\top \delta\mathbf{q} - \xi_l^\top \delta\mathbf{p} - \xi_r^\top \delta\theta = (\tau^\top - \xi^\top \mathbf{J}(\mathbf{q}))\delta\mathbf{q} = 0 \implies \tau = \mathbf{J}(\mathbf{q})^\top \xi \quad (2.26)$$

stating that the relationship between the external generalized forces ξ applied to the end-effector and the joint generalized forces τ is established by the transpose of the Jacobian matrix $\mathbf{J}(\mathbf{q})$. Therefore, at a given configuration \mathbf{q} , the range space $\mathcal{R}(\mathbf{J}^\top(\mathbf{q}))$ represents the subspace of \mathbb{R}^n where the joint generalized forces τ can balance the external generalized forces ξ exerted over the end-effector. Moreover, the null space $\mathcal{N}(\mathbf{J}^\top(\mathbf{q}))$ correspond to the subspace of \mathbb{R}^n where the generalized forces ξ applied to end-effector are entirely absorbed by the mechanical structure of the robot.

Robot manipulability Given a non-singular robot configuration $\mathbf{q} \in \mathcal{Q}$, and its corresponding end-effector pose $\mathbf{r} = f(\mathbf{q}) \in \mathcal{W}$, it is possible to quantify the robot ability to arbitrarily change the end-effector pose by identifying the set $\mathcal{V} \subset \mathcal{T}_{\mathbf{r}}\mathcal{W}^7$ of instantaneous end-effector motions generated by the set of unit joint velocities

$$\{\dot{\mathbf{q}} \in \mathcal{T}_{\mathbf{q}}\mathcal{Q} \mid \dot{\mathbf{q}}^\top \dot{\mathbf{q}} = 1\} \quad (2.27)$$

We can consider the general least norm inversion (2.19), which substituted in (2.27) and after some straightforward algebraic computations⁸ gives

$$\dot{\mathbf{q}}^\top \dot{\mathbf{q}} = \dot{\mathbf{r}}^\top (\mathbf{J}\mathbf{J}^\top)^{-1} \dot{\mathbf{r}} + \dot{\mathbf{q}}_0^\top (\mathbf{I} - \mathbf{J}^\dagger \mathbf{J}) \dot{\mathbf{q}}_0$$

⁷ $\mathcal{T}_p M$ denotes the *tangent space* of the manifold M at p .

⁸It is enough to remember that the projector operator $\mathbf{N} = (\mathbf{I} - \mathbf{J}^\dagger \mathbf{J})$ is idempotent and equals its transpose, therefore $\mathbf{N}^\top \mathbf{N} = \mathbf{N}$. Moreover, by definition, it is always verified that $\mathbf{J}^\dagger \mathbf{N} = 0$.

where the explicit dependence of the Jacobian matrix $J(\mathbf{q})$ on the configuration \mathbf{q} has been omitted for better readability, omission that will be held until the end of the section. Therefore, \mathcal{V} can be formally defined as

$$\mathcal{V} = \left\{ \dot{\mathbf{r}} \in \mathcal{T}_{\mathbf{r}}\mathcal{W} \mid \dot{\mathbf{r}}^\top (\mathbf{J}\mathbf{J}^\top)^{-1} \dot{\mathbf{r}} = 1 - \dot{\mathbf{q}}_0^\top (\mathbf{I} - \mathbf{J}^\dagger \mathbf{J}) \dot{\mathbf{q}}_0 \right\} \quad (2.28)$$

representing an ellipsoid in $\mathcal{T}_{\mathbf{r}}\mathcal{W}$, called the *velocity manipulability ellipsoid*. Note that any feasible self-motion $\dot{\mathbf{q}}_0$, when projected in the null space of \mathbf{J} , must have a norm strictly less than the unit to properly define a non-degenerate set \mathcal{V} . However, any non zero self-motion induces a global scale of the manipulability ellipsoid, without affecting any of its intrinsic properties. Hence, without any loss of generality, in the context of manipulability analysis we can always assume $\dot{\mathbf{q}}_0=0$.

The geometric properties of the ellipsoid (2.28) are defined by the core of its quadratic form $\mathbf{J}\mathbf{J}^\top$. Since the eigen-decomposition of the matrix $\mathbf{J}\mathbf{J}^\top$ can be written in terms of the singular value decomposition of \mathbf{J}

$$\mathbf{J} = \mathbf{U}\mathbf{S}\mathbf{V}^\top \implies \mathbf{J}\mathbf{J}^\top = \mathbf{U}\mathbf{S}^2\mathbf{U}^\top$$

it follows that the directions of the principal axis of the manipulability ellipsoid coincides with the columns \mathbf{u}_i of \mathbf{U} (i.e., with the left singular vectors of \mathbf{J}), $i=1, \dots, m$, while the dimensions of the axes with the singular values σ_i of \mathbf{J} . The end-effector can reach the largest velocity along the direction of the major axis of the ellipsoid, and the smallest along the direction of the minor axis. Thus, the eccentricity of the ellipsoid provide a measure of the end-effector ability to isotropically move along all directions of $\mathcal{T}_{\mathbf{r}}\mathcal{W}$. Consequently, we can define a global manipulability measure [110] as⁹

$$w(\mathbf{q}) = \sqrt{\det(\mathbf{J}\mathbf{J}^\top)} = \sqrt{\prod_{i=1}^m \lambda_i(\mathbf{J}\mathbf{J}^\top)} = \prod_{i=1}^m \sqrt{\lambda_i(\mathbf{J}\mathbf{J}^\top)} = \prod_{i=1}^m \sigma_i(\mathbf{J}) \quad (2.29)$$

that is, as the volume of the manipulability ellipsoid. As such, this quantity is always positive except in singular configurations when it becomes zero, thus it can be adopted as a distance measure of of a given configuration \mathbf{q} from a kinematic singularities. However, this measure completely lacks a physical consistency [29], since it is not invariant to scale nor to rigid body transformations. An alternative measure $\tilde{w}(\mathbf{q})$ correspond to the ratio between the minimum and maximum singular values of \mathbf{J} , i.e. the condition number of the Jacobian. Assuming that the singular

⁹In the case of a non-redundant robot, such manipulability measure reduces to $w(\mathbf{q}) = |\det \mathbf{J}|$.

values σ_i in the sequence $\{\sigma_i\}_{i=1}^m$ are given in decreasing order, such scale invariant alternative measure results

$$\tilde{w}(\mathbf{q}) = \frac{\sigma_m^2}{\sigma_1^2} \quad (2.30)$$

and can be related to the ellipsoid eccentricity by $e(\mathbf{q}) = \sqrt{1 - \tilde{w}^2(\mathbf{q})}$.

It is worthy of remark, both manipulability measures (2.29) and (2.30) have a high computational complexity. On real robotics applications, this may represent a bottleneck when the manipulability measure is used to support a decision process on real-time. To alleviate this issue, different precomputed representation of the manipulability measure has been proposed [101, 79] in order to serve manipulability queries efficiently.

On the basis of the duality relation between kinematics and statics (2.26), it is possible to define the (generalized) *force manipulability ellipsoid* as

$$\boldsymbol{\tau}^\top \boldsymbol{\tau} = 1 \implies \boldsymbol{\xi}^\top \mathbf{J} \mathbf{J}^\top \boldsymbol{\xi} = 1 \quad (2.31)$$

Since the core of the quadratic form of (2.31) equals the inverse of the core of (2.28), the force ellipsoid has the same principal axis directions of the velocity ellipsoid, with axis dimensions in inverse proportion. This notable property implies that a good velocity manipulability along some direction implies a poor generalized force control along the same direction and vice versa.

Manipulability maximization Among others uses, manipulability can be successfully exploited as a design quality measure, a proprioceptive stimuli for autonomous actions, a motion planning constraint, a feedback measure for assisted teleoperation and a control optimization goal. To cite a few, Jung et al. [53] demonstrate how an accurate analysis of manipulability and other quality indexes can be used to improve the design of a versatile humanoid robot for teleoperated disaster response. In [41] is addressed the problem of assisting a human operator in a reach-to-grasp task, such as to reduce the control effort of the manipulator during the post-grasp task, by means of force cues computed through a task-oriented velocity manipulability cost function. Torabi et al. [98] defined a manipulability measure of a combined master-slave system for robotic teleoperated minimally invasive surgeries, such measure is used to assist the design choices of the robotic platform, aimed to improve surgeons control, minimize the footprint of the master robot and avoid singularities and joint limits of the master and slave robots. Bayle et al. [4] generalize the standard definition of manipulability to the case

of mobile manipulators with nonholonomic constraints, coupling base locomotion and arm movements for manipulability maximization during the task execution. Vahrenkamp et al. [101] extended the concept of manipulability ellipsoid including constraints coming from joint limits, workspace obstacles or self-distance into its definition, and used such geometrical object to build a manipulability distribution over the workspace, allowing to capture the robot capabilities in terms of operational freedom during task planning and execution. It is worth noticing that, from design to control and planning, manipulability represents an ubiquitous quality measure.

Based on the generalized least norm inversion (2.19), it is possible to maximize the robot manipulability during the task execution by computing $\dot{\mathbf{q}}_0$ as the gradient of a manipulability measure. It is worth noticing that it is always possible to find a closed form solution of (2.29), while it is not the case for the alternative measure (2.30). However, computations of the closed form solution of (2.29) are generally tricky and cumbersome. Moreover, instead of considering (2.29) as the optimization goal, it is computationally more efficient to consider the cost function $H_0(\mathbf{q})=w^2(\mathbf{q})$. With this in mind, a manipulability maximizing self-motion $\dot{\mathbf{q}}_0$ can be defined as [112]

$$\dot{\mathbf{q}}_{0,i} = \eta \left(\frac{\partial}{\partial \mathbf{q}_i} H_0(\mathbf{q}) \right)^\top = \eta \det(\mathbf{J}\mathbf{J}^\top) \operatorname{tr} \left((\mathbf{J}\mathbf{J}^\top)^{-1} \left(\frac{\partial \mathbf{J}}{\partial \mathbf{q}_i} \mathbf{J}^\top + \mathbf{J} \frac{\partial \mathbf{J}^\top}{\partial \mathbf{q}_i} \right) \right)$$

2.1.4 Remote center of motion (RCM) constraint

The remote center of motion is defined by a fixed point $\mathbf{p}_{rcm} \in \mathbb{R}^3$, contained in the workspace \mathcal{W} of the robot, around which part of the robot chain is constrained to rotate. This kinematics constraint is widely known in the field of minimally invasive surgery (MIS) [18, 1, 81, 82], where all surgical tools are constrained to pass through and to rotate around small incisions in the patient's body.

Let's consider a robot manipulator with configuration space $\mathcal{Q} \subset \mathbb{R}^n$ and denote with $\mathbf{p}_i(\mathbf{q}) \in \mathbb{R}^3$ the origin of the link's i reference frame, computed at the configuration $\mathbf{q} \in \mathcal{Q}$, with $i=1, \dots, n$. Also, let's assume that the remote center of motion constraint $\mathbf{p}_{rcm}(\mathbf{q})$ has been imposed to link i of the chain, that is, to lie on the line segment between $\mathbf{p}_i(\mathbf{q})$ and $\mathbf{p}_{i+1}(\mathbf{q})$

$$\mathbf{p}_{rcm}(\mathbf{q}) = \mathbf{p}_i(\mathbf{q}) + \lambda(\mathbf{p}_{i+1}(\mathbf{q}) - \mathbf{p}_i(\mathbf{q})) \quad (2.32)$$

with $0 < \lambda < 1$ indicating the relative distance of $\mathbf{p}_{rcm}(\mathbf{q})$ from $\mathbf{p}_i(\mathbf{q})$. For any given

configuration motion $\mathbf{q}(t) \in \mathcal{Q}$ of the robot, the RCM constraint imposes that

$$\dot{\mathbf{p}}_{rcm}(\mathbf{q}(t)) \equiv 0 \quad \forall t \in \mathbb{R}^+ \quad (2.33)$$

now by differentiating (2.32) with respect to time and imposing the differential constraint (2.33), after some straightforward manipulations we obtain

$$\dot{\mathbf{p}}_{rcm} = \begin{pmatrix} \mathbf{J}_i + \lambda(\mathbf{J}_{i+1} - \mathbf{J}_i) & \mathbf{p}_{i+1} - \mathbf{p}_i \end{pmatrix} \begin{pmatrix} \dot{\mathbf{q}} \\ \dot{\lambda} \end{pmatrix} = \mathbf{J}_{rcm} \begin{pmatrix} \dot{\mathbf{q}} \\ \dot{\lambda} \end{pmatrix} = 0 \quad (2.34)$$

where we have dropped all dependencies for better readability and $\mathbf{J}_i = \mathbf{J}_i(\mathbf{q}(t))$ represents the Jacobian matrix of $\mathbf{p}_i(\mathbf{q})$. It is worth noting that the introduction of the variable λ increases the degrees of freedom of the system by one, while the constraint (2.34) reduces the available number of degrees of freedom by 3. Therefore, the RCM constraint reduces the robot's degrees of freedom by 2. This is an expected result since (2.34) imposes that no motion can be performed along the orthogonal plane of the line joining $\mathbf{p}_i(\mathbf{q})$ to $\mathbf{p}_{i+1}(\mathbf{q})$.

For a given manipulation task $\mathbf{r}(t) \in \mathcal{W}_{\mathbf{r}}$, subject to the RCM constraint (2.34), we can define a control strategy either in terms of the task augmentation scheme or the task priority scheme (giving the highest priority to the most critical task between the two), as described in Section 2.1.2. In the case of task augmentation, dropping all temporal dependencies, we have

$$\dot{\mathbf{r}}_{\mathcal{A}} = \begin{pmatrix} \dot{\mathbf{r}} \\ 0 \end{pmatrix} = \begin{pmatrix} \mathbf{J} & 0 \\ \mathbf{J}_{rcm} \end{pmatrix} \begin{pmatrix} \dot{\mathbf{q}} \\ \dot{\lambda} \end{pmatrix} = \mathbf{J}_{\mathcal{A}} \begin{pmatrix} \dot{\mathbf{q}} \\ \dot{\lambda} \end{pmatrix}$$

In the general case of a redundant manipulator, the generalized least norm solution of the augmented task results

$$\dot{\mathbf{q}}^* = \mathbf{J}_{\mathcal{A}}^\dagger \dot{\mathbf{r}}_{\mathcal{A}} + (\mathbf{I} - \mathbf{J}_{\mathcal{A}}^\dagger \mathbf{J}_{\mathcal{A}}) \begin{pmatrix} \dot{\mathbf{q}}_0 \\ \dot{\lambda}_0 \end{pmatrix}$$

where the self-motions $\dot{\mathbf{q}}_0$ and $\dot{\lambda}$ can be chosen independently of each other. In particular, assuming that the compatibility with $\dot{\mathbf{r}}_{\mathcal{A}}$ is verified, it is possible to modulate the position of the RCM along the line joining $\mathbf{p}_i(\mathbf{q})$ to $\mathbf{p}_{i+1}(\mathbf{q})$ by means of the following quadratic cost function

$$H(\lambda) = \frac{1}{2}(\lambda - \bar{\lambda})^2 \quad \implies \quad \dot{\lambda}_0 = -\eta \left(\frac{\partial H(\lambda)}{\partial \lambda} \right)^\top = \eta(\bar{\lambda} - \lambda)$$

where $\bar{\lambda}$ represents the desired relative distance and $\eta > 0$.

2.2 Dynamics modeling and estimation

End-effector motions and joint efforts can be related through the dynamic model of the manipulator. Such a dynamic model plays a prominent role in the field of robotic design and control. In the former case, synthesized joint generalized forces from predefined task motions provide useful hints for joints, transmissions and actuators design. In the latter case, different control and planning strategies can be tested and validated in a safe virtual environment. Dynamic modeling allows the computation of fast reactive control laws, required in critical safety application like collaborative robotic and autonomous driving.

The rest of the section is organized as follows. Section 2.2.1, briefly describes the Euler-Lagrange equations motions of manipulators. Section 2.2.2 introduce the the dynamic model of mobile manipulators. Finally, an empirical method for the identification of dynamic parameters is described in Section 2.2.3. In the following paragraphs, any explicit time dependency will be omitted for better readability of the formulae.

2.2.1 Euler-Lagrange formulation

Let's consider a manipulator with n rigid links and suppose that, to each link i ¹⁰ of the robot chain, we rigidly attach a reference frame to its center of mass, aligned with the principal axes of inertia of the link, whose pose, at configuration $\mathbf{q} \in \mathcal{Q}$, $\mathcal{Q} \subset \mathbb{R}^n$, will be denoted by $\mathbf{r}_i(\mathbf{q}) \in \mathcal{W}$. The velocity $\dot{\mathbf{r}}_i(\mathbf{q}, \dot{\mathbf{q}})$ of the frame, given an instantaneous velocity $\dot{\mathbf{q}} \in \mathcal{T}_{\mathbf{q}}\mathcal{Q}$, results

$$\dot{\mathbf{r}}_i(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{J}_i(\mathbf{q})\dot{\mathbf{q}} = \begin{pmatrix} \mathbf{J}_{p,i}(\mathbf{q}) \\ \mathbf{J}_{o,i}(\mathbf{q}) \end{pmatrix} \dot{\mathbf{q}}$$

where $\mathbf{J}_{p,i}(\mathbf{q})\dot{\mathbf{q}}$ and $\mathbf{J}_{o,i}(\mathbf{q})\dot{\mathbf{q}}$ provides, respectively, the linear and angular velocities of the link i . Given the mass m_i and moment of inertia $\Upsilon_i(\mathbf{q})$ ¹¹ of the link i , its

¹⁰By link we refer to real link and the actuator's motor altogether.

¹¹ $\Upsilon_i(\mathbf{q})$ is computed with respect the world frame, hence it is a function of the robot configuration \mathbf{q} . In barycentric coordinates such matrix is constant and is defined by the inertia tensor $\tilde{\Upsilon}_i$. Denoting with $\mathbf{R}_i(\mathbf{q})$ the orientation matrix of the link i , whose origin coincides with the link's center of mass, both matrices are related by $\Upsilon_i(\mathbf{q}) = \mathbf{R}_i(\mathbf{q})\tilde{\Upsilon}_i\mathbf{R}_i^\top(\mathbf{q})$.

kinetic $\mathcal{T}_i(\mathbf{q}, \dot{\mathbf{q}})$ and potential $\mathcal{U}_i(\mathbf{q})$ energies are defined as

$$\mathcal{T}_i(\mathbf{q}, \dot{\mathbf{q}}) = \frac{1}{2} \dot{\mathbf{r}}_i^\top(\mathbf{q}, \dot{\mathbf{q}}) \begin{pmatrix} m_i \mathbf{I} & 0 \\ 0 & \Upsilon_i(\mathbf{q}) \end{pmatrix} \dot{\mathbf{r}}_i(\mathbf{q}, \dot{\mathbf{q}}) = \frac{1}{2} \dot{\mathbf{q}}^\top \mathbf{J}_i^\top(\mathbf{q}) \begin{pmatrix} m_i \mathbf{I} & 0 \\ 0 & \Upsilon_i(\mathbf{q}) \end{pmatrix} \mathbf{J}_i(\mathbf{q}) \dot{\mathbf{q}} = \frac{1}{2} \dot{\mathbf{q}}^\top \mathbf{B}_i(\mathbf{q}) \dot{\mathbf{q}}$$

$$\mathcal{U}_i(\mathbf{q}) = -m_i \mathbf{g}_0^\top \mathbf{r}_i(\mathbf{q})$$

where $\mathbf{B}_i(\mathbf{q})$ is a symmetric positive definite ($n \times n$) matrix, representing the generalized inertia matrix of the link i and \mathbf{g}_0 corresponds to the (constant) gravity vector with respect the world frame. For example, in the three dimensional Cartesian space¹², the gravity vector results $\mathbf{g}_0 = (0, 0, -g, 0, 0, 0)^\top$. Then, the total kinetic $\mathcal{T}(\mathbf{q}, \dot{\mathbf{q}})$ and potential $\mathcal{U}(\mathbf{q})$ energies of the robot correspond to

$$\begin{aligned} \mathcal{T}(\mathbf{q}, \dot{\mathbf{q}}) &= \frac{1}{2} \sum_{i=1}^n \dot{\mathbf{q}}^\top \mathbf{B}_i(\mathbf{q}) \dot{\mathbf{q}} = \frac{1}{2} \dot{\mathbf{q}}^\top \left(\sum_{i=1}^n \mathbf{B}_i(\mathbf{q}) \right) \dot{\mathbf{q}} = \frac{1}{2} \dot{\mathbf{q}}^\top \mathbf{B}(\mathbf{q}) \dot{\mathbf{q}} \\ \mathcal{U}(\mathbf{q}) &= - \sum_{i=1}^n m_i \mathbf{g}_0^\top \mathbf{r}_i(\mathbf{q}) = -\mathbf{g}_0^\top \sum_{i=1}^n m_i \mathbf{r}_i(\mathbf{q}) \end{aligned}$$

with $\mathbf{B}(\mathbf{q})$ representing the generalized inertia matrix of the robot. Now, defining the Lagrangian of the robot as

$$\mathcal{L}(\mathbf{q}, \dot{\mathbf{q}}) = \mathcal{T}(\mathbf{q}, \dot{\mathbf{q}}) - \mathcal{U}(\mathbf{q}) = \frac{1}{2} \dot{\mathbf{q}}^\top \mathbf{B}(\mathbf{q}) \dot{\mathbf{q}} - \mathcal{U}(\mathbf{q})$$

the robot's equations of motions are thus given by the Euler-Lagrange equations

$$\frac{d}{dt} \left(\frac{\partial \mathcal{L}(\mathbf{q}, \dot{\mathbf{q}})}{\partial \dot{\mathbf{q}}} \right)^\top - \left(\frac{\partial \mathcal{L}(\mathbf{q}, \dot{\mathbf{q}})}{\partial \mathbf{q}} \right) = \tilde{\tau} \quad (2.35)$$

where $\tilde{\tau}$ assimilates all contributions of non-conservative forces, including the actuators generalized forces τ minus the friction effects and reactions due contact with the environment. Evaluating each partial derivative of (2.35), we have

$$\begin{aligned} \frac{d}{dt} \left(\frac{\partial \mathcal{L}(\mathbf{q}, \dot{\mathbf{q}})}{\partial \dot{\mathbf{q}}} \right)^\top &= \mathbf{B}(\mathbf{q}) \ddot{\mathbf{q}} + \sum_{i=1}^n \left(\frac{\partial \mathbf{b}_i(\mathbf{q})}{\partial \mathbf{q}} \right) \dot{\mathbf{q}} \dot{\mathbf{q}}_i \\ \left(\frac{\partial \mathcal{L}(\mathbf{q}, \dot{\mathbf{q}})}{\partial \mathbf{q}} \right)^\top &= \frac{1}{2} \sum_{i=1}^n \left(\frac{\partial \mathbf{b}_i(\mathbf{q})}{\partial \mathbf{q}} \right)^\top \dot{\mathbf{q}} \dot{\mathbf{q}}_i - \left(\sum_{i=1}^n m_i \mathbf{J}_i^\top(\mathbf{q}) \right) \mathbf{g}_0 \end{aligned}$$

¹²In such a case \mathcal{W} correspond to a six-dimensional manifold.

where $\mathbf{b}_i(\mathbf{q})$ represents the i -th column of the generalized inertia matrix $\mathbf{B}(\mathbf{q})$. Putting all together we obtain the equations of motion

$$\mathbf{B}(\mathbf{q})\ddot{\mathbf{q}} + \left[\sum_{i=1}^n \left(\left(\frac{\partial \mathbf{b}_i(\mathbf{q})}{\partial \mathbf{q}} \right) - \frac{1}{2} \left(\frac{\partial \mathbf{b}_i(\mathbf{q})}{\partial \mathbf{q}} \right)^\top \right) \dot{\mathbf{q}}_i \right] \dot{\mathbf{q}} + \left(\sum_{i=1}^n m_i \mathbf{J}_i^\top(\mathbf{q}) \right) \mathbf{g}_0 = \tilde{\boldsymbol{\tau}}$$

or, equivalently, in compact form

$$\mathbf{B}(\mathbf{q}) \ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \tilde{\boldsymbol{\tau}} \quad (2.36)$$

where the matrix $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}}$ represents the induced centrifugal and Coriolis effects and the vector $\mathbf{g}(\mathbf{q})$ accounts for the joint generalized forces generated by the gravitational force. To make explicit the joint generalized forces $\boldsymbol{\tau}$ in the dynamic model (2.36), we can assume that: viscous friction is given as $\mathbf{F}_v \dot{\mathbf{q}}$, where \mathbf{F}_v is an $(n \times n)$ diagonal matrix of viscous coefficients; static friction (Coulomb friction) is given by $\mathbf{F}_s \text{sgn}(\dot{\mathbf{q}})$, with \mathbf{F}_s is an $(n \times n)$ diagonal matrix and $\text{sgn}(\dot{\mathbf{q}})$ is the component-wise sign function; contact reactions of the end-effector are given by $\mathbf{J}^\top \boldsymbol{\xi}$. Under this set of assumption, the dynamic model of the robot becomes

$$\mathbf{B}(\mathbf{q}) \ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{q}} + \mathbf{F}_v \dot{\mathbf{q}} + \mathbf{F}_s \text{sgn}(\dot{\mathbf{q}}) + \mathbf{g}(\mathbf{q}) = \boldsymbol{\tau} - \mathbf{J}^\top \boldsymbol{\xi} \quad (2.37)$$

To conclude, we highlight some fundamental properties of the model. First, as already mentioned, the generalized inertia matrix $\mathbf{B}(\mathbf{q})$ is positive definite and symmetric matrix, depending only on the robot's configuration \mathbf{q} . This function is upper and lower bounded, that is, such that the following inequalities hold

$$B < \|\mathbf{B}(\mathbf{q})\| < \bar{B}$$

for some positive constants B and \bar{B} , $B < \bar{B}$. As consequence of the above properties, the inverse of the inertia matrix $\mathbf{B}^{-1}(\mathbf{q})$ is also upper and lower bounded.

Second, the matrix $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}}$, representing the centrifugal and Coriolis effects, is quadratic in the joint velocities, thus it is bounded by

$$\|\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\| < C(\mathbf{q}) \|\dot{\mathbf{q}}\|^2$$

for some positive scalar function $C(\mathbf{q})$. In the case of a manipulator having only revolute joints, then $C(\mathbf{q}) = \bar{C}$, with \bar{C} some positive constant value.

Third, from the *principle of conservation of energy*, we know that the total

energy of the system

$$\mathcal{E}(\mathbf{q}, \dot{\mathbf{q}}) = \mathcal{T}(\mathbf{q}, \dot{\mathbf{q}}) + \mathcal{U}(\mathbf{q})$$

remains constant when all generalized forces acting on the robot are balanced, i.e., when $\tilde{\tau}=0$. In other words, when all generalized forces are balanced, then $\dot{\mathcal{E}}(\mathbf{q}, \dot{\mathbf{q}})$ equals zero. By taking such derivative and after few computations we obtain

$$\dot{\mathcal{E}}(\mathbf{q}, \dot{\mathbf{q}}) = \dot{\mathbf{q}}^\top \mathbf{B}(\mathbf{q}) \ddot{\mathbf{q}} + \frac{1}{2} \dot{\mathbf{q}}^\top \dot{\mathbf{B}}(\mathbf{q}) \dot{\mathbf{q}} + \dot{\mathbf{q}}^\top \mathbf{g}(\mathbf{q}) = \frac{1}{2} \dot{\mathbf{q}}^\top (\dot{\mathbf{B}}(\mathbf{q}) - 2\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})) \dot{\mathbf{q}}$$

which implies that $\mathbf{N}(\mathbf{q}, \dot{\mathbf{q}}) = \dot{\mathbf{B}}(\mathbf{q}) - 2\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$ is a skew-symmetric matrix, for all values of \mathbf{q} and $\dot{\mathbf{q}}$. By considering the symmetry of $\mathbf{B}(\mathbf{q})$, thus the symmetry of $\dot{\mathbf{B}}(\mathbf{q})$, we have that

$$\mathbf{N}^\top(\mathbf{q}, \dot{\mathbf{q}}) = -\mathbf{N}(\mathbf{q}, \dot{\mathbf{q}}) \quad \implies \quad \dot{\mathbf{B}}(\mathbf{q}) = \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{C}^\top(\mathbf{q}, \dot{\mathbf{q}}) \quad (2.38)$$

Fourth, both gravity $\mathbf{g}(\mathbf{q})$ and viscous friction $\mathbf{F}_v \dot{\mathbf{q}}$ vectors are also bounded

$$\|\mathbf{g}(\mathbf{q})\| < g(\mathbf{q}) \quad \|\mathbf{F}_v \dot{\mathbf{q}}\| < \bar{F}_v \|\dot{\mathbf{q}}\|$$

with $g(\mathbf{q})$ some positive scalar function and \bar{F}_v some positive constant. In the case of a manipulator having only rotative joints, then $g(\mathbf{q}) = \bar{g}$, with \bar{g} some positive constant value.

Fifth, the dynamic model is linear with respect to the dynamic parameters, that is, it is always possible to rewrite the dynamic model in the form

$$\mathbf{Y}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}) \mathbf{a} = \boldsymbol{\tau}$$

where $\mathbf{a} \in \mathbb{R}^p$ is known as the vector of *dynamic coefficients*. Further insight on this property are provided in Section 2.2.3.

As a last remark, the Euler-Lagrange formulation is systematic and conceptually simple, providing a differential model perfectly suited for the analysis and synthesis of control laws. However, it is not computationally efficient. As an alternative, the Newton-Euler formulation exploits the chain structure of the manipulator and provides a computationally efficient recursive model. For a complete description of the Newton-Euler formulation we refer the reader to [93, 90].

2.2.2 Dynamic model of mobile manipulators

The derivation of the dynamic model of a mobile manipulator follows an analogous procedure as described in Section 2.2.1. The main difference relies in the Euler-Lagrange equation (2.35), due to the inclusion of the motion constraints of mobile platform. Assuming that $\mathbf{q} \in \mathcal{Q}$ is defined as in (2.1), given a transmission mapping $\mathbf{S}(\mathbf{q})$ from the external control inputs \mathbf{u} to the generalized forces τ_b acting on the mobile platform and denoting with $\mathbf{A}(\mathbf{q})$ the transpose of the non-holonomic Pfaffian constraints (2.4), from the constrained Euler-Lagrange equation of the mobile manipulator we obtain

$$\frac{d}{dt} \left(\frac{\partial \mathcal{L}(\mathbf{q}, \dot{\mathbf{q}})}{\partial \dot{\mathbf{q}}} \right)^\top - \left(\frac{\partial \mathcal{L}(\mathbf{q}, \dot{\mathbf{q}})}{\partial \mathbf{q}} \right) = \begin{pmatrix} \tau_a \\ \mathbf{S}(\mathbf{q})\mathbf{u} - \mathbf{A}(\mathbf{q})\lambda \end{pmatrix} \quad (2.39)$$

$$\mathbf{B}(\mathbf{q}) \ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \begin{pmatrix} \tau_a \\ \mathbf{S}(\mathbf{q})\mathbf{u} - \mathbf{A}(\mathbf{q})\lambda \end{pmatrix} \quad (2.40)$$

where τ_a represents the joint generalized forces acting on the manipulator. Based on the decomposition (2.1) of \mathbf{q} , the dynamic model (2.40) can be rewritten as

$$\begin{pmatrix} \mathbf{B}_{aa}(\mathbf{q}) & \mathbf{B}_{ab}(\mathbf{q}) \\ \mathbf{B}_{ab}^\top(\mathbf{q}) & \mathbf{B}_{bb}(\mathbf{q}) \end{pmatrix} \begin{pmatrix} \ddot{\mathbf{q}}_a \\ \ddot{\mathbf{q}}_b \end{pmatrix} + \begin{pmatrix} \mathbf{C}_{aa}(\mathbf{q}, \dot{\mathbf{q}}) & \mathbf{C}_{ab}(\mathbf{q}, \dot{\mathbf{q}}) \\ \mathbf{C}_{ba}(\mathbf{q}, \dot{\mathbf{q}}) & \mathbf{C}_{bb}(\mathbf{q}, \dot{\mathbf{q}}) \end{pmatrix} \begin{pmatrix} \dot{\mathbf{q}}_a \\ \dot{\mathbf{q}}_b \end{pmatrix} + \begin{pmatrix} \mathbf{g}_a(\mathbf{q}) \\ \mathbf{g}_b(\mathbf{q}) \end{pmatrix} = \begin{pmatrix} \tau \\ \mathbf{S}(\mathbf{q})\mathbf{u} - \mathbf{A}(\mathbf{q})\lambda \end{pmatrix} \quad (2.41)$$

Now, differentiating (2.2) with respect to time, we obtain,

$$\ddot{\mathbf{q}}_b = \mathbf{G}(\mathbf{q}_b) \ddot{\tilde{\mathbf{q}}}_b + \dot{\mathbf{G}}(\mathbf{q}_b) \dot{\tilde{\mathbf{q}}}_b \quad (2.42)$$

where the columns of the matrix $\mathbf{G}(\mathbf{q}_b)$ span the null space of the nonholonomic constraints (2.4) of the mobile base and $\dot{\tilde{\mathbf{q}}}_b$ represents the vector of *pseudo-velocities*. It is possible to eliminate the Lagrange multipliers λ in (2.41) by premultiplying the lower set of equations by $\mathbf{G}^\top(\mathbf{q}_b)$. With this idea in mind and substituting (2.2) and (2.42) in (2.41), it is possible to obtain the following reduced dynamic model

$$\tilde{\mathbf{B}}(\mathbf{q}) \ddot{\tilde{\mathbf{q}}} + \tilde{\mathbf{C}}(\mathbf{q}, \dot{\tilde{\mathbf{q}}}) \dot{\tilde{\mathbf{q}}} + \tilde{\mathbf{g}}(\mathbf{q}) = \begin{pmatrix} \tau \\ \mathbf{G}^\top(\mathbf{q})\mathbf{S}(\mathbf{q})\mathbf{u} \end{pmatrix} \quad (2.43)$$

where

$$\begin{aligned}\dot{\mathbf{q}} &= \begin{pmatrix} \dot{q}_a \\ \dot{q}_b \end{pmatrix}, \quad \tilde{\mathbf{G}}(\mathbf{q}) = \begin{pmatrix} \mathbf{I} & 0 \\ 0 & \mathbf{G}(\mathbf{q}) \end{pmatrix}, \quad \tilde{\mathbf{B}}(\mathbf{q}) = \tilde{\mathbf{G}}^\top(\mathbf{q}) \mathbf{B}(\mathbf{q}) \tilde{\mathbf{G}}(\mathbf{q}), \quad \tilde{\mathbf{g}}(\mathbf{q}) = \tilde{\mathbf{G}}^\top(\mathbf{q}) \mathbf{g}(\mathbf{q}) \\ \tilde{\mathbf{C}}(\mathbf{q}, \dot{\mathbf{q}}) &= \tilde{\mathbf{G}}^\top(\mathbf{q}) \left[\mathbf{C}(\mathbf{q}, \tilde{\mathbf{G}}(\mathbf{q}) \dot{\mathbf{q}}) \tilde{\mathbf{G}}(\mathbf{q}) + \mathbf{B}(\mathbf{q}) \dot{\tilde{\mathbf{G}}}(\mathbf{q}) \right],\end{aligned}$$

2.2.3 Identification of dynamic parameters

We have already mention in Section 2.2.1, that the dynamic model can be written as a linear function of the dynamic parameters. The present paragraph aims to provide a formal definition of such parameters and to describe an empirical method for their estimation.

Linear parametrization of the dynamic model As before, let's consider a manipulator with n rigid links and suppose that, to each link i of the robot chain, there is a rigidly attached reference frame aligned with its principal axes of inertia, but centered at any arbitrary point of the link. Also, let's denote with m_i the link's mass and with $\bar{\Upsilon}_i$ its inertia tensor. Let's also assume that the link's center of mass is located at $\mathbf{r}_{c,i}^i \in \mathbb{R}^3$, with $\mathbf{r}_{c,i}^{i\top} = (x_{c,i}, y_{c,i}, z_{c,i})$. The superscript i denotes that quantities are defined with respect the local frame of the link i . Moreover, by the Huygens-Steiner's theorem, the inertia matrix in frame coordinates results

$$\Upsilon_i^i = \bar{\Upsilon}_i + m_i \left[\mathbf{r}_{c,i}^{i\top} \mathbf{r}_{c,i}^i \mathbf{I} - \mathbf{r}_{c,i}^i \mathbf{r}_{c,i}^{i\top} \right] = \bar{\Upsilon}_i + m_i \mathbf{S}^\top(\mathbf{r}_{c,i}^i) \mathbf{S}(\mathbf{r}_{c,i}^i)$$

with $\mathbf{S}(\cdot)$ denoting the skew-symmetric operator.

Now, assuming that the reference frame of the link i is moving with a linear velocity \mathbf{v}_i^i and rotating with angular velocity ω_i^i , the linear velocity of the center of mass results

$$\mathbf{v}_{c,i}^i = \mathbf{v}_i^i + \omega_i^i \times \mathbf{r}_{c,i}^i = \mathbf{v}_i^i + \mathbf{S}(\omega_i^i) \mathbf{r}_{c,i}^i = \mathbf{v}_i^i - \mathbf{S}(\mathbf{r}_{c,i}^i) \omega_i^i$$

Then, the total kinetic and potential energy of link i result

$$\begin{aligned}\mathcal{T}_i(\mathbf{q}, \dot{\mathbf{q}}) &= \frac{1}{2} m_i \left(\mathbf{v}_i^i - \mathbf{S}(\mathbf{r}_{c,i}^i) \omega_i^i \right)^\top \left(\mathbf{v}_i^i - \mathbf{S}(\mathbf{r}_{c,i}^i) \omega_i^i \right) + \frac{1}{2} \omega_i^{i\top} \bar{\Upsilon}_i^i \omega_i^i \\ &= \frac{1}{2} m_i \mathbf{v}_i^{i\top} \mathbf{v}_i^i - m_i \mathbf{v}_i^{i\top} \mathbf{S}(\mathbf{r}_{c,i}^i) \omega_i^i + \frac{1}{2} \omega_i^{i\top} \left(\bar{\Upsilon}_i^i + m_i \mathbf{S}^\top(\mathbf{r}_{c,i}^i) \mathbf{S}(\mathbf{r}_{c,i}^i) \right) \omega_i^i \\ &= \frac{1}{2} m_i \mathbf{v}_i^{i\top} \mathbf{v}_i^i - m_i \mathbf{v}_i^{i\top} \mathbf{S}(\mathbf{r}_{c,i}^i) \omega_i^i + \frac{1}{2} \omega_i^{i\top} \Upsilon_i^i \omega_i^i\end{aligned}$$

As can be observed, the first term on the right side of the last equation is a linear with respect the mass m_i . Also, the second term is linear with respect to the first mass moment $m_i \mathbf{r}_{c,i}^i$, i.e., with respect parameters $m_i x_{c,i}$, $m_i y_{c,i}$ and $m_i z_{c,i}$. Finally, third term is linear with respect to the 6 independent parameters of the inertia matrix. The same applies for the potential energy, which depends linearly on the mass m_i and the first mass moment.

Therefore, for each link i it is possible to define a vector of dynamic parameters $\tilde{\mathbf{a}}_i \in \mathbb{R}^{10}$ such that the kinetic and potential energies are linear with respect to them. Since the Euler-Lagrange equations involve only linear operations over the energy functionals, then the dynamic model (2.37) of the robot is necessarily linear with respect such parameters. Moreover, such model is also linear with respect the coefficients of the matrices \mathbf{F}_v and \mathbf{F}_s modeling, respectively, the viscous and static friction generalized forces.

Denoting by $\tilde{\mathbf{a}} \in \mathbb{R}^{10n}$ the vector of all dynamic parameters of the robot, defined by the concatenation of all $\tilde{\mathbf{a}}_i$, $i=1, \dots, n$, then the dynamic model can be parametrized as the following linear system

$$\tilde{\mathbf{Y}}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}) \tilde{\mathbf{a}} = \boldsymbol{\tau}$$

where the matrix $\tilde{\mathbf{Y}}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})$ has an upper triangular block structure, as a result from the open kinematic chain structure of the manipulator, and only depends on kinematics quantities.

However, in practice for a given robot not all dynamic parameters appear in the dynamic model or may only appear in fixed combinations with others. As consequence, many rows of $\tilde{\mathbf{Y}}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}) \tilde{\mathbf{a}}$ may be zero or linearly dependent. Then, it is necessary to identify a subset of independent parameters $\mathbf{a} \in \mathbb{R}^a$, $a \ll 10n$, such that the regression matrix is well defined

$$\tilde{\mathbf{a}} = \mathbf{T}\mathbf{a} \quad \implies \quad \tilde{\mathbf{Y}}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}) \tilde{\mathbf{a}} = \tilde{\mathbf{Y}}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}) \mathbf{T}\mathbf{a} = \mathbf{Y}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}) \mathbf{a} = \boldsymbol{\tau}$$

where \mathbf{T} is a suitable constant ($10n \times a$) linear map from the set of redundant parameters $\tilde{\mathbf{a}}$ to a set of *dynamic coefficients* \mathbf{a} of the robot and $\mathbf{Y}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})$ is the parametrization of the robot model. One important remark is that only the dynamic coefficients can be identified from experimental data, that is, any standard link parameter (i.e., mass, inertia or coordinates of the center of mass), in general, may not be recovered alone.

Dynamic coefficients identification In principle, given a parametric model of the robot, it is possible to recover the set of dynamic coefficients based on empirical demonstrations. Assuming that a set of samples $(\mathbf{q}_k, \dot{\mathbf{q}}_k, \ddot{\mathbf{q}}_k, \tau_k)$ has been already collected, $k=1, \dots, K$, $K \gg a$, we can try using linear least squares to fit the measured data. The first step is to build the regression matrix $\hat{\mathbf{Y}}$ based on the parametric model and the collected data

$$\hat{\mathbf{Y}}_k = \mathbf{Y}(\mathbf{q}_k, \dot{\mathbf{q}}_k, \ddot{\mathbf{q}}_k) \quad \Longrightarrow \quad \hat{\mathbf{Y}} = \begin{pmatrix} \hat{\mathbf{Y}}_1 \\ \vdots \\ \hat{\mathbf{Y}}_K \end{pmatrix}$$

Now, denoting by $\hat{\boldsymbol{\tau}} = (\tau_1, \dots, \tau_K)^\top$, we can state the identification problem as

$$\mathbf{a}^* = \underset{\mathbf{a} \in \mathbb{R}^a}{\operatorname{argmin}} \left(\hat{\boldsymbol{\tau}} - \hat{\mathbf{Y}}\mathbf{a} \right)^\top \left(\hat{\boldsymbol{\tau}} - \hat{\mathbf{Y}}\mathbf{a} \right)$$

and, as described in Section 2.1.2, the optimal estimate of the dynamic coefficients of the robot model are thus given by

$$\mathbf{a}^* = \hat{\mathbf{Y}}^\dagger \hat{\boldsymbol{\tau}} = \left(\hat{\mathbf{Y}}^\top \hat{\mathbf{Y}} \right)^{-1} \hat{\mathbf{Y}}^\top \hat{\boldsymbol{\tau}}$$

However, in practice, some fundamental issues need to be taken into account before proceeding with the computation of the optimal estimate. First, it is necessary to ensure that the motion trajectory followed by the robot during the experiments excites all the components of the dynamic model. The standard approach, introduced by [95], defines the robot trajectory in terms of a finite Fourier series, allowing time-domain data averaging, noise estimation, calculation of the joint velocities and accelerations in an analytic way and specification of the bandwidth of the excitation trajectories. Second, an iterative estimation of the model [10] allow to compute accurate bounds of the input noise and to eliminate spurious measurement outliers that notably degrade the quality of the estimate. Moreover, the covariance matrix of the noise can be used to weight the least squares, increasing accuracy and robustness of the solution. Third, the computed parametric model, based on rigid body dynamics, always represents an approximation of the real dynamics of the robot. In high-performance robot control the missing dynamics knowledge is essential for improved performance and robustness [58], therefore in such cases accounting for friction and elasticity (due flexible joints or distributed link flexibilities) in the dynamic model is a must. Fourth, often direct measurements of physical quantities is not available in industrial robotics

platforms [40]. This implies that the identification of the robot parameters is subject to indirect measures provided by the manufacturer, since there's no direct access to the sensor measurements nor to real robot commands. Last but not least, alternatives optimization approaches for system identification allow to formulate physical-consistency constraints on the dynamic coefficients, accelerating convergence to a global optimum and increasing robustness to noisy data [107].

2.3 Incipient collision detection

With the advent of autonomous robotics, collaborative robotics, rehabilitation and health care robotics, etc., the classical paradigm of robotics safety by segregation has become obsolete. Indeed, nowadays many research efforts, including design strategies, control methods, planning techniques and perceptual awareness, are focused in the concept of *intrinsic safety*. In such context, two main issues need to be addressed [24, 48]: the protection of the dynamic environment –specially in the presence of humans– and the protection of the robotic structure. The first and most critical issue, in terms of situational awareness, is prediction and prevention of risky circumstances. The second, in terms of self-sensing capabilities, is proprioceptive awareness and prompt compliant reactions after unexpected risky interactions.

It is worth noticing that, among the wide variety of risky or hazardous circumstances, the most dangerous risk specific to robots are unexpected collisions or unwanted force exertions between the machine and the environment [89]. Collision avoidance implies foreseeing dangerous situations, and thus relies on sensing, situational awareness, planing and decision making capabilities. When an unexpected collision occurs, impact forces may be eased through lightweight design and compliant mechanisms and control. Prompt reactions after incipient collisions requires real-time fault diagnosis, fast reflexes and intelligent compensatory behaviours to recover safe operative conditions. The scope of next paragraph is to explore one particular method in the area of real-time fault diagnosis.

Real-time fault detection Fault diagnosis may refer either to detection of unexpected events (fault detection), or to identifying the type and causes of such incidents (fault isolation), or both processes as a whole [7]. However, the scope of this section is limited to the detection of faults due unexpected collisions with the environment, thus we refer to fault diagnosis as the only fault detection process. It is worth mentioning that in real robotics applications there are plenty

of fault sources (sensor failures, actuator failures, network delays, unstructured disturbances, etc.) that may affect the robotic system. In Chapter 4 we consider a less conservative setup, where unexpected collisions and different unstructured disturbances are taken into account.

There exist two main approaches for fault detection: signal based and model based. Signal based approaches search for faulty patterns over sensor readings. Instead, model based approaches compute a *residual signal*, measuring the discrepancy between sensor readings and predicted (nominal) values. In the case of dynamical systems, pure signal based approaches lead to inaccurate results since the dynamics of the system induce deviations from the nominal fault-free operation that cannot be predicted without proper dynamics modeling [28]. Moreover, in the case of incipient collision detection, the use of specific external sensors (vision, strain gauges, load cells, etc.) increases the cost and complexity of the robotics system [22]. Besides, model based approaches are subject to accurate identification of the dynamics coefficients of the model [39] and may require full-state feedback, including accelerations which, in practice, are inherently noisy due to numerical estimation [28]. To deal with those issues, De Luca and Mattone [21, 22] proposed a sensorless method for detection and isolation of faults based on a dynamic observer of the generalized momenta of the robot, that does not require acceleration estimates nor simulation of the entire nominal dynamics. The rest of the section is devoted to the description of this fault detection scheme.

Sensorless collision detection In the following, any explicit time dependency will be omitted for better readability. With reference to the dynamic model of the manipulator (2.37), let's consider the following disturbed dynamic model

$$\mathbf{B}(\mathbf{q}) \ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{q}} + \mathbf{F}_v \dot{\mathbf{q}} + \mathbf{F}_s \operatorname{sgn}(\dot{\mathbf{q}}) + \mathbf{g}(\mathbf{q}) = \boldsymbol{\tau} - \boldsymbol{\tau}_f \quad (2.44)$$

where $\boldsymbol{\tau}_f$ represents the unknown (faulty) joint generalized force. The generalized momenta of the robot is defined as

$$\mathbf{p}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{B}(\mathbf{q}) \dot{\mathbf{q}} \quad (2.45)$$

By taking the time derivative of (2.45) and considering the expressions (2.44) and (2.38), we obtain the following first order ordinary differential equation

$$\dot{\mathbf{p}}(\mathbf{q}, \dot{\mathbf{q}}) = \dot{\mathbf{B}}(\mathbf{q}) \dot{\mathbf{q}} + \mathbf{B}(\mathbf{q}) \ddot{\mathbf{q}} = \boldsymbol{\tau} - \boldsymbol{\tau}_f - \mathbf{n}(\mathbf{q}, \dot{\mathbf{q}})$$

where $\mathbf{n}(\mathbf{q}, \dot{\mathbf{q}})$ is given by

$$\mathbf{n}(\mathbf{q}, \dot{\mathbf{q}}) = -\mathbf{C}^T(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{q}} + \mathbf{F}_v \dot{\mathbf{q}} + \mathbf{F}_s \operatorname{sgn}(\dot{\mathbf{q}}) + \mathbf{g}(\mathbf{q})$$

Now, let's define the residual vector \mathbf{r} as

$$\mathbf{r} = \mathbf{K} \left[\int (\tau - \mathbf{n}(\mathbf{q}, \dot{\mathbf{q}}) - \mathbf{r}) dt - \mathbf{p}(\mathbf{q}, \dot{\mathbf{q}}) \right] \quad (2.46)$$

where \mathbf{K} is a positive definite diagonal matrix. As can be observed, (2.46) does not depend on the joint accelerations $\ddot{\mathbf{q}}$ nor in the inversion of the generalized inertia matrix $\mathbf{B}(\mathbf{q})$. It is possible to show that the residual dynamics represents linear exponentially stable system, driven by the faulty joint generalized force τ_f

$$\dot{\mathbf{r}} = -\mathbf{K}\mathbf{r} + \mathbf{K}\tau_f$$

Finally, an efficient computation of the residual vector can be obtained through a nonlinear dynamic observer [26] of the form

$$\begin{aligned} \dot{\hat{\mathbf{p}}} &= \tau - \mathbf{n}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{K}(\mathbf{p}(\mathbf{q}, \dot{\mathbf{q}}) - \hat{\mathbf{p}}) \\ \mathbf{r} &= \mathbf{K}(\mathbf{p}(\mathbf{q}, \dot{\mathbf{q}}) - \hat{\mathbf{p}}) \end{aligned} \quad (2.47)$$

2.4 Reactive control of mobile manipulators

Controlling involve an action or a sequence of actions regulating the behaviour of a measured physical quantity. The control goal is to make such behaviour mimic a desired one, regardless the presence of disturbances. Accordingly, controlling a mobile manipulator consist in determining a time series of commands (either generalized forces or velocities) so as to guarantee the proper execution of a given task, even in the presence of disturbances. Reactive control imply the real-time selection of such control commands in response to short-term sensory information.

The present paragraph only provides a brief review of classical reactive control-schemes used during the implementation of the CAROTE framework. For a deeper review on reactive control schemes, specially in the case of uncertain systems and optimal control approaches, we refer the reader to [19, 111, 106] and the bibliography therein. The rest of the paragraph is organized as follows. In Section 2.4.1 the fundamental kinematics-based and dynamics-based control schemes of manipulators are introduced. Then, Section 2.4.2 describes specific reactive control strategies for mobile robots. In the following paragraphs, any explicit time dependency or configuration dependency will be omitted for better readability.

2.4.1 Control of manipulators

Let's us assume that a given manipulator is modeled with a set \mathbf{q} of generalized coordinates, defined inside a configuration space $\mathcal{Q} \subset \mathbb{R}^n$, whose end-effector can reach any pose \mathbf{r} in the workspace $\mathcal{W} \subset \mathbb{R}^m$. Given a time-varying feasible task $\mathbf{r}_d \in \mathcal{W}$, our goal is to determine a control law such that for some finite time $T \in \mathbb{R}^+$ and a small positive constant ϵ , the norm of the pose error $\mathbf{e} = \mathbf{r}_d - \mathbf{r}$ satisfies that

$$\|\mathbf{e}\| = \|\mathbf{r}_d - \mathbf{r}\| \leq \epsilon \quad \forall t \geq T \quad (2.48)$$

To this end, let's compute the time derivative of the error

$$\dot{\mathbf{e}} = \frac{d}{dt}(\mathbf{r}_d - \mathbf{r}) = \dot{\mathbf{r}}_d - \dot{\mathbf{r}} = \dot{\mathbf{r}}_d - \mathbf{J}\dot{\mathbf{q}} \quad (2.49)$$

and observe that to satisfy the stability condition (2.48) we need to compute a suitable value of the joint velocities $\dot{\mathbf{q}} = \dot{\mathbf{q}}(\mathbf{e})$ such that equation (2.49) becomes an asymptotically stable system of differential equations, i.e., such that the error converges to zero.

Jacobian inversion An intuitive choice for the control action could be to apply both a proportional and a derivative correction terms. It is possible to map such control action into joint velocities through the Jacobian inversion. As described in Section 2.1.2, in the general case of a redundant manipulator, we have

$$\dot{\mathbf{q}} = \mathbf{J}^\dagger(\dot{\mathbf{r}}_d + \mathbf{K}\mathbf{e}) + (\mathbf{I} - \mathbf{J}^\dagger\mathbf{J})\dot{\mathbf{q}}_0 \quad (2.50)$$

with \mathbf{K} some constant ($m \times m$) symmetric positive definite matrix. By substituting (2.50) in (2.49) we obtain

$$\dot{\mathbf{e}} = \dot{\mathbf{r}}_d - \mathbf{J}[\mathbf{J}^\dagger(\dot{\mathbf{r}}_d + \mathbf{K}\mathbf{e}) + (\mathbf{I} - \mathbf{J}^\dagger\mathbf{J})\dot{\mathbf{q}}_0] = -\mathbf{K}\mathbf{e} \quad \implies \quad \dot{\mathbf{e}} + \mathbf{K}\mathbf{e} = 0$$

which states that the error dynamics are asymptotically stable, with a convergence rate proportional to the eigenvalues of \mathbf{K} .

Jacobian transpose Let's consider the following Lyapunov function

$$V(\mathbf{e}) = \frac{1}{2}\mathbf{e}^\top \mathbf{K}\mathbf{e}$$

with \mathbf{K} some constant ($m \times m$) symmetric positive definite matrix. Such function is always positive $\forall \mathbf{e} \neq 0$ and zero when the error equals zero, i.e., $V(0) = 0$. Its first

order time derivate is given by

$$\dot{V}(\mathbf{e}) = \mathbf{e}^\top \mathbf{K} \dot{\mathbf{e}} = \mathbf{e}^\top \mathbf{K}(\dot{\mathbf{r}}_d - \dot{\mathbf{r}}) = \mathbf{e}^\top \mathbf{K}(\dot{\mathbf{r}}_d - \mathbf{J}\dot{\mathbf{q}}) = \mathbf{e}^\top \mathbf{K}\dot{\mathbf{r}}_d - \mathbf{e}^\top \mathbf{K}\mathbf{J}\dot{\mathbf{q}} \quad (2.51)$$

Now, let's define the control action as $\dot{\mathbf{q}} = \mathbf{J}^\top \mathbf{K} \mathbf{e}$ and substitute it in (2.51).

$$\dot{V}(\mathbf{e}) = \mathbf{e}^\top \mathbf{K}\dot{\mathbf{r}}_d - \mathbf{e}^\top \mathbf{K}\mathbf{J}\mathbf{J}^\top \mathbf{K} \mathbf{e} \quad (2.52)$$

In the case of a constant task, i.e., $\dot{\mathbf{r}}_d = 0$, equation (2.52) could be either negative definite or negative semi-definite, depending on the rank of the Jacobian matrix. When \mathbf{J} is full rank, $\dot{V}(\mathbf{e})$ is negative definite and, as consequence, the error asymptotically converges to zero. Instead, when \mathbf{J} has a nonempty null space $\mathcal{N}(\mathbf{J})$, for any error vector \mathbf{e} such that $\mathbf{K}\mathbf{e} \in \mathcal{N}(\mathbf{J})$, we have that $\dot{V}(\mathbf{e}) = 0$. Which implies that the robot can get stuck without zeroing the error. However, such a condition implies that the required task lies outside the bounds of the workspace of the manipulator.

In the case of a time-varying task, i.e., $\dot{\mathbf{r}}_d \neq 0$, it is not possible to achieve asymptotically stability with the chosen control action. However, it can be shown that the norm of the error is bounded in terms of the norm of the matrix \mathbf{K} .

Jacobian inversion at the acceleration level Differentiating (2.49) with respect to time, leads to the following relation

$$\ddot{\mathbf{e}} = \frac{d}{dt}(\dot{\mathbf{r}}_d - \mathbf{J}\dot{\mathbf{q}}) = \ddot{\mathbf{r}}_d - \dot{\mathbf{J}}\dot{\mathbf{q}} - \mathbf{J}\ddot{\mathbf{q}} \quad (2.53)$$

Now, let's consider the following control law, consisting in proportional and derivative actions plus a correction term

$$\ddot{\mathbf{q}} = \mathbf{J}^\dagger(\ddot{\mathbf{r}}_d + \mathbf{K}_D \dot{\mathbf{e}} + \mathbf{K}_P \mathbf{e} - \dot{\mathbf{J}}\dot{\mathbf{q}}) + (\mathbf{I} - \mathbf{J}^\dagger \mathbf{J})\ddot{\mathbf{q}}_0 \quad (2.54)$$

with \mathbf{K}_D and \mathbf{K}_P some constant ($m \times m$) symmetric positive definite matrices. By substituting (2.54) in (2.53) we obtain

$$\ddot{\mathbf{e}} = \ddot{\mathbf{r}}_d - \dot{\mathbf{J}}\dot{\mathbf{q}} - \mathbf{J}[\mathbf{J}^\dagger(\ddot{\mathbf{r}}_d + \mathbf{K}_D \dot{\mathbf{e}} + \mathbf{K}_P \mathbf{e} - \dot{\mathbf{J}}\dot{\mathbf{q}}) + (\mathbf{I} - \mathbf{J}^\dagger \mathbf{J})\ddot{\mathbf{q}}_0] = -\mathbf{K}_D \dot{\mathbf{e}} - \mathbf{K}_P \mathbf{e}$$

which implies that the error dynamics are given by the following asymptotically stable second order linear system of ordinary differential equations

$$\ddot{\mathbf{e}} + \mathbf{K}_D \dot{\mathbf{e}} + \mathbf{K}_P \mathbf{e} = 0$$

PD control with gravity compensation With reference to the dynamic model of the manipulator (2.37), let's consider the following simplified dynamic model

$$\mathbf{B}(\mathbf{q}) \ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{q}} + \mathbf{F}_v \dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \boldsymbol{\tau} \quad (2.55)$$

where static friction forces has been neglected and no interaction with the environment has been assumed. Moreover, let's assume that the given task is constant, i.e., $\dot{\mathbf{r}}_d=0$, and consider the following Lyapunov function

$$V(\dot{\mathbf{q}}, \mathbf{e}) = \frac{1}{2} \dot{\mathbf{q}}^\top \mathbf{B}(\mathbf{q}) \dot{\mathbf{q}} + \frac{1}{2} \mathbf{e}^\top \mathbf{K}_P \mathbf{e}$$

where $\mathbf{B}(\mathbf{q})$ represents the inertia matrix of the dynamic model of the manipulator and \mathbf{K}_P a constant $(m \times m)$ symmetric positive definite matrix. Such function is always positive $\forall (\dot{\mathbf{q}}, \mathbf{e}) \neq 0$ and zero when $(\dot{\mathbf{q}}, \mathbf{e})$ equals zero, i.e., $V(0, 0)=0$. Its first order time derivate is given by

$$\dot{V}(\dot{\mathbf{q}}, \mathbf{e}) = \dot{\mathbf{q}}^\top \mathbf{B}(\mathbf{q}) \ddot{\mathbf{q}} + \frac{1}{2} \dot{\mathbf{q}}^\top \dot{\mathbf{B}}(\mathbf{q}) \dot{\mathbf{q}} - \dot{\mathbf{q}}^\top \mathbf{J}^\top \mathbf{K}_P \mathbf{e} \quad (2.56)$$

Based on (2.55), we can substitute the vector $\mathbf{B}(\mathbf{q}) \ddot{\mathbf{q}}$ in (2.56), to obtain

$$\begin{aligned} \dot{V}(\dot{\mathbf{q}}, \mathbf{e}) &= \dot{\mathbf{q}}^\top \left(\boldsymbol{\tau} - \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{q}} - \mathbf{F}_v \dot{\mathbf{q}} - \mathbf{g}(\mathbf{q}) \right) + \frac{1}{2} \dot{\mathbf{q}}^\top \dot{\mathbf{B}}(\mathbf{q}) \dot{\mathbf{q}} - \dot{\mathbf{q}}^\top \mathbf{J}^\top \mathbf{K}_P \mathbf{e} \\ &= \frac{1}{2} \dot{\mathbf{q}}^\top \left(\dot{\mathbf{B}} - 2\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) \right) \dot{\mathbf{q}} - \dot{\mathbf{q}}^\top \mathbf{F}_v \dot{\mathbf{q}} + \dot{\mathbf{q}}^\top \left(\boldsymbol{\tau} - \mathbf{g}(\mathbf{q}) - \mathbf{J}^\top \mathbf{K}_P \mathbf{e} \right) \\ &= -\dot{\mathbf{q}}^\top \mathbf{F}_v \dot{\mathbf{q}} + \dot{\mathbf{q}}^\top \left(\boldsymbol{\tau} - \mathbf{g}(\mathbf{q}) - \mathbf{J}^\top \mathbf{K}_P \mathbf{e} \right) \end{aligned} \quad (2.57)$$

Now, let's define the control action $\boldsymbol{\tau}$ as

$$\boldsymbol{\tau} = \mathbf{g}(\mathbf{q}) + \mathbf{J}^\top \mathbf{K}_P \mathbf{e} - \mathbf{J}^\top \mathbf{K}_D \mathbf{J} \dot{\mathbf{q}} \quad (2.58)$$

where \mathbf{K}_D is a constant $(m \times m)$ symmetric positive definite matrix. Now substituting (2.58) in (2.57) gives

$$\dot{V}(\dot{\mathbf{q}}, \mathbf{e}) = -\dot{\mathbf{q}}^\top \mathbf{F}_v \dot{\mathbf{q}} - \dot{\mathbf{q}}^\top \mathbf{J}^\top \mathbf{K}_D \mathbf{J} \dot{\mathbf{q}}$$

which implies that the Lyapunov function is always decreasing and, as consequence, the closed loop system is asymptotically stable. It is worth noticing that the control law (2.58) performs a nonlinear compensation of joint generalized gravitational forces and a linear PD control action on the task error.

Inverse dynamics control With reference to the dynamic model of the manipulator (2.37), let's consider the following compact representation

$$\mathbf{B}(\mathbf{q}) \ddot{\mathbf{q}} + \mathbf{n}(\mathbf{q}, \dot{\mathbf{q}}) = \boldsymbol{\tau} \quad (2.59)$$

where no interaction with the environment has been assumed and $\mathbf{n}(\mathbf{q}, \dot{\mathbf{q}})$ accounts for the Coriolis, centrifugal, gravitational and frictional terms. Since the inertia matrix is positive definite it can be inverted for any robot configuration \mathbf{q} , therefore, by defining the nonlinear control action

$$\boldsymbol{\tau} = \mathbf{B}(\mathbf{q})\mathbf{u} + \mathbf{n}(\mathbf{q}, \dot{\mathbf{q}}) \quad (2.60)$$

we obtain the the following linear system

$$\ddot{\mathbf{q}} = \mathbf{u} \quad (2.61)$$

where the new control input \mathbf{u} need to be defined in terms of the task constraints. It is worth noticing that the nonlinear control action (2.60) allow to compensate the whole dynamics of manipulator (2.59) leading to an equivalent decoupled linear system (2.61) with respect the new control input \mathbf{u} . Based on (2.54), we can define the following stabilizing control law

$$\mathbf{u} = \mathbf{J}^\dagger \left(\ddot{\mathbf{r}}_d + \mathbf{K}_D \dot{\mathbf{e}} + \mathbf{K}_P \mathbf{e} - \dot{\mathbf{J}} \dot{\mathbf{q}} \right) + \left(\mathbf{I} - \mathbf{J}^\dagger \mathbf{J} \right) \ddot{\mathbf{q}}_0$$

with \mathbf{K}_D and \mathbf{K}_P some constant $(m \times m)$ symmetric positive definite matrices, which ensures that the task error \mathbf{e} converges to zero asymptotically, with a convergence rate modulated by \mathbf{K}_D and \mathbf{K}_P .

2.4.2 Control of mobile robots

Let's us consider the unicycle model

$$\dot{\mathbf{q}} = \begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} \cos \theta \\ \sin \theta \\ 0 \end{pmatrix} v + \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \omega = \begin{pmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} v \\ \omega \end{pmatrix} = \mathbf{G}(\mathbf{q}) \begin{pmatrix} v \\ \omega \end{pmatrix} \quad (2.62)$$

with $\mathbf{q} \in \mathcal{Q}$, $\mathcal{Q} = \mathbb{R}^2 \times SO(2)$. Given a time-varying feasible task $\mathbf{q}_d \in \mathcal{Q}$, i.e., such that $\dot{\mathbf{q}}_d = \mathbf{G}(\mathbf{q}_d)(v_d \ \omega_d)^\top$ for all $t \in \mathbb{R}^+$, with v_d and ω_d bounded functions with bounded derivatives, our goal is to determine a control law such that the norm of the pose

error

$$\mathbf{e} = \begin{pmatrix} e_1 \\ e_2 \\ e_3 \end{pmatrix} = \begin{pmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_d - x \\ y_d - y \\ \theta_d - \theta \end{pmatrix} \quad (2.63)$$

satisfies the stability condition (2.48). As before, the stability analysis requires the computation of the first order time derivative of the error

$$\begin{aligned} \dot{\mathbf{e}} &= \begin{pmatrix} -\sin \theta & \cos \theta & 0 \\ -\cos \theta & -\sin \theta & 0 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} x_d - x \\ y_d - y \\ \theta_d - \theta \end{pmatrix} \dot{\theta} + \begin{pmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \dot{x}_d - \dot{x} \\ \dot{y}_d - \dot{y} \\ \dot{\theta}_d - \dot{\theta} \end{pmatrix} \\ &= \begin{pmatrix} e_e \omega - (\dot{x} \cos \theta + \dot{y} \sin \theta) + (\dot{x}_d \cos \theta + \dot{y}_d \sin \theta) \\ -e_1 \omega + (\dot{x} \sin \theta - \dot{y} \cos \theta) - (\dot{x}_d \sin \theta - \dot{y}_d \cos \theta) \\ \omega_d - \omega \end{pmatrix} \\ &= \begin{pmatrix} e_e \omega - v + v_d (\cos \theta_d \cos \theta + \sin \theta_d \sin \theta) \\ -e_1 \omega - v_d (\cos \theta_d \sin \theta - \sin \theta_d \cos \theta) \\ \omega_d - \omega \end{pmatrix} = \begin{pmatrix} e_2 \omega + v_d \cos e_3 - v \\ v_d \sin e_3 - e_1 \omega \\ \omega_d - \omega \end{pmatrix} \end{aligned}$$

which, after the introduction of the following input transformation, results

$$\mathbf{u} = \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} = \begin{pmatrix} v_d \cos e_3 - v \\ \omega_d - \omega \end{pmatrix} \implies \begin{pmatrix} \dot{e}_1 \\ \dot{e}_2 \\ \dot{e}_3 \end{pmatrix} = \begin{pmatrix} e_2 \omega + u_1 \\ v_d \sin e_3 - e_1 \omega \\ u_2 \end{pmatrix} \quad (2.64)$$

Nonlinear trajectory tracking Let's consider the following control law

$$\begin{aligned} u_1 &= -k_1(v_d, \omega_d) e_1 \\ u_2 &= -k_2 v_d \frac{\sin e_3}{e_3} e_2 - k_3(v_d, \omega_d) e_3 \end{aligned}$$

with $k_1(\cdot, \cdot)$ and $k_3(\cdot, \cdot)$ bounded functions with bounded derivatives and k_2 a positive constant. The closed loop error dynamics results

$$\begin{aligned} \dot{e}_1 &= e_2 \omega - k_1(v_d, \omega_d) e_1 \\ \dot{e}_2 &= -e_1 \omega + v_d \sin e_3 \\ \dot{e}_3 &= -k_2 v_d \frac{\sin e_3}{e_3} e_2 - k_3(v_d, \omega_d) e_3 \end{aligned} \quad (2.65)$$

Now, let's define the following positive definite Lyapunov function

$$V(\mathbf{e}) = \frac{k_2}{2}(e_1^2 + e_2^2) + \frac{e_3^2}{2}$$

Its first order derivative along the trajectories of the system results

$$\dot{V}(\mathbf{e}) = k_2(e_1\dot{e}_1 + e_2\dot{e}_2) + e_3\dot{e}_3 = -k_2k_1(v_d, \omega_d)e_1^2 - k_3(v_d, \omega_d)e_3^2$$

Such derivative is negative-semidefinite and, as consequence, the Lyapunov function candidate $V(\mathbf{e})$ is bounded from below and tends to a limit value as $t \rightarrow \infty$.

Omitting the explicit functional dependencies of $k_1(\cdot, \cdot)$ and $k_3(\cdot, \cdot)$ for better readability, the second order derivative of $V(\mathbf{e})$ along the trajectories of the system results

$$\ddot{V}(\mathbf{e}) = -k_2e_1^2(\dot{k}_1 - k_1^2) - e_3^2(\dot{k}_3 - k_3^2) + k_2e_2(k_3v_d \sin e_3 - k_1e_1\omega)$$

As can be observed, this function is bounded from above, since we have already demonstrated that the error is bounded and, by assumption, $k_1(\cdot, \cdot)$, $k_3(\cdot, \cdot)$, v_d and ω_d are bounded with bounded derivatives. This implies that the first order derivative $\dot{V}(\mathbf{e})$ of the Lyapunov function candidate is uniformly continuous and, by Barbălat's lemma [32], converges to zero as $t \rightarrow \infty$. As a result, $e_1 \rightarrow 0$ and $e_3 \rightarrow 0$ as $t \rightarrow \infty$. From this results and considering the closed loop error dynamics (2.65) it is possible to show that

$$\lim_{t \rightarrow \infty} (\omega_d^2 + v_d^2)e_2 = 0$$

which states that the asymptotically stability of the system is guaranteed as long as one of the input references is persistent on time.

Input/output linearization Let's assume that we would like to follow a Cartesian trajectory (x_d, y_d) , that has been assigned to some output point $\mathbf{q}_b \in \mathcal{Q}$ of the form

$$\mathbf{q}_b = \begin{pmatrix} x + b \cos \theta \\ y + b \sin \theta \\ \theta \end{pmatrix}$$

with b some constant value. Note that the Cartesian coordinates (x_b, y_b) of \mathbf{q}_b lie along the sagittal axis of the unicycle, in front or behind the contact point of the

wheel depending on the sign of b . Based on the kinematic model (2.62), the output velocity results

$$\dot{\mathbf{q}}_b = \begin{pmatrix} v \cos \theta - b\omega \sin \theta \\ v \sin \theta + b\omega \cos \theta \\ \omega \end{pmatrix} = \begin{pmatrix} \cos \theta & -b \sin \theta \\ \sin \theta & b \cos \theta \\ 0 & 1 \end{pmatrix} \begin{pmatrix} v \\ \omega \end{pmatrix} = \begin{pmatrix} \mathbf{T}(\theta) \\ 0 & 1 \end{pmatrix} \begin{pmatrix} v \\ \omega \end{pmatrix}$$

Assuming that $b \neq 0$, then $\mathbf{T}(\theta)$ is always nonsingular. Therefore, by defining the following input transformation

$$\begin{pmatrix} v \\ \omega \end{pmatrix} = \begin{pmatrix} \cos \theta & \sin \theta \\ -\sin \theta/b & \cos \theta/b \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} = \mathbf{T}^{-1}(\theta) \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} = \mathbf{T}^{-1}(\theta) \mathbf{u}$$

we obtain the following input-output static feedback linearization of the system

$$\dot{\mathbf{q}}_b = \begin{pmatrix} \mathbf{T}(\theta) \\ 0 & 1 \end{pmatrix} \mathbf{T}^{-1}(\theta) \mathbf{u} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ -\sin \theta/b & \cos \theta/b \end{pmatrix} \mathbf{u}$$

Now, by defining the control action as

$$\mathbf{u} = \begin{pmatrix} \dot{x}_d + k_1(x_d - x_b) \\ \dot{y}_d + k_2(y_d - y_b) \end{pmatrix}$$

with k_1 and k_2 two positive constants, then the closed loop systems results globally exponentially stable.

Posture regulation Suppose that the given task \mathbf{q}_b is constant, i.e., $\dot{\mathbf{q}}_b=0$, and without loss of generality, let's assume that the desired configuration correspond to the origin of the configuration space $\mathbf{q}_d=0$. Now, let's introduce the coordinates change $\tilde{\mathbf{q}}=(\rho, \gamma, \delta)^\top$ having the following form

$$\begin{aligned} \rho &= \sqrt{x^2 + y^2} & \dot{\rho} &= -v \cos \gamma \\ \gamma &= \text{Atan2}(y, x) + \pi - \theta & \dot{\gamma} &= v \frac{\sin \gamma}{\rho} - \omega \\ \delta &= \gamma + \theta & \dot{\delta} &= v \frac{\sin \gamma}{\rho} \end{aligned} \quad \Longrightarrow$$

where ρ represents the length of the vector between the Cartesian coordinates of the unicycle and the origin, γ the angle between the sagittal axis of the robot and such vector, and δ the pointing error. Consider the following control action and

the resulting closed loop dynamics

$$\begin{aligned} v &= k_1 \rho \cos \gamma \\ \omega &= k_2 \gamma + k_1(\gamma + k_3 \delta) \frac{\cos \gamma \sin \gamma}{\gamma} \end{aligned} \implies \begin{aligned} \dot{\rho} &= -k_1 \rho \cos^2 \gamma \\ \dot{\gamma} &= -k_2 \gamma - k_1 k_3 \frac{\delta}{\gamma} \cos \gamma \sin \gamma \\ \dot{\delta} &= k_1 \cos \gamma \sin \gamma \end{aligned} \quad (2.66)$$

with k_1 , k_2 and k_3 some positive constants. Now, let's introduce the following positive definite Lyapunov function

$$V(\tilde{\mathbf{q}}) = \frac{1}{2}(\rho^2 + \gamma^2 + k_3 \delta^2)$$

Its first order derivative along the trajectories of the system results

$$\dot{V}(\tilde{\mathbf{q}}) = \rho \dot{\rho} + \gamma \dot{\gamma} + k_3 \delta \dot{\delta} = -k_1 \rho^2 \cos^2 \gamma - k_2 \gamma^2 \leq 0 \quad (2.67)$$

Such derivative is negative-semidefinite, as consequence, the Lyapunov function candidate $V(\tilde{\mathbf{q}})$ is bounded from bellow and tends to a limit value as $t \rightarrow \infty$. The second order derivative of $V(\tilde{\mathbf{q}})$ along the trajectories of the system results

$$\ddot{V}(\tilde{\mathbf{q}}) = -2k_1 \rho \cos \gamma (\dot{\rho} \cos \gamma - \dot{\gamma} \rho \sin \gamma) - 2k_2 \gamma \dot{\gamma}$$

which implies that

$$\begin{aligned} |\ddot{V}(\tilde{\mathbf{q}})| &\leq 2k_1 |\rho \cos \gamma (\dot{\rho} \cos \gamma - \dot{\gamma} \rho \sin \gamma)| + 2k_2 |\gamma \dot{\gamma}| \\ &\leq 2k_1 |\rho| |\dot{\rho}| + 2k_1 \rho^2 |\dot{\gamma}| + 2k_2 |\gamma| |\dot{\gamma}| \\ &\leq 2k_1 |\rho| |\dot{\rho}| + 2(k_1 \rho^2 + k_2 |\gamma|) |\dot{\gamma}| \end{aligned}$$

Because of (2.67), ρ , γ and δ are all bounded. Hence, by virtue of (2.66) also $\dot{\rho}$, $\dot{\gamma}$ and $\dot{\delta}$ are all bounded. Moreover, $\ddot{V}(\tilde{\mathbf{q}})$ is bounded from above, which implies that $\dot{V}(\tilde{\mathbf{q}})$ is uniformly continuous and, by Barbălat's lemma [32], converges to zero as $t \rightarrow \infty$. As a result, $\rho \rightarrow 0$ and $\gamma \rightarrow 0$ as $t \rightarrow \infty$. Finally, by taking the limit $t \rightarrow \infty$ on each equation of the closed loop system (2.66), we obtain that necessarily $\delta \rightarrow 0$ as $t \rightarrow \infty$. Ergo, the closed loop system asymptotically converges to the desired posture.

2.4.3 On-line planning with artificial potential fields

A robot can be thought as a particle \mathbf{q} moving in the configuration space \mathcal{Q} . To avoid obstacles, singular configurations, self-collisions, or any other types of configuration constraint, the motion of such a particle needs to be constrained to lie within the bounds of the *free configuration space* \mathcal{Q}_{free} , i.e., the subset of robot configurations that don't cause a constraint violation.

Often such constraints are given in workspace rather than in configuration space and, in general, mapping such constraints cannot be performed in a closed form. However, it is possible to exploit partial knowledge of the environment to locally influence the motion of the robot so as to avoid a constraint violation. For example, in the surroundings of an obstacle the robot can be *repelled* to move far from it, or might be globally *attracted* towards the goal. It is worth noticing that there is a one-to-one correspondence between the examples and the action of a conservative vector field (electromagnetic, gravitational, etc.). Indeed, the idea behind the artificial potential fields is exactly to define a scalar potential function $\mathcal{U}(\mathbf{q})$, $\mathcal{U}: \mathcal{Q} \rightarrow \mathbb{R}$, whose field action induces a robot motion entirely contained in \mathcal{Q}_{free} . Defining the field action as

$$-\nabla\mathcal{U}(\mathbf{q}) = -\left(\frac{\partial\mathcal{U}(\mathbf{q})}{\partial\mathbf{q}}\right)^\top$$

then, at each configuration $\mathbf{q} \in \mathcal{Q}$, the artificial potential field directs the robot toward the minimum of the potential, by inducing a gradient descent optimization step at each iteration of the control loop.

Clearly, in the presence of multiple constraints, defining a single scalar potential function is not an easy task. A natural alternative could be to define one scalar potential function for each constraint. In general, however, constraints are mostly unilateral, then in practice only two kind of parametric scalar potential functions need to be defined: attractive potentials and repulsive potentials. Consequently, we just need to identify for each constraint the corresponding field type and provide a suitable set of parameters for its definition.

There is a strong connection between the self-motion generation techniques for redundancy exploitation (Section 2.1.2) and the on-line planning scheme with artificial potential fields. Actually, there's no conceptual difference between these approaches, the difference lies in the priority given to the task. In the case of redundancy resolution, the optimization of the scalar objective function is *constrained* into the null space of the Jacobian of the primary task. Here, the task itself is defined by the action of the potential field. Therefore, any method consid-

ered so far for self-motion generation (e.g., manipulability maximization, described in Section 2.1.3) is defined in terms of ad hoc artificial potential fields.

Attractive potential This potential is designed to attract the robot toward a predefined goal \mathbf{q}_g configuration. To this end, the potential must be defined as a strictly increasing monotonic function of the distance to the goal, i.e., the attractive potential must have a unique global minimum at the goal. Also, the potential should be smooth at the goal configuration, to avoid discontinuities in the attractive field. A common choice is to define the attractive potential as a paraboloid with vertex at the goal

$$\begin{aligned}\mathcal{U}_a(\mathbf{q}) &= \frac{k_1}{2}(\mathbf{q}_d - \mathbf{q})^\top(\mathbf{q}_d - \mathbf{q}) = \frac{k_1}{2}\|\mathbf{q}_d - \mathbf{q}\|^2 \\ -\nabla\mathcal{U}_a(\mathbf{q}) &= k_1(\mathbf{q}_d - \mathbf{q})\end{aligned}$$

with k_1 some positive constant. As can be observed, the field action is linear with respect to the configuration error and, as consequence, unbounded. To obtain a bounded field action, it is possible to combine a paraboloid section in the surroundings of the goal with a conic one outside the goal's surroundings

$$\begin{aligned}\mathcal{U}_a(\mathbf{q}) &= \begin{cases} \frac{k_1}{2}\|\mathbf{q}_d - \mathbf{q}\|^2 & \text{if } \|\mathbf{q}_d - \mathbf{q}\| \leq \alpha \\ k_2\|\mathbf{q}_d - \mathbf{q}\| & \text{otherwise} \end{cases} \\ -\nabla\mathcal{U}_a(\mathbf{q}) &= \begin{cases} k_1(\mathbf{q}_d - \mathbf{q}) & \text{if } \|\mathbf{q}_d - \mathbf{q}\| \leq \alpha \\ k_2\frac{\mathbf{q}_d - \mathbf{q}}{\|\mathbf{q}_d - \mathbf{q}\|} & \text{otherwise} \end{cases}\end{aligned}\tag{2.68}$$

where k_1 , k_2 and α are positive constants, such that $k_2 = k_1\alpha$ to ensure a continuous field action.

Repulsive potential The objective is to keep the robot away from a subset $\mathcal{Q}_{taboo} \subset \mathcal{Q} \setminus \mathcal{Q}_{free}$ of robot configurations. Close to the boundary $\partial\mathcal{Q}_{taboo}$, the potential must increase indefinitely to avoid the robot getting close, or worst, collide/entering the forbidden region. Contrary, far from $\partial\mathcal{Q}_{taboo}$, the potential should exert little or no influence over the robot motion. In other words, the potential function must be defined as a strictly decreasing monotonic unbounded function of the distance to the forbidden region, with a singularity at zero. Consequently,

a convenient choice for the repulsive potential is given by

$$\begin{aligned} \mathcal{U}_r(\mathbf{q}) &= \begin{cases} \frac{k_3}{r} \left(\frac{1}{d(\mathbf{q})} - \frac{1}{d_0} \right)^r & \text{if } d(\mathbf{q}) \leq d_0 \\ 0 & \text{otherwise} \end{cases} \\ -\nabla \mathcal{U}_r(\mathbf{q}) &= \begin{cases} \frac{k_3}{d^2(\mathbf{q})} \left(\frac{1}{d(\mathbf{q})} - \frac{1}{d_0} \right)^{r-1} \nabla d(\mathbf{q}) & \text{if } d(\mathbf{q}) \leq d_0 \\ 0 & \text{otherwise} \end{cases} \end{aligned} \quad (2.69)$$

where k_3 is a positive constant, d_0 the range of influence of the field, $r=1,2,\dots$ and

$$d(\mathbf{q}) = \min_{\hat{\mathbf{q}} \in \partial \mathcal{Q}_{taboo}} \|\mathbf{q} - \hat{\mathbf{q}}\|$$

i.e., the distance from the robot configuration to the set \mathcal{Q}_{taboo} . This potential tends to infinity when the robot approaches $\partial \mathcal{Q}_{taboo}$, with a growth rate modulated by the exponent r .

Total potential and the field action In the presence of many constraints over the configuration space, let's say m constraints, we can define a repulsive potential $\mathcal{U}_{r,i}(\mathbf{q})$ for each constraint, $i=1,\dots,m$. Then, the total artificial potential corresponds to the sum of all repulsive potentials plus the attractive one

$$\begin{aligned} \mathcal{U}(\mathbf{q}) &= \mathcal{U}_a(\mathbf{q}) + \sum_{i=1}^m \mathcal{U}_{r,i}(\mathbf{q}) \\ -\nabla \mathcal{U}(\mathbf{q}) &= -\nabla \mathcal{U}_a(\mathbf{q}) - \sum_{i=1}^m \nabla \mathcal{U}_{r,i}(\mathbf{q}) \end{aligned}$$

Depending on the application, the artificial potential field can be used to control the robot either in terms of joint generalized forces τ , joint accelerations $\ddot{\mathbf{q}}$ or joint velocities $\dot{\mathbf{q}}$. However, only the last choice guarantee the asymptotic stability of the system, since in the other two cases the robot may reach the desired goal with a nonzero velocity. To achieve asymptotic stability in such cases, a damping term proportional to the robot velocity is needed. Regardless the choice, it is worth noticing that the artificial potential field produces a reactive feedback control over the robot motion, guiding the robot towards the goal while keeping it away from undesired configurations. In fact, since a control action (vector) is assigned at to each configuration in $\mathbf{q} \in \mathcal{Q}$, then the motion of the robot defines an *integral curve* along the artificial potential field.

The “curse” of local minima One relevant issue with artificial potential fields, an inherent problem of gradient descent algorithms, is that the robot may reach an equilibrium far from the desired goal, due to local balancing between repulsive and attractive fields. Such a local critical point¹³ can be either a saddle point or a minimum (or a maximum in the case of a gradient ascent search). In the former case, considering the structural instability of such critical point, the robot will never reach a stable equilibrium on it. While in the latter case, the robot will get stuck on a stable local equilibrium.

Different techniques for avoiding local minima has been defined for off-line planning: navigation potential functions, heuristic search, wave-front planning, etc. [14, 93, 89]. However, in the case of reactive planning, there’s no way to ensure that the robot will arrive to a desired goal. Actually, this is not a proper limitation, since the idea behind reactive planning is not to replace an off-line planner, but to extend it through fast and efficient control actions when unexpected events occur during the robot motion. Therefore, when a reactive method get stuck on a “local minimum” it implies that the best possible action, given the partial knowledge of the environment and the occurrence of an unexpected event, has been taken. In other words, reactive planning is locally optimal but by no means globally optimal.

¹³A point where the gradient of the artificial potential $\mathcal{U}(\mathbf{q})$ vanishes.

Chapter 3

Semiautonomous teleoperation of redundant mobile manipulators

In the context of robotic teleoperation, it is fundamental to provide an *abstract interaction layer* between the operator and the robot and its surrounding environment, allowing the operator to focus exclusively on the task execution. Not by chance, teleoperation represents a common source of failures [12] in many complex robotics applications. Common issues are the increasing control complexity, specially in the case of mobile manipulators; inadequate or incomplete feedback to the user; low-level motion directives incompatible with constraints or obstacles imposed by the environment. Therefore, such an abstraction layer *must* be able to hide the robot mechanical structure and low-level control mechanisms. However, simple abstract motion rules may degrade the task execution when facing unexpected interactions with the environment. For example, a limited set of motion primitives may lead to unrepeatable Cartesian motions [109, 36], and in the case of collision avoidance may limit the dexterity of the robot. As consequence, the abstract layer between the operator and the robot *should* include methods for adapting and reacting to unexpected environmental conditions without requiring explicit human intervention. Such an autonomous behavior not only improve task execution in terms of accuracy, but also in term of reliability and safety [12]. This abstract interaction layer is commonly known as *operator control Unit* (OCU).

Regardless the underlying application [109, 56, 67, 74, 47, 91, 61], dexterous grasping represents the “life’s” milestone of a mobile manipulator. Beyond the inherent problems associated to the grasping action –that fall beyond the scope of this thesis–, before its execution, a mobile manipulator first needs to move close to the target, then approach the end-effector to the target ensuring the feasibility of the imminent grasp action. On semi-autonomous contexts, the optimal placement

of the end-effector requires human intervention [56, 74]. One way to facilitate this *inspection* instance is to improve the operator control unit, by considering intuitive or natural control interfaces to simplify the task and to improve the situational awareness [60]. In the particular case of vision based feedback, different operator control units has been proposed, many of them inspired from the augmented reality, virtual reality and computer games contexts.

Ögren et al. [73] proposed and implemented an intuitive *free look control* mode for teleoperated unmanned ground vehicles, improving the situational awareness on the surroundings of the robot. The authors exploit the feedback linearization technique to abstract the orientation of the robot chassis, enabling a direct control over orientation and translation of the camera on-board. More recently, Båberg et al. [9] introduced a new concept of *orbit control* mode, which in contrast to the previous approach, implements an object-centered visualization scheme. The authors consider a redundant mobile platform and explored how different constraints can be added to the visualization control mode and introduce a constrained based programming framework for the computation of optimal solutions. Claret et al. [15] implements a similar control mode by coupling the motion of a manipulator together with the motion of quad-rotor carrying an on board camera, i.e., a *free-flying* camera. Such configuration provides a visualization from a *third-person* viewpoint instead of the *first-person* viewpoint considered in the previous approaches. The motion of the two robots is tightly coupled at a kinematics level.

In this work we consider a problem similar the one presented in [9], where a redundant mobile manipulator providing a *first-person* visual feedback to the operator, is required to operate in *orbit control* mode while avoiding possible obstacles and improving the end-effector dexterity during the task execution. The main difference relies in the explicit consideration of the dexterity of the end-effector as an objective function while orbiting, so as to ensure that post-orbit robot configurations allow isotropic motions of the end-effector in workspace. We define the end-effector dexterity in terms of the robot’s manipulability [110, 4, 101], a common measure in the field of robotics teleoperation and control of mobile manipulators. To cite a few, Jung et al. [53] demonstrate how an accurate analysis of manipulability and other quality indexes can be used to improve the design of a versatile humanoid robot for for teleoperated disaster response. In [41] the problem of assisting a human operator in a reach-to-grasp task is addressed, such as to reduce the control effort of the manipulator during the post-grasp task, by means of force cues computed through a task-oriented velocity manipulability cost function. Torabi et al. [98] defined a manipulability measure of a combined

master-slave system for robotic teleoperated minimally invasive surgeries, such a measure is used to assist the design choices of the robotic platform, aimed to improve surgeons control, minimize the footprint of the master robot and avoid singularities and joint limits of the master and slave robots. Bayle et al. [4] generalize the standard definition of manipulability to the case of mobile manipulators with nonholonomic constraints, coupling base locomotion and arm movements for manipulability maximization during the task execution. Vahrenkamp et al. [101] extended the concept of manipulability ellipsoid including constraints coming from joint limits, workspace obstacles or self-distance into its definition, and used such geometrical object to build a manipulability distribution over the workspace, allowing to capture the robot capabilities in terms of operational freedom during task planning and execution. As in [4], we use the manipulability maximization as coupling mechanism between the end-effector motion and the base locomotion.

This work introduces three fundamental contributions respect the state-of-the-art. First, as previously described, includes the dexterity of the manipulator to couple the manipulator motion and the base locomotion during the execution of the *orbit control* tasks. Second, we introduce a purely reactive control scheme based on the *remote center of motion constraint* (RCM) for satisfying the orbit constraints. This kinematics constraint is widely known in the field of minimally invasive surgery (MIS) [18, 1, 81, 82], where all surgical tools are constrained to pass through and to rotate around small incisions in the patient’s body (see Section 3.2.1). Third, we introduce a prioritized control scheme [88, 37] that exploits the robot’s redundancy to satisfy all the imposed constraints. We follow a reactive control strategy for local redundancy resolution [23, 84, 25, 72, 36], without requiring optimization strategies nor motion planning. Nonetheless, our approach can also be effectively used to track a reference trajectory provided that the platform is holonomic.

The rest of the chapter is organized as follows. Section 3.1 formally introduces the problem under study. Section 3.2 describes the proposed approach for the solution, including the RCM constraint and its application to the *orbit control* mode. Section 3.3 discusses the experimental evaluation. Finally, the concluding remarks together with a discussion regarding some limitations and possible extensions of the approach for future work are found in Section 3.4.

3.1 Problem statement

Suppose that a mobile manipulator is modeled by a n -dimensional vector \mathbf{q} of generalized coordinates defined inside a configuration space $\mathcal{Q} \subset \mathbb{R}^n$. Any configuration \mathbf{q} is given as the concatenation of the joint variables \mathbf{q}_a of the arm together with the generalized coordinates \mathbf{q}_b of the mobile base

$$\mathbf{q} \triangleq \begin{pmatrix} \mathbf{q}_a \\ \mathbf{q}_b \end{pmatrix}, \quad n_a \triangleq \dim(\mathbf{q}_a), \quad n_b \triangleq \dim(\mathbf{q}_b), \quad n = n_a + n_b.$$

The end-effector of the robot can reach any pose \mathbf{r}_e in the workspace $\mathcal{W} \subset \mathbb{R}^m$ and can be computed through the direct kinematics map $\mathbf{q} \mapsto f(\mathbf{q})$, $f: \mathcal{Q} \rightarrow \mathcal{W}$. Any pose in \mathcal{W} is defined by a Cartesian position \mathbf{p} and an orientation vector \mathbf{o} , defined in terms of any orientation representation (Euler angles, quaternions, etc.),

$$\mathbf{r} \triangleq \begin{pmatrix} \mathbf{p} \\ \mathbf{o} \end{pmatrix}, \quad m_p \triangleq \dim(\mathbf{p}), \quad m_o \triangleq \dim(\mathbf{o}), \quad m = m_p + m_o.$$

It is assumed that the robot is localized with respect to some *world* reference frame and that all geometrical quantities considered here and in following sections are defined with respect to this global reference frame. Explicit time dependence will be omitted for better readability of the chapter.

Denoting with \mathbf{e}_z the z -axis of the end-effector's reference frame and assuming that the roll axis of the end-effector coincides with \mathbf{e}_z , given a target pose $\mathbf{r}_t \in \mathcal{W}$ and an orbit distance ρ , we define the *orbit manifold* centered at the target pose \mathbf{p}_t as

$$\mathcal{O} = \left\{ \mathbf{r} = \begin{pmatrix} \mathbf{p} \\ \mathbf{o} \end{pmatrix} \in \mathcal{W} \mid \|\mathbf{p}_t - \mathbf{p}\|^2 = \rho^2, \quad \mathbf{e}_z \times (\mathbf{p}_t - \mathbf{p}) = 0, \quad \mathbf{e}_z^\top (\mathbf{p}_t - \mathbf{p}) > 0 \right\}. \quad (3.1)$$

Now, given a set of velocity commands $\dot{\mathbf{r}}_o \in \mathcal{T}_{\mathbf{r}}\mathcal{W}^1$ imposed by the operator through the OCU interface, we introduce the following problem

Problem 1. *Determine a control law such that the pose of the end-effector \mathbf{r} of the mobile manipulator reaches the orbit manifold \mathcal{O} and keeps sliding (or orbiting) on it, mapping the operation intentions into geodesic motions along the manifold, i.e., such that $\dot{\mathbf{r}}_o \mapsto \dot{\mathbf{r}}$, $\dot{\mathbf{r}} \in \mathcal{T}_{\mathbf{r}}\mathcal{O}$, $\forall t > t_0$, with $t_0 \in \mathbb{R}$ representing the time instant when the end-effector reaches the orbit manifold \mathcal{O} , if not already sliding on it.*

Moreover, assuming that the mobile manipulator is kinematically redundant

¹ $\mathcal{T}_p M$ denotes the *tangent space* of the manifold M at p .

for the orbiting task, i.e., $n > m_p^2$, we have the following *extended* problem

Problem 2. *Determine a control law able to solve Problem 1, while maximizing the dexterity of the end-effector and avoiding any Cartesian obstacles during the motion.*

It is worth noticing that, in general, Problem 2 may not have a solution, since the existence of a solution highly depends on the obstacles location and the mechanical structure of the robot. Moreover, without a proper planning action there's no way to ensure the completeness of an incremental reactive solution. However, considering that in many practical applications the robot safety has a greater priority than the task execution, we considered the following modified version of Problem 2, where the orbit constraints is subject to the feasibility in terms of collision avoidance

Problem 3. *Determine a safe control law that provides, in the presence of workspace obstacles, a feasible solution –if exist– to Problem 1 that also maximizes the dexterity of the end-effector during the motion.*

3.2 Proposed approach

As previously stated, we propose a reactive control approach to follow geodesic motions in the orbiting manifold \mathcal{O} . We follow a similar approach to Pepe et al. [74], where the motions of a robot manipulator and a free-flying camera are coupled together. However, we take the inspiration from [4], where the manipulability maximization is used as a coupling mechanism between the end-effector motion and the base locomotion. In contrast to [9] we introduce an analytical representation of the orbit constraint through the concept of remote center of motion (RCM) [1]. [56] propose a series of control modalities to accomplish complex task in disaster scenarios. Here we only consider two specific tasks, namely the approaching task toward the orbiting manifold and the geodesic motion along the manifold, and provide a single control mechanism able to handle both tasks.

3.2.1 RCM constraint in *orbit mode* control

The remote center of motion is defined by a fixed point $\mathbf{p}_{rcm} \in \mathbb{R}^{n_p}$, contained in \mathcal{W} , around which part of the kinematic chain of the robot is constrained to rotate. This

²The orbiting constraint imposes $m_p - 1$ linearly independent constraints on the end-effector motion. For example if $\mathbf{p} \in \mathbb{R}^3$, then the dimension of the orbiting constrain equals 2.

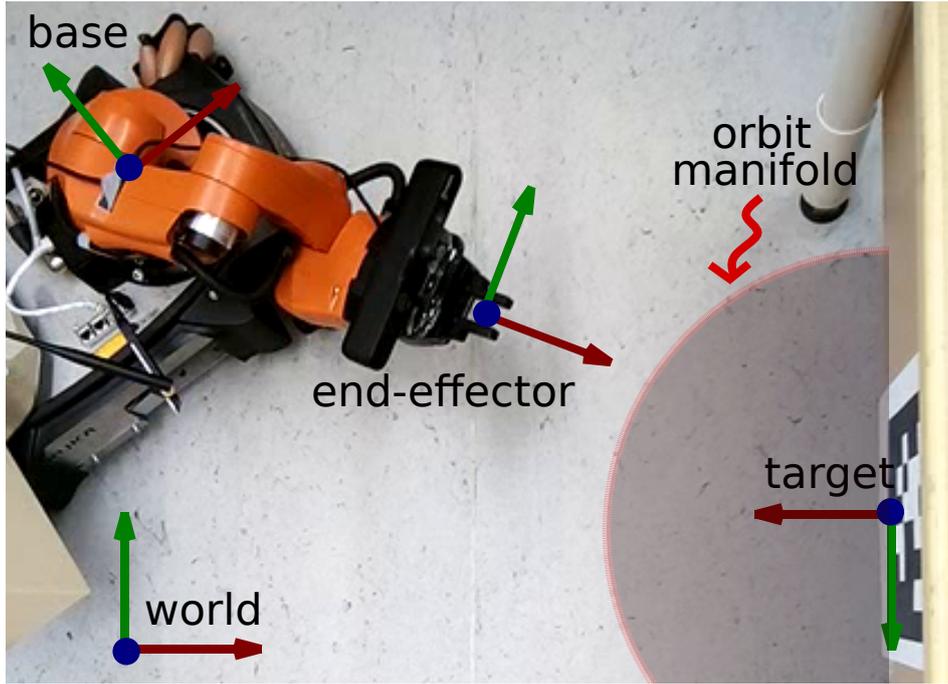


Figure 3.1: Qualitative top view representation of the problem under study. There are four fundamental reference frames involved in the problem definition (*world*, *base*, *end-effector* and *target*). In the problem statement and following discussions it is assumed that all frame orientations and workspace poses are referred to with respect to the *world* frame. The orbit manifold correspond to a spherical surface centered at the target location. As discussed in Section 3.2, the proposed controller not only maps the operator intentions in geodesic motions along the orbit manifold, but also guarantee the asymptotic stability while approaching it.

kinematics constraint is widely known in the field of minimally invasive surgery (MIS) [18, 1, 81, 82], where all surgical tools are constrained to pass through and to rotate around small incisions in the patient’s body.

Let’s denote with $\mathbf{p}_i \in \mathbb{R}^{n_p}$ the current position of the each link i of the arm of the robot, with $i=1, \dots, n_a$, and let’s assume that the remote center of motion constraint \mathbf{p}_{rcm} has been imposed to link i of the chain, that is, to lie on the line segment between \mathbf{p}_i and \mathbf{p}_{i+1}

$$\mathbf{p}_{rcm} = \mathbf{p}_i + \tilde{\alpha}(\mathbf{p}_{i+1} - \mathbf{p}_i)$$

with $0 < \tilde{\alpha} < 1$ indicating the relative distance of \mathbf{p}_{rcm} from \mathbf{p}_i . The RCM constraint imposes that

$$\dot{\mathbf{p}}_{rcm} \equiv 0 \implies \dot{\mathbf{p}}_{rcm} = \left(\mathbf{J}_i + \tilde{\alpha}(\mathbf{J}_{i+1} - \mathbf{J}_i) \right) \begin{pmatrix} \dot{\mathbf{q}} \\ \dot{\tilde{\alpha}} \end{pmatrix} = \mathbf{J}_{rcm} \begin{pmatrix} \dot{\mathbf{q}} \\ \dot{\tilde{\alpha}} \end{pmatrix} = 0$$

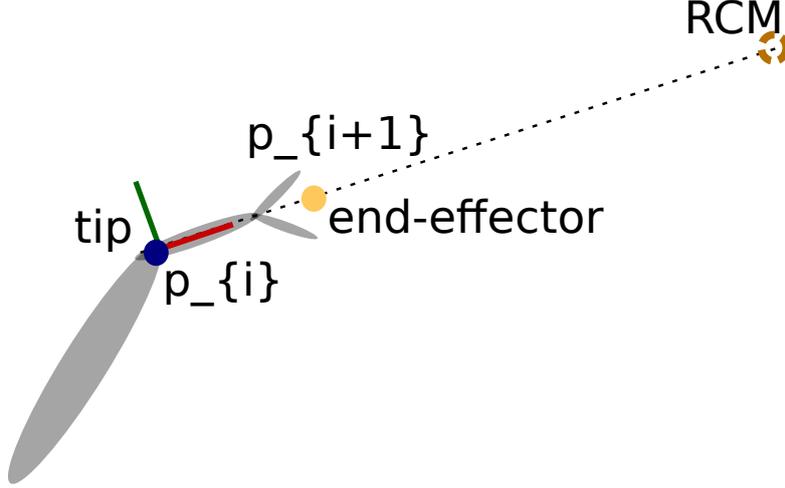


Figure 3.2: RCM definition for orbit control mode.

where J_i represents the Jacobian matrix of \mathbf{p}_i . Let's suppose now, that the RCM is not between links i and $i+1$ but along the line joining them, at some distance ρ from the link $i+1$. In such a case

$$\tilde{\alpha} = 1 + \frac{\rho}{\|\mathbf{p}_{i+1} - \mathbf{p}_i\|} \implies \mathbf{p}_{rcm} = \mathbf{p}_{i+1} + \underbrace{\frac{\rho}{\|\mathbf{p}_{i+1} - \mathbf{p}_i\|}}_{\alpha} (\mathbf{p}_{i+1} - \mathbf{p}_i) = \mathbf{p}_{i+1} + \alpha (\mathbf{p}_{i+1} - \mathbf{p}_i)$$

which implies that

$$\dot{\mathbf{p}}_{rcm} = \left(J_{i+1} + \alpha (J_{i+1} - J_i) \quad \mathbf{p}_{i+1} - \mathbf{p}_i \right) \begin{pmatrix} \dot{\mathbf{q}} \\ \dot{\alpha} \end{pmatrix} = J_{rcm} \begin{pmatrix} \dot{\mathbf{q}} \\ \dot{\alpha} \end{pmatrix} = 0. \quad (3.2)$$

Letting \mathbf{p}_{i+1} coincide with the position of the end-effector and ρ the desired orbit distance from the target \mathbf{p}_t (see Figure 3.2), then by imposing that $\mathbf{p}_{rcm} \triangleq \mathbf{p}_t$ the RCM constraint (3.2) equals the orbiting constraint (3.1). In particular, the parameters $\dot{\alpha}$ allows to update the orbiting distance, as described in Section 3.2.3.

3.2.2 Mapping operator intentions

This is a simple task and consist of the orthogonal projection of the commanded velocity $\dot{\mathbf{r}}_o \in \mathcal{T}_{\mathbf{r}}\mathcal{Q}$ into the tangent space of the orbit manifold \mathcal{O} . Suppose that the end-effector of the robot has already reached the orbit manifold, i.e., $\mathbf{r} \in \mathcal{O}$. Then, by the RCM constraint (3.2), or equivalently, the orbiting constraint (3.1), only the Cartesian motions orthogonal to the sagittal axis of the end effector are allowed. Moreover, only rotations about the same axis (roll) are allowed. The resulting constrained velocities will be denoted as $\dot{\hat{\mathbf{r}}}_o \in \mathcal{O}$.

3.2.3 Reactive redundancy resolution

For the given operator commands $\dot{\hat{\mathbf{r}}}_o$, subject to the orbit constraint (3.1) we can define the orbit error as

$$\dot{\mathbf{e}}_{\mathcal{O}} = \begin{pmatrix} \dot{\hat{\mathbf{r}}}_o \\ \dot{\hat{\mathbf{p}}}_t \end{pmatrix} - \begin{pmatrix} \mathbf{J} & \mathbf{0} \\ \mathbf{J}_{rcm} & \end{pmatrix} \begin{pmatrix} \dot{\mathbf{q}} \\ \dot{\alpha} \end{pmatrix} = \dot{\mathbf{r}}_{\mathcal{O}} - \mathbf{J}_{\mathcal{O}} \begin{pmatrix} \dot{\mathbf{q}} \\ \dot{\alpha} \end{pmatrix} \quad (3.3)$$

which correspond to a task augmentation control scheme. Since we're dealing with a redundant manipulator, we consider the generalized least norm inversion of the task Jacobian [23, 84, 25, 72, 36] using a proportional derivative (PD) control action. we consider the following control law

$$\begin{pmatrix} \dot{\mathbf{q}}_{\mathcal{O}}^* \\ \dot{\alpha}^* \end{pmatrix} = \mathbf{J}_{\mathcal{O}}^\dagger (\dot{\mathbf{r}}_{\mathcal{O}} + \mathbf{K} \mathbf{e}_{\mathcal{O}}) + (\mathbf{I} - \mathbf{J}_{\mathcal{O}}^\dagger \mathbf{J}_{\mathcal{O}}) \begin{pmatrix} \dot{\mathbf{q}}_0 \\ \dot{\alpha}_0 \end{pmatrix} \quad (3.4)$$

with \mathbf{K} some constant symmetric positive definite matrix. This control law ensures that the error asymptotically converges to zero. In particular, we implemented the iterative algorithm of Flacco et al. [36] for optimal task scaling based on velocity bounds. Near singularities we consider the damped least squares inversion [63, 84]. Self-motions $\dot{\mathbf{q}}_0$ and $\dot{\alpha}$ can be chosen independently of each other. In particular, it is possible to modulate the radius of the orbit manifold \mathcal{O} by means of the following quadratic cost function

$$H(\alpha) = \frac{1}{2}(\alpha - \bar{\alpha}(\rho))^2 \quad \implies \quad \dot{\alpha}_0 = -\eta_\rho \left(\frac{\partial H(\alpha)}{\partial \alpha} \right)^\top = \eta_\rho (\bar{\alpha}(\rho) - \alpha)$$

where $\bar{\alpha}(\rho)$ represents the desired relative distance and $\eta_\rho > 0$.

3.2.4 Manipulability maximization

It has been proved that the eccentricity of the velocity manipulability ellipsoid provide a measure of the end-effector ability to isotropically move along all directions of $\mathcal{T}_r \mathcal{W}$ [29, 4, 101]. A global manipulability measure [110] is defined as

$$w(\mathbf{q}) = \sqrt{\det(\mathbf{J}\mathbf{J}^\top)}$$

that is, as the volume of the manipulability ellipsoid. As such, this quantity is always positive except in singular configurations when it becomes zero, thus it can be adopted as a distance measure of a given configuration \mathbf{q} from a kinematic singularity. It is possible to maximize the robot manipulability during the task

execution by computing $\dot{\mathbf{q}}_0$ in (3.4) as the gradient of the manipulability measure. In particular, it is computationally more efficient to consider the following cost function

$$H_0(\mathbf{q})=w^2(\mathbf{q}).$$

Then, the manipulability maximizing self-motion results

$$\dot{\mathbf{q}}_{0,i} = \eta_m \left(\frac{\partial}{\partial \mathbf{q}_i} H_0(\mathbf{q}) \right)^\top = \eta_m \det(\mathbf{J}\mathbf{J}^\top) \operatorname{tr} \left((\mathbf{J}\mathbf{J}^\top)^{-1} \left(\frac{\partial \mathbf{J}}{\partial \mathbf{q}_i} \mathbf{J}^\top + \mathbf{J} \frac{\partial \mathbf{J}^\top}{\partial \mathbf{q}_i} \right) \right)$$

for $i=1, \dots, n$, where $\eta_m > 0$.

3.2.5 Proposed control scheme

As stated in Section 3.1, we give the maximum priority to collision avoidance. To keep the robot away from a given obstacle in the surroundings of the robot we define a repulsive potential field of the form

$$\mathcal{U}(\mathbf{q}) = \begin{cases} \frac{k_3}{r} \left(\frac{1}{d(\mathbf{q})} - \frac{1}{d_0} \right)^r & \text{if } d(\mathbf{q}) \leq d_0 \\ 0 & \text{otherwise} \end{cases}$$

$$-\nabla \mathcal{U}(\mathbf{q}) = \begin{cases} \frac{k_3}{d^2(\mathbf{q})} \left(\frac{1}{d(\mathbf{q})} - \frac{1}{d_0} \right)^{r-1} \nabla d(\mathbf{q}) & \text{if } d(\mathbf{q}) \leq d_0 \\ 0 & \text{otherwise} \end{cases}$$

where k_3 is a positive constant, d_0 the range of influence of the field, $r=1,2,\dots$ and $d(\mathbf{q})$ the clearness between the robot and the obstacle. This potential tends to infinity when the robot approaches to the obstacle, with a growth rate modulated by the exponent r . Beyond the range of influence, the potential field is zero. In the case of l obstacles, we have a total repulsive potential field of the form

$$\mathcal{U}(\mathbf{q}) = \sum_{i=1}^l \mathcal{U}_i(\mathbf{q}).$$

We introduce a prioritized scheme [88, 37] to map lower priority tasks constraints in the null space of collision avoidance task. Assuming that \mathbf{J}_c represents the Jacobian matrix associated to the collision avoidance constraints, we can define

the optimal solution as

$$\dot{\mathbf{q}}_{\mathcal{C}}^* = -\mathbf{J}_{\mathcal{C}}^\dagger \nabla \mathcal{U}(\mathbf{q}) + (\mathbf{I} - \mathbf{J}_{\mathcal{C}}^\dagger \mathbf{J}_{\mathcal{C}}) \dot{\mathbf{q}}_{\mathcal{O}}$$

now we can choose $\dot{\mathbf{q}}_{\mathcal{O}}^*$ so as to satisfy the orbit task (3.4). That is, determine the optimal solution of $\dot{\mathbf{q}}_{\mathcal{O}}^*$ subject to the task constraint

$$\begin{aligned} \dot{\mathbf{r}}_{\mathcal{O}} - \mathbf{J}_{\mathcal{O}} \begin{pmatrix} \dot{\mathbf{q}}_{\mathcal{C}}^* \\ 0 \end{pmatrix} &= \dot{\mathbf{r}}_{\mathcal{O}} - \mathbf{J}_{\mathcal{O}} \begin{pmatrix} -\mathbf{J}_{\mathcal{C}}^\dagger \nabla \mathcal{U}(\mathbf{q}) + (\mathbf{I} - \mathbf{J}_{\mathcal{C}}^\dagger \mathbf{J}_{\mathcal{C}}) \dot{\mathbf{q}}_{\mathcal{O}} \\ \dot{\mathbf{a}}^* \end{pmatrix} \\ &= \dot{\mathbf{r}}_{\mathcal{O}} + \mathbf{J}_{\mathcal{O}} \begin{pmatrix} \mathbf{J}_{\mathcal{C}}^\dagger \nabla \mathcal{U}(\mathbf{q}) \\ 0 \end{pmatrix} - \mathbf{J}_{\mathcal{O}} \begin{pmatrix} (\mathbf{I} - \mathbf{J}_{\mathcal{C}}^\dagger \mathbf{J}_{\mathcal{C}}) \dot{\mathbf{q}}_{\mathcal{O}} \\ \dot{\mathbf{a}}^* \end{pmatrix} \\ &= \underbrace{\dot{\mathbf{r}}_{\mathcal{O}} + \mathbf{J}_{\mathcal{O}} \begin{pmatrix} \mathbf{J}_{\mathcal{C}}^\dagger \nabla \mathcal{U}(\mathbf{q}) \\ 0 \end{pmatrix}}_{\dot{\tilde{\mathbf{r}}}_{\mathcal{O}}} - \underbrace{\mathbf{J}_{\mathcal{O}} \begin{pmatrix} \mathbf{I} - \mathbf{J}_{\mathcal{C}}^\dagger \mathbf{J}_{\mathcal{C}} & 0 \\ 0 & 0 \end{pmatrix}}_{\tilde{\mathbf{J}}_{\mathcal{O}}} \begin{pmatrix} \dot{\mathbf{q}}_{\mathcal{O}} \\ \dot{\mathbf{a}}^* \end{pmatrix} \\ &= \dot{\tilde{\mathbf{r}}}_{\mathcal{O}} - \tilde{\mathbf{J}}_{\mathcal{O}} \begin{pmatrix} \dot{\mathbf{q}}_{\mathcal{O}} \\ \dot{\mathbf{a}}^* \end{pmatrix} \end{aligned}$$

where $\dot{\tilde{\mathbf{r}}}_{\mathcal{O}}$ accounts for the contribution induced by the solution of the priority task $\mathbf{r}_{\mathcal{C}}$ over $\mathbf{r}_{\mathcal{O}}$ and $\tilde{\mathbf{J}}_{\mathcal{O}}$ correspond to the projection of $\mathbf{J}_{\mathcal{O}}$ into the null space $\mathcal{N}(\mathbf{J}_{\mathcal{C}})$. Therefore, only the rows of $\mathbf{J}_{\mathcal{O}}$ belonging to $\mathcal{N}(\mathbf{J}_{\mathcal{C}})$ will be employed in the next optimization step. The generalized (possibly weighted) least norm solution of $\dot{\mathbf{q}}_{\mathcal{O}}^*$ is thus given by

$$\begin{pmatrix} \dot{\mathbf{q}}_{\mathcal{O}}^* \\ \dot{\mathbf{a}}^* \end{pmatrix} = \tilde{\mathbf{J}}_{\mathcal{O}}^\dagger (\dot{\tilde{\mathbf{r}}}_{\mathcal{O}} + \mathbf{K} \mathbf{e}_{\mathcal{O}}) + (\mathbf{I} - \tilde{\mathbf{J}}_{\mathcal{O}}^\dagger \tilde{\mathbf{J}}_{\mathcal{O}}) \begin{pmatrix} \dot{\mathbf{q}}_0 \\ \dot{\mathbf{a}}_0 \end{pmatrix} \quad (3.5)$$

3.3 Experimental evaluation

To assess the validity of the approach, we used the KUKA youBot [6], a mobile manipulator composed by a holonomic platform with 3-DoF and a planar manipulator with 5-DoF and a soft two-finger gripper. The localization was based on AprilTags2 [105], used to localize the robot with respect to the fixed target. The visual sensor of the robot consisted of the (calibrated) RGB camera of the depth-sensing Asus Xtion. The OCU consisted on a minimal graphical interface for the visualization of the feedback images and three control bars to define the desired robot pose directly in spherical coordinates with respect to the target. As a consequence, the operator commands are automatically compatible with the

orbit constraints. Such control commands are defined by the position vector (in workspace) relative to the target frame

$$\mathbf{p}_o = \begin{pmatrix} \lambda \\ \phi \\ \rho \end{pmatrix} \quad (3.6)$$

with λ defining the longitude in degrees, ϕ the latitude in degrees and ρ the orbit radius in meters. The orientation of the end effector is defined by the sagittal axis pointing toward the RCM, and the up vector was constrained to lie on the plane defined by the sagittal axis of the end effector and the world up vector. This choice allows a more natural visualization of the object during the orbit. For this reason, the fourth joint of the manipulator (rolling angle of the wrist) was not actuated during the experiments to ensure the desired fixed orientation of the up-vector.

Since the arm of the youBot corresponds to a planar manipulator, we can split the orbit task in two independent tasks along the sagittal and transverse plane of the end-effector frame. The advantage of this splitting is that the maximization of the manipulability is performed exclusively on the sagittal plane coupling the motion between the shoulder, elbow and wrist joints of the arm together with the linear contribution of the base along the sagittal axis of the end-effector (see Figure 3.1), while along the transverse plane is possible to maximize secondary goals in terms of the relative orientation between the base and the end-effector. In the experiments, as a secondary task on the traverse plane, we considered the alignment between the sagittal axis of the base with the sagittal axis of the end-effector. Together with the manipulability maximization, a self-collision cost was added between the end-effector and the shoulder, to avoid that the robot completely retracts itself for maximum dexterity along the sagittal direction.

One practical issue that we faced in the experimental evaluation was network delays, causing unwanted oscillations of the robot around the desired pose. To alleviate this issue we substitute the matrix \mathbf{K} in (3.4) for a monotonic increasing function of the norm of the error $\mathbf{K}(\mathbf{e})$, with a rapid decay to zero when the norm of the error falls below a threshold, allowing to reducing the norm the control action close to the steady state configurations. The asymptotically stability of the system can be readily demonstrated through the Lyapunov stability analysis. Moreover, a weighted Jacobian inversion was required to make the robot move with equally distributed velocity contributions. In particular, Cartesian velocities of the platform, the angular velocity of the platform and the angular velocities of the joints required one different weight for each group.

The experiments were designed so as to measure the performance of the controller with respect to each input parameter in (3.6), by considering independently and joint variations. In particular, Figures 3.3 and 3.4 present the outcomes for a planar orbit along the transversal plane of the end-effector frame, with the orbit radius fixed. As expected, the norm of errors exponentially decay to zero, likewise to the joint velocities and the base twist. Also, the manipulability measure has remained constant during the task executions since there are no control actions performed along sagittal plane of the end-effector frame. This can be observed in Figure 3.13, together with the mobile platform reconfiguration due to the imposed alignment constraint. Figure 3.6 present the outcomes for a planar approach task along the transversal plane of the end-effector frame. Also in this case the norm of the errors exponentially decay to zero, likewise to the joint velocities and the base twist. As can be observed the manipulability rapidly decreases when the robot is near a singular configuration completely stretched to satisfy the task constraints.

Figure 3.5 presents the outcomes for a planar orbit along the sagittal plane of the end-effector frame. Again, the norm of the errors exponentially decay to zero, likewise to the joint velocities and the base twist. In this case it is possible to observe how the maximization of the manipulability measure becomes inconsistent with the orbit task when the robot configuration has a negative latitude angle. When the inconsistency ceases, the maximization of the manipulability allow to produce a more dexterous pose on the end-effector (see Figure 3.12). Figure 3.7 present the outcomes for a planar approach task along the sagittal plane of the end-effector frame. Also in this case the norm of the errors exponentially decay to zero, likewise to the joint velocities and the base twist. Like before, the manipulability rapidly decreases when the robot is near a singular configuration completely stretched to satisfy the task constraints. A similar situation is depicted in Figure 3.11, where the robot executes an approach task along the sagittal plane.

Similar conclusions can be obtained by analyzing the outcomes of the geodesic orbit represented in Figure 3.8 and the cyclic motion defined in Figure 3.9. Finally, Figure 3.10 shows the manipulability maximization and RCM error trends when varying the self-collision threshold. We observe that the RCM error is kept bounded during the whole self-motion, exception made for the range subject to very high levels of noise. The manipulability measure monotonically decreases while the self-collision avoidance threshold (tip-shoulder distance) is increased; in particular, it is observed that although a local manipulability maximum is attained during reconfiguration, it is quickly lost by the effect of the self-collision avoidance constraint.

3.4 Conclusions and future work

In this work we described a reactive control scheme for the semi-autonomous teleoperation of mobile manipulator operating in *orbit control* mode. Semi-autonomous teleoperation is required to reduce the cognitive load of the operator and to improve task execution time, accuracy, reliability and safety [12]. Different control strategies has been proposed for intuitive teleoperation inspired from computer games interfaces Ögren et al. [73], Båberg et al. [9], Claret et al. [15]. Following a similar inspiration, the proposed approach introduces the concept of *remote center of motion* in the *orbit control* mode and demonstrates that both constraints are equivalent. Such equivalence can be exploited to introduce a prioritized control strategy that, in the case of kinematics redundancy, can introduce different subsidiary goals during the task execution. The validity of the approach has been tested on a real robotics platform, and the results are promising.

A practical limitation of the proposed approach lies in its reactive nature. In the case of unstructured environments a planning strategy is required. However, the control scheme can be applied out-of-the-box for trajectory tracking or posture regulation in the case of holonomic mobile platforms.

Moreover, in the experimental evaluation we had faced problems with network delays and noise disturbances. Network delays were mitigated by relaxing the convergence rate near the equilibrium. Noise was rejected by the controller but spurious transients degraded the quality of the control. Such noisy transients are associated with the eye-in-hand configuration used for the robot localization and the network delays during the image frames transmission. It is possible to obtain finite time convergence when approaching the orbit surface, even in the presence of network delays or spurious disturbances, using nonlinear robust approaches like the sliding mode control [34, 76]. This nonlinear control approach seems promising since the orbit manifold represents a *sliding surface*.

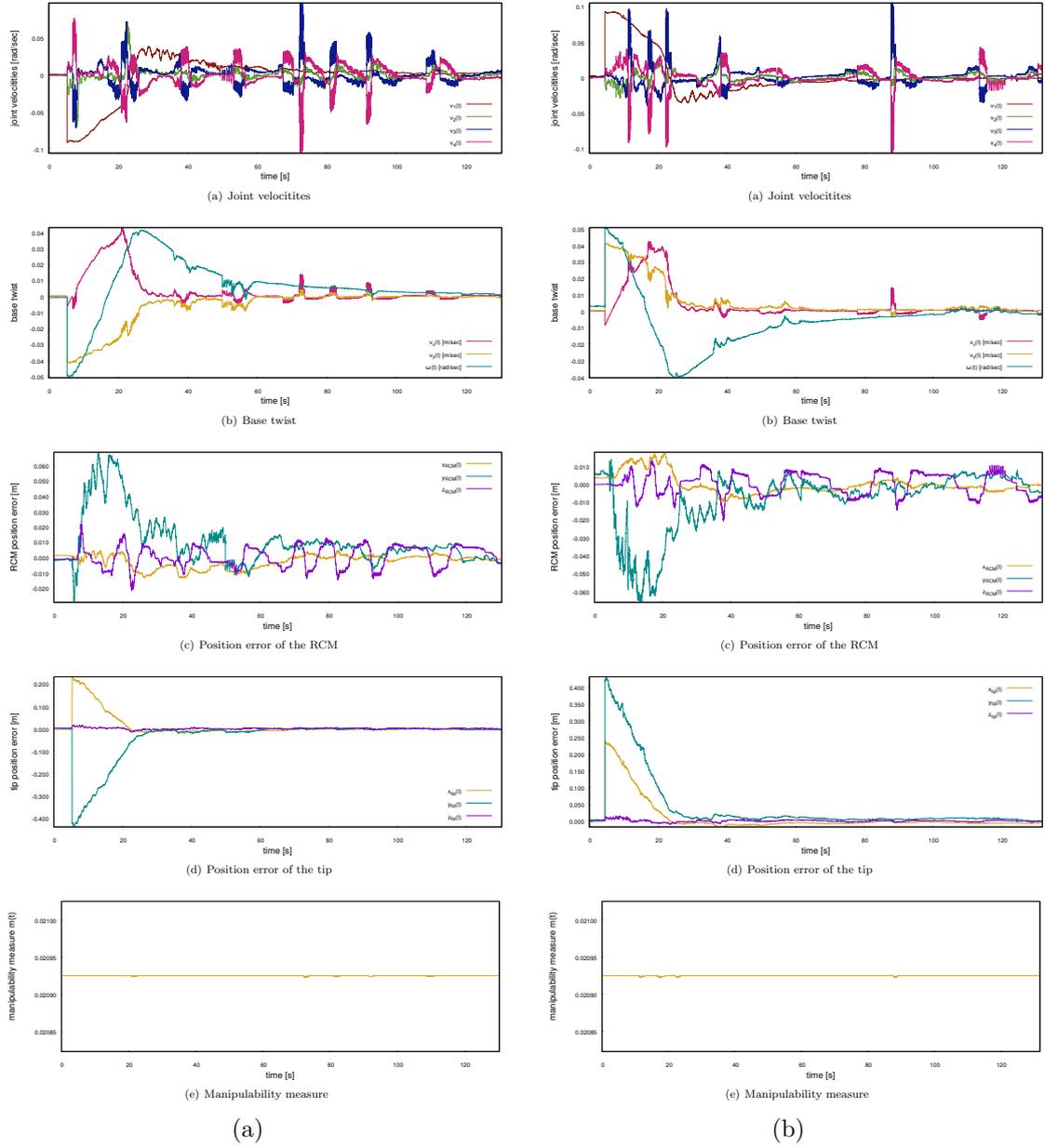


Figure 3.3: Execution of a planar orbit (parallel to the transverse plane), with $-30.0^\circ \leq \lambda \leq 30.0^\circ$, $\phi = -12.0^\circ$ and $\rho = 0.5m$. The column (a) presents the outcomes when varying λ from -30.0° to 30.0° , while the column (b) the outcomes of the backward motion (varying λ from 30.0° to -30.0°). The first and second row presents the control commands (joint velocities and base twist, respectively); third and fourth rows, respectively, the RCM and tip errors; the fifth and last row the value of the manipulability measure.

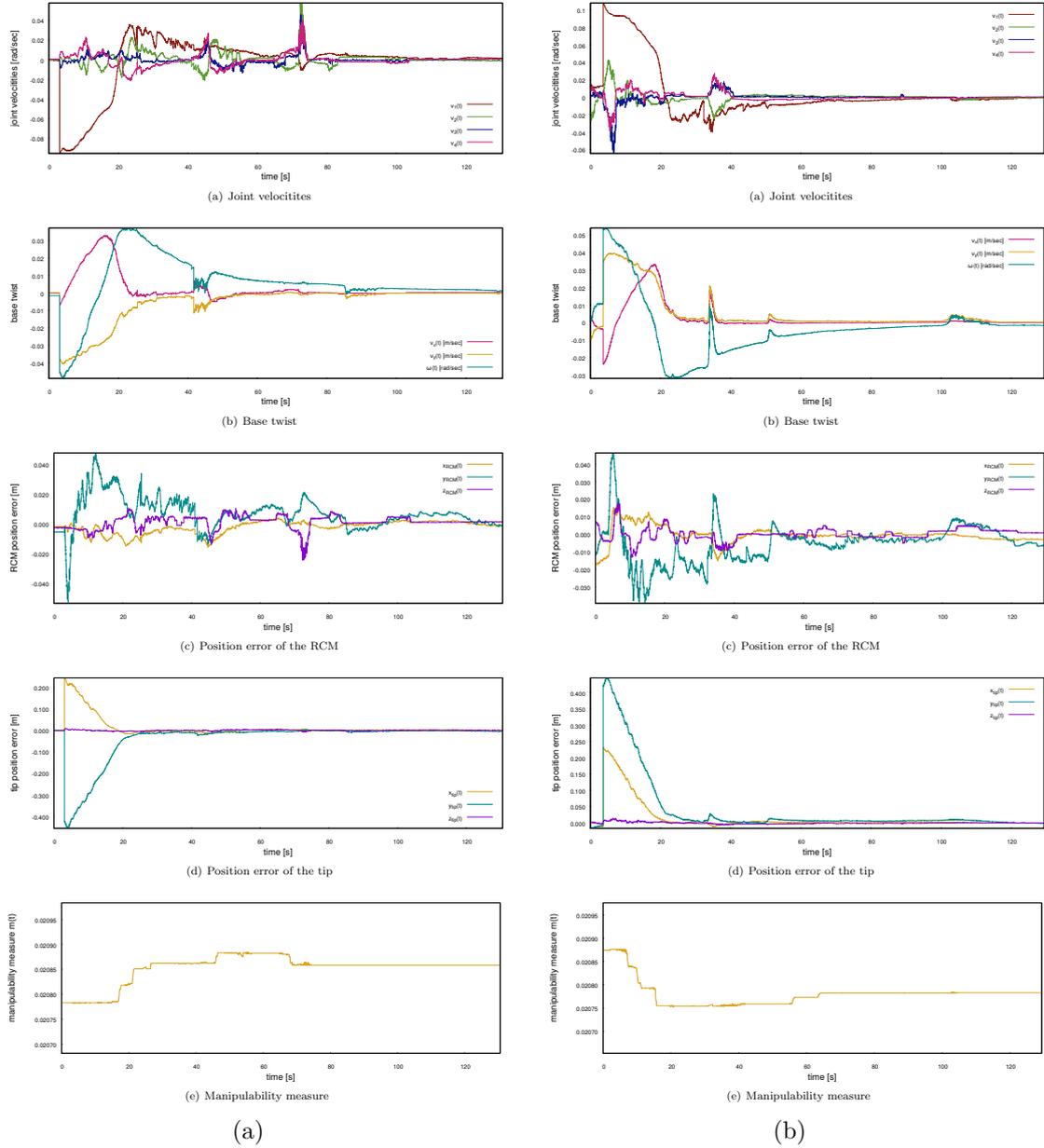


Figure 3.4: Execution of a planar orbit (parallel to the transverse plane), with $-30.0^\circ \leq \lambda \leq 30.0^\circ$, $\phi = 12.0^\circ$ and $\rho = 0.5m$. The column (a) presents the outcomes when varying λ from -30.0° to 30.0° , while the column (b) the outcomes of the backward motion (varying λ from 30.0° to -30.0°). The first and second row presents the control commands (joint velocities and base twist, respectively); third and fourth rows, respectively, the RCM and tip errors; the fifth and last row the value of the manipulability measure.

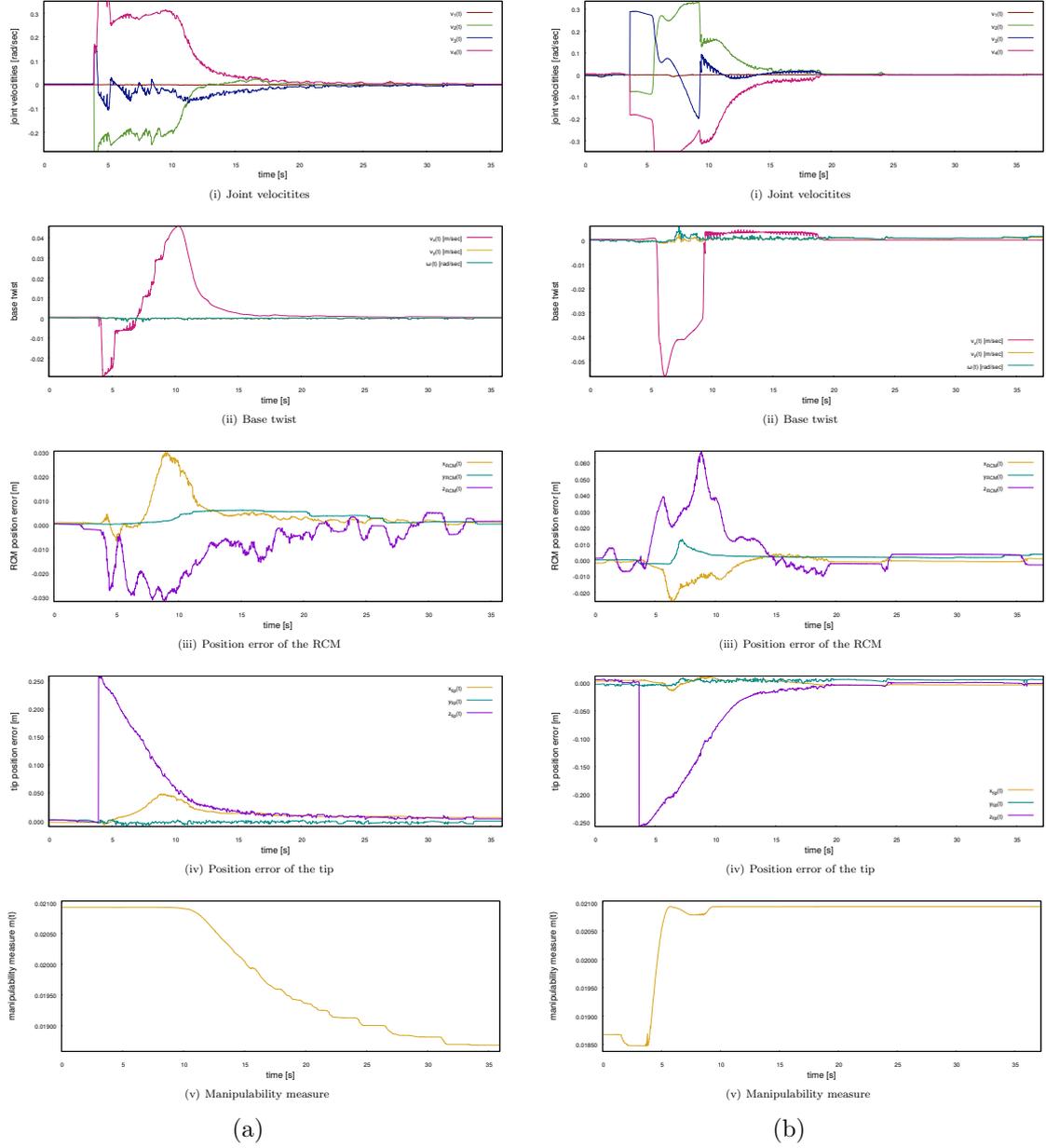


Figure 3.5: Execution of a planar orbit (parallel to the sagittal plane), with $\lambda=0.0^\circ$, $-15.0^\circ \leq \phi \leq 15.0^\circ$ and $\rho=0.5m$. The column (a) presents the outcomes when varying ϕ from -15.0° to 15.0° , while the column (b) the outcomes of the backward motion (varying ϕ from 15.0° to -15.0°). The first and second row presents the control commands (joint velocities and base twist, respectively); third and fourth rows, respectively, the RCM and tip errors; the fifth and last row the value of the manipulability measure.

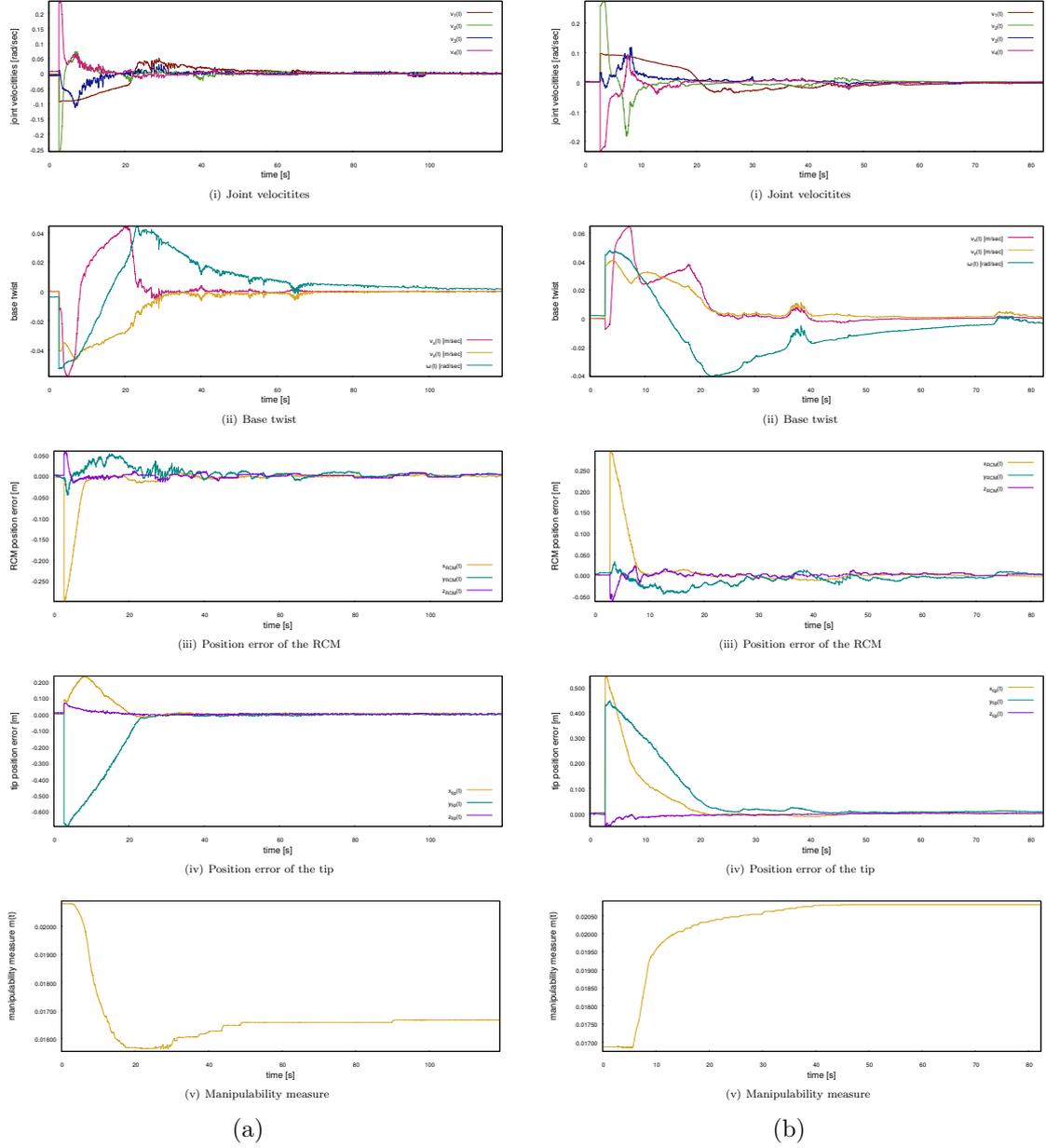


Figure 3.6: Execution of a planar approach task (parallel to the transverse plane), with $-30.0^\circ \leq \lambda \leq 30.0^\circ$, $|\phi| = 12.0^\circ$ and $0.5m \leq \rho \leq 0.8m$. Column (a) presents the outcomes when varying (λ, ρ) from $(-30.0^\circ, 0.5m)$ to $(30.0^\circ, 0.8m)$ with $\phi = 12.0^\circ$. Column (b) presents the outcomes when varying (λ, ρ) from $(30.0^\circ, 0.8m)$ to $(-30.0^\circ, 0.5m)$ with $\phi = -12.0^\circ$. The first and second row presents the control commands (joint velocities and base twist, respectively); third and fourth rows, respectively, the RCM and tip errors; the fifth and last row the value of the manipulability measure.

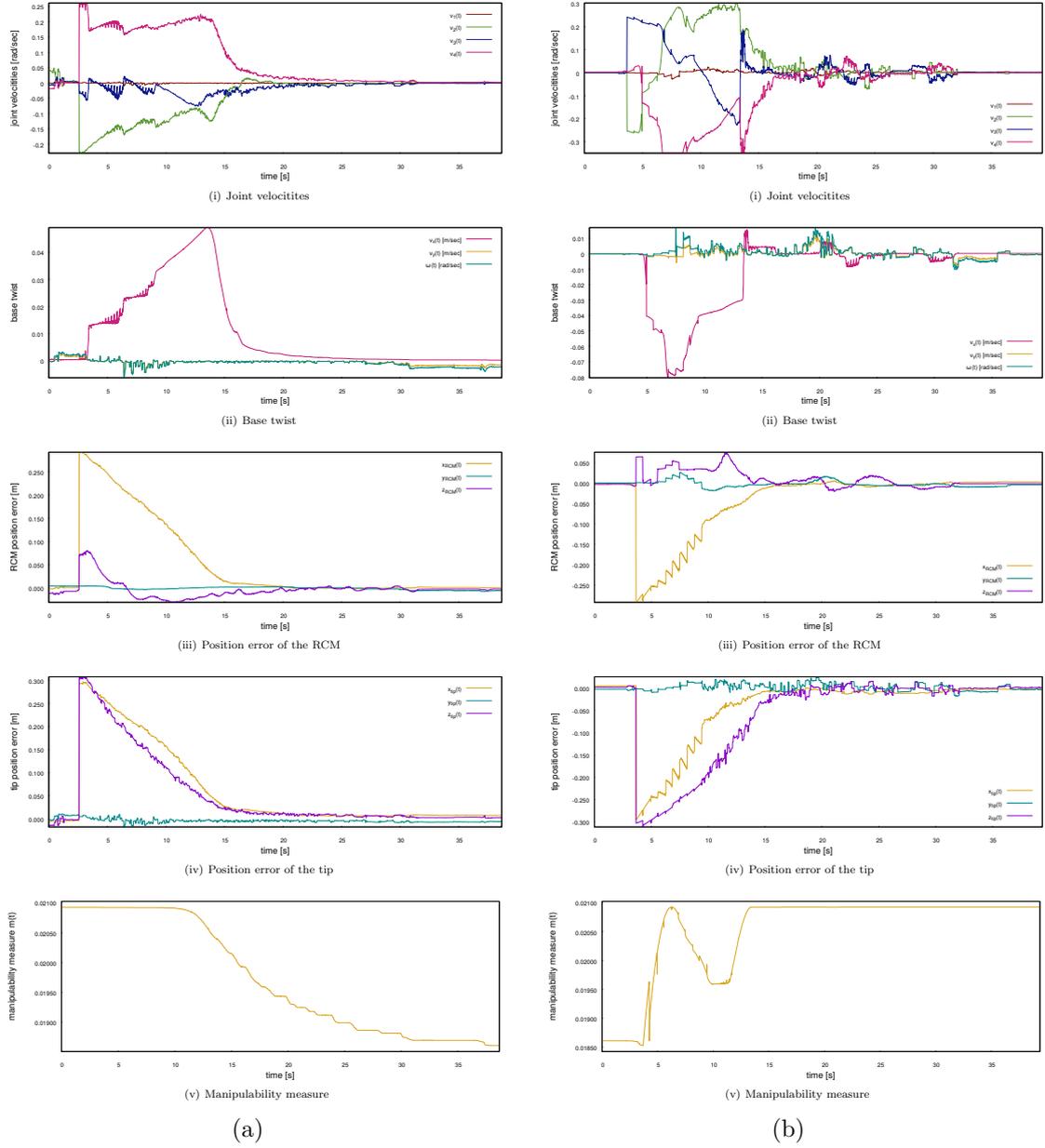


Figure 3.7: Execution of a planar approach task (parallel to the sagittal plane), with $\lambda=0.0^\circ$, $-12.0^\circ \leq \phi \leq 15.0^\circ$ and $0.5m \leq \rho \leq 0.8m$. Column (a) presents the outcomes when varying (ϕ, ρ) from $(-12.0^\circ, 0.8m)$ to $(15.0^\circ, 0.5m)$. Column (b) presents the outcomes when varying (ϕ, ρ) from $(15.0^\circ, 0.5m)$ to $(-12.0^\circ, 0.8m)$. The first and second row presents the control commands (joint velocities and base twist, respectively); third and fourth rows, respectively, the RCM and tip errors; the fifth and last row the value of the manipulability measure.

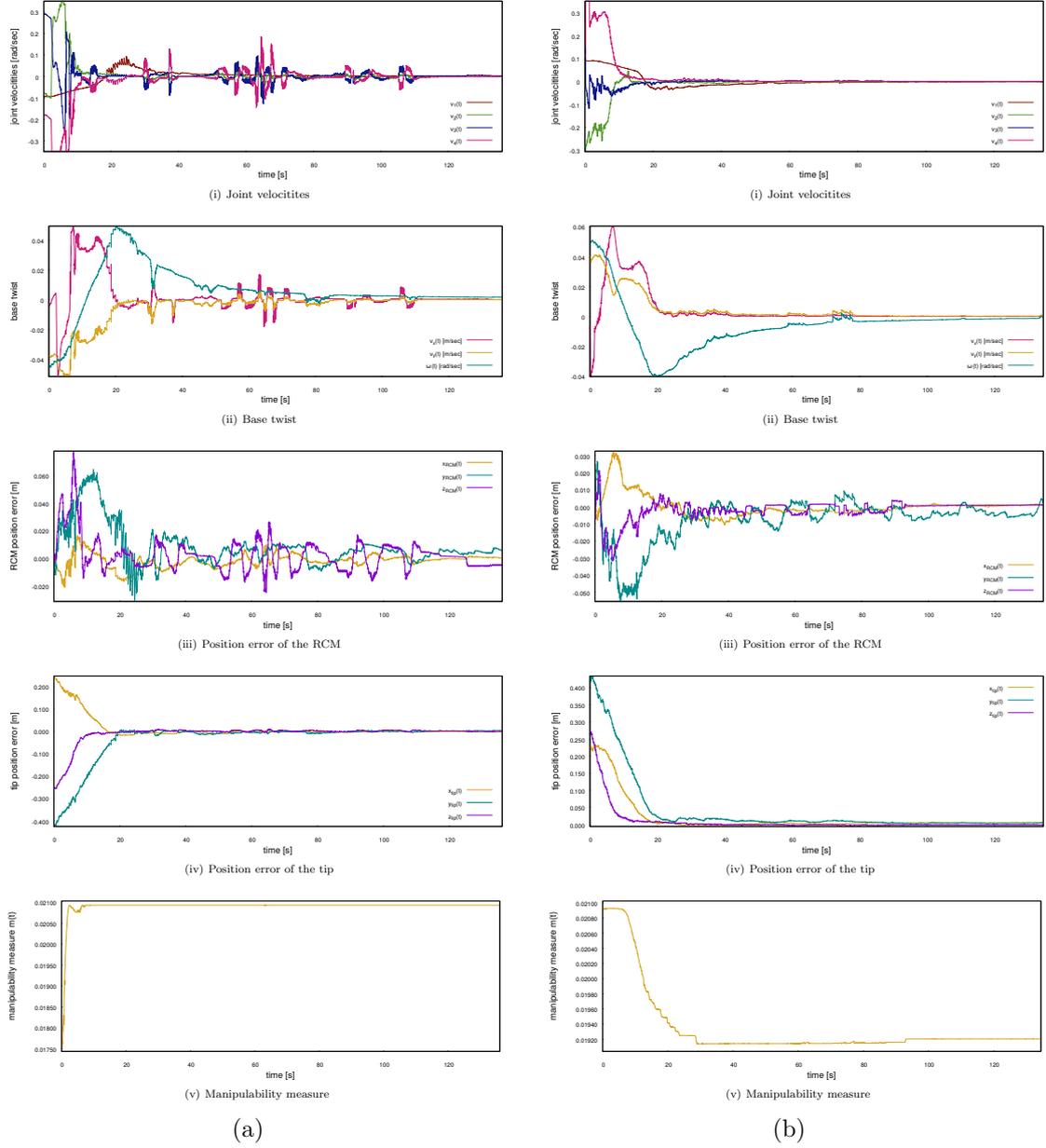


Figure 3.8: Execution of a geodesic orbit with constant $\rho=0.5m$, with $-30.0^\circ \leq \lambda \leq 30.0^\circ$ and $-15.0^\circ \leq \phi \leq 15.0^\circ$. Column (a) presents the outcomes when varying (λ, ϕ) from $(-30.0^\circ, 15.0^\circ)$ to $(30.0^\circ, -15.0^\circ)$. Column (b) presents the outcomes when varying (λ, ϕ) from $(30.0^\circ, -15.0^\circ)$ to $(-30.0^\circ, 15.0^\circ)$. The first and second row presents the control commands (joint velocities and base twist, respectively); third and fourth rows, respectively, the RCM and tip errors; the fifth and last row the value of the manipulability measure.

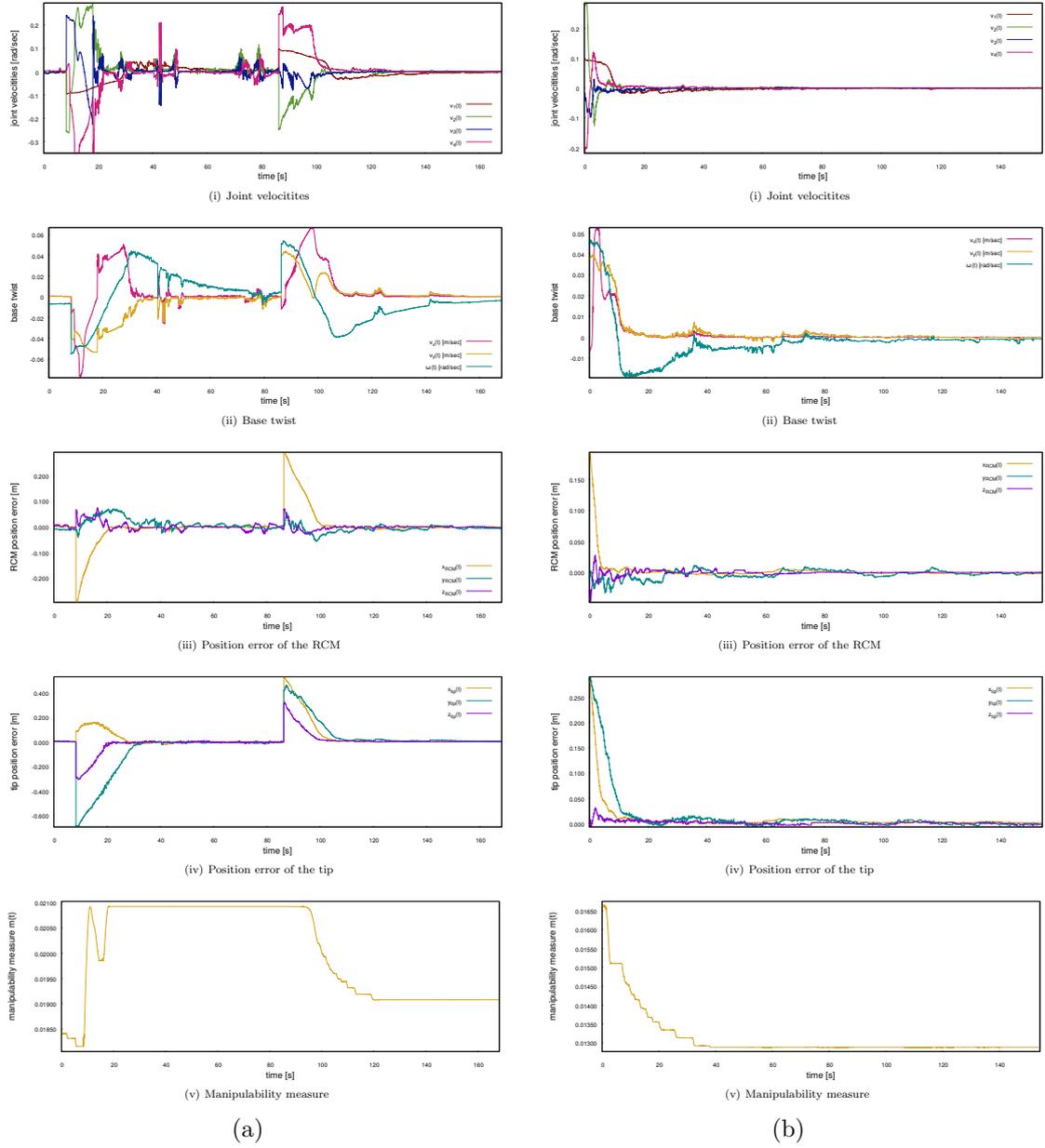
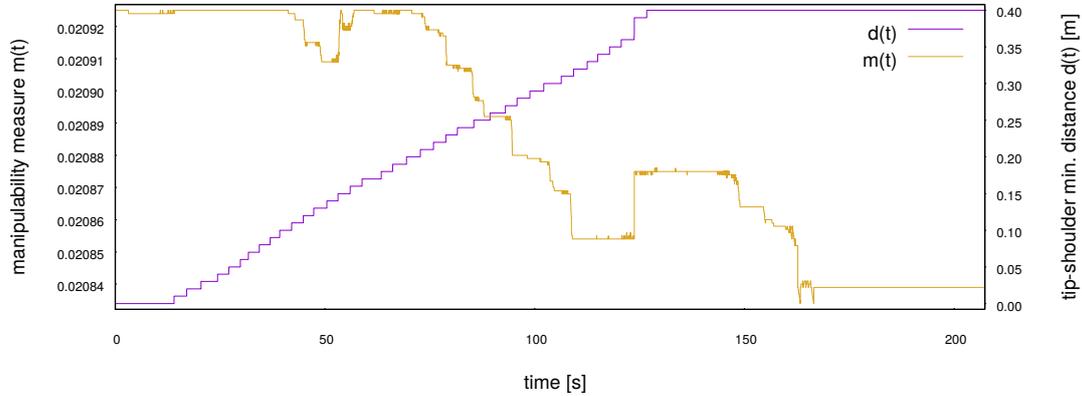
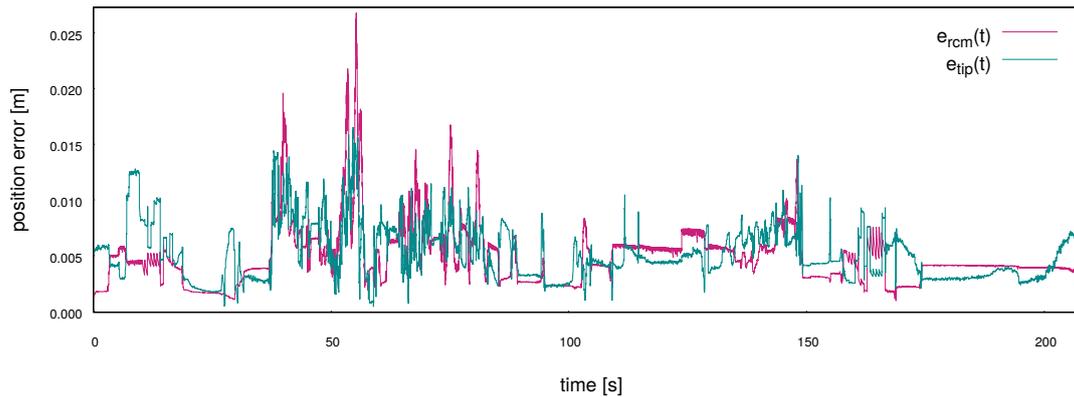


Figure 3.9: (a) Execution of a cyclic 3D task starting and ending at $(\lambda, \phi, \rho) = (-30.0^\circ, 15.0^\circ, 0.5m)$ while passing through $(\lambda, \phi, \rho) = (30.0^\circ, -12.0^\circ, 0.8m)$. (b) Execution of a point to point 3D task from $(\lambda, \phi, \rho) = (30.0^\circ, 12.0^\circ, 0.8m)$ to $(\lambda, \phi, \rho) = (0.0^\circ, 15.0^\circ, 0.6m)$. In both cases, the first and second row presents the control commands (joint velocities and base twist, respectively); third and fourth rows, respectively, the RCM and tip errors; the fifth and last row the value of the manipulability measure.



(a) Manipulability measure with respect the self-collision avoidance threshold.



(b) Evolution of the tip and RCM positional errors during reconfiguration by self-motion.

Figure 3.10: Evaluation of self-motion reconfiguration during the execution of a static task. (a) The manipulability measure monotonically decreases while the self-collision avoidance threshold (tip-shoulder distance) is increased; in particular, it is observed that although a local manipulability maximum is attained during reconfiguration, it is quickly lost by the effect of the self-collision avoidance constraint. (b) Task errors present bounded variations under 1.5cm during self-reconfigurations of the manipulator arm and base, except in cases where noise peaks locally disturb the robot state; during the experimental evaluation two different noise sources has been identified: localized random fluctuations of the target pose estimation and non-smooth joint velocities jumps during reconfiguration due under damped control of joint torques near singularities.

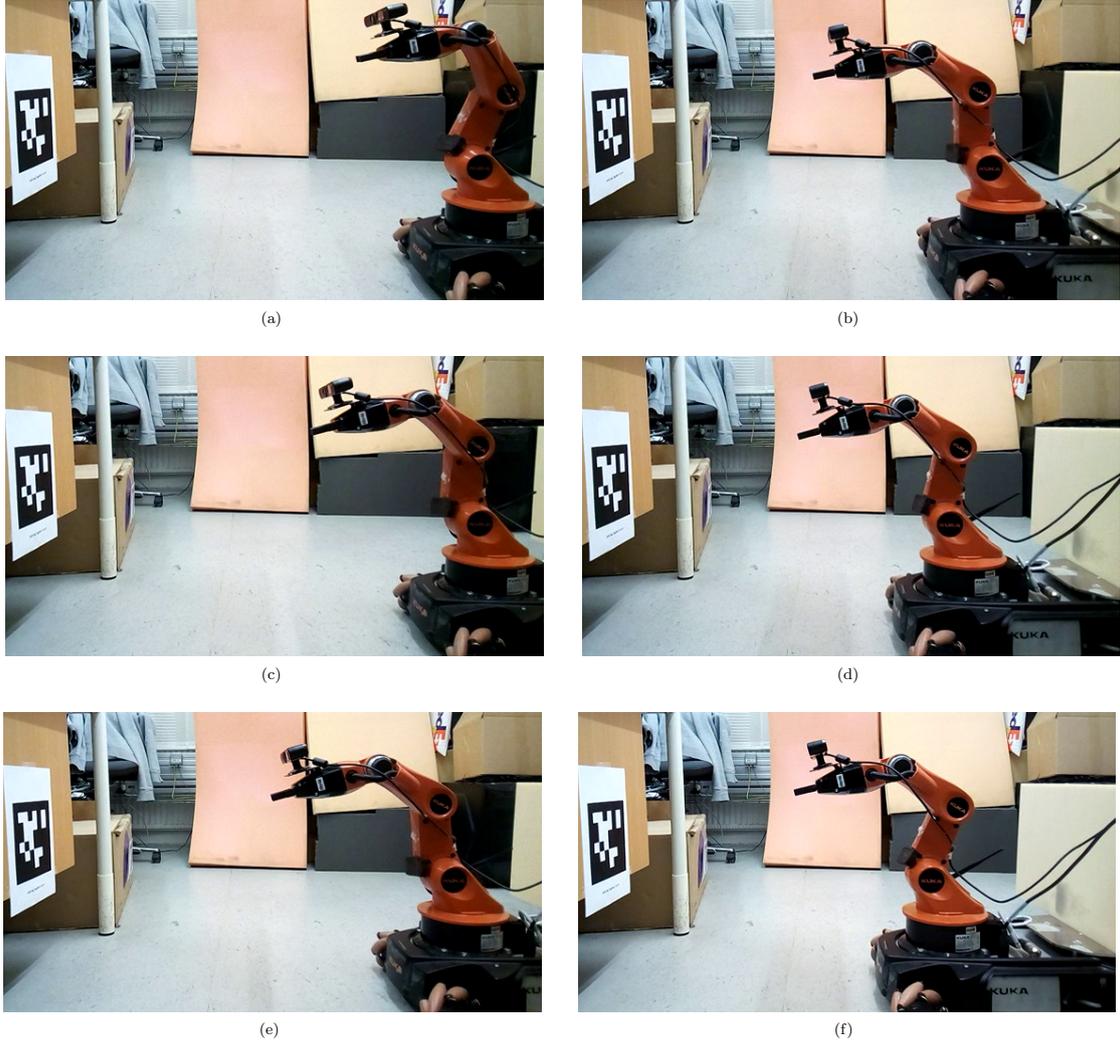


Figure 3.11: Example of self-reconfiguration of the robot along the sagittal plane during the execution of constant latitude $\phi=15.0^\circ$ and longitude $\lambda=0.0^\circ$ task, with orbit distance ρ varying from $0.9m$ to $0.5m$. Image frames are organized by column, from top to bottom and left to right. As observed on frames (a), (c) and (e), the initial robot motion is mainly driven by the primary task goal, moving forward the end-effector toward the desired position. As soon as the arm configuration lost a dexterous pose, as observed in frames (e), (b) and (d), the manipulability maximizing potential field forces a self-reconfiguration to produce a dexterous pose by coupling arm velocities and base twist commands. Asymptotically, the robot reach the final pose shown in frame (f). The symmetric task, with orbit distance ρ varying from $0.5m$ to $0.9m$ presents the symmetric behaviour, implying that the proposed approach leads to repeatable solutions.

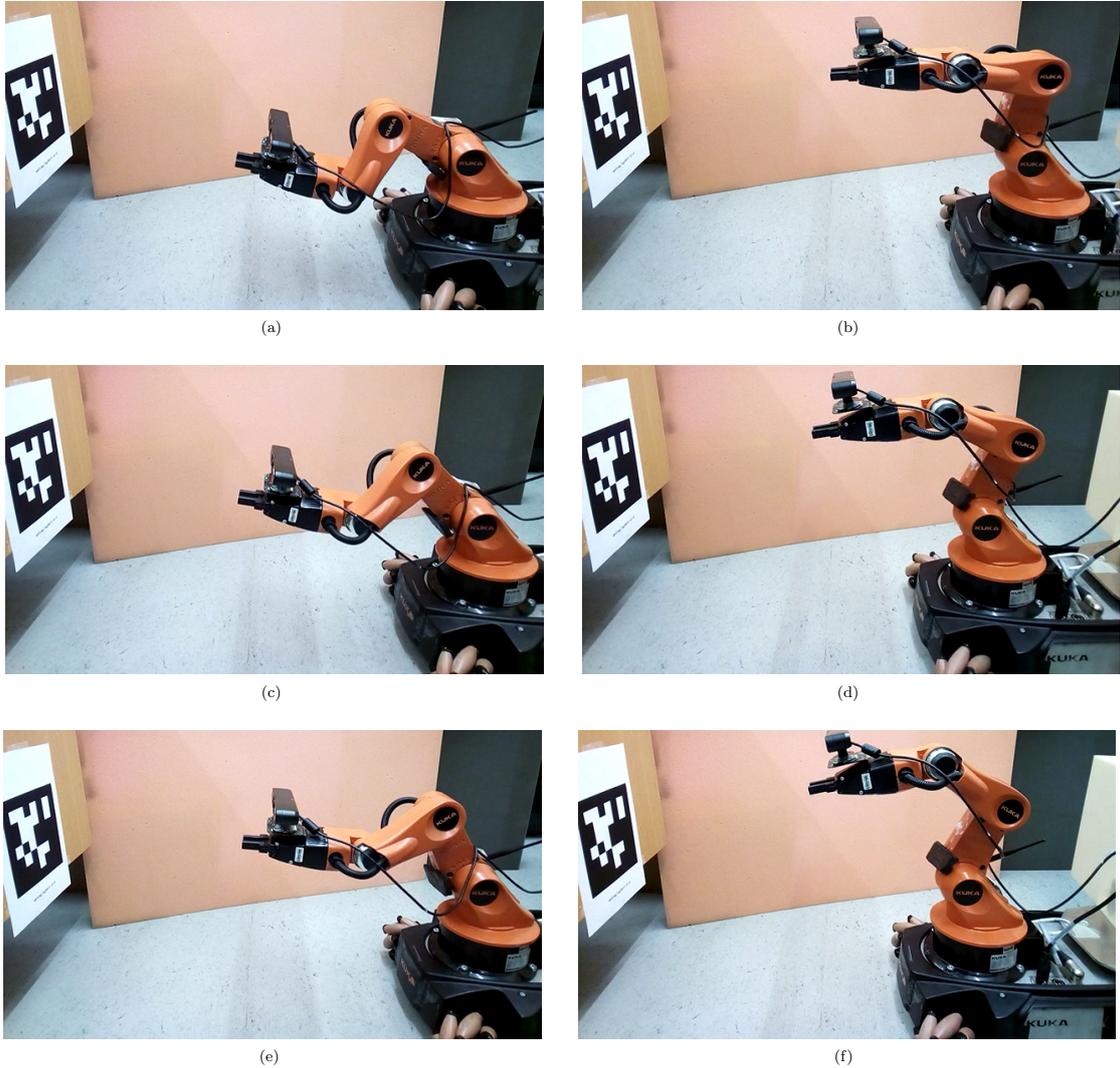


Figure 3.12: Example of self-reconfiguration of the robot along the sagittal plane during the execution of constant longitude $\lambda=0.0^\circ$ and orbit distance $\rho=0.5m$ task, with latitude ϕ varying from -15.0° to 15.0° . Image frames are organized by column, from top to bottom and left to right. As observed on frames (a), (c) and (e), the initial robot motion is driven by the primary task goal, moving upwards the end-effector toward the desired position. As soon as the arm configuration lost a dexterous pose, as observed in frames (e), (b) and (d), the manipulability maximizing potential field forces a self-reconfiguration to produce a dexterous pose by coupling arm velocities and base twist commands. Asymptotically, the robot reach the final pose shown in frame (f). The symmetric task, with latitude ϕ varying from 15.0° to -15.0° presents the symmetric behaviour, implying that the proposed approach leads to repeatable solutions.

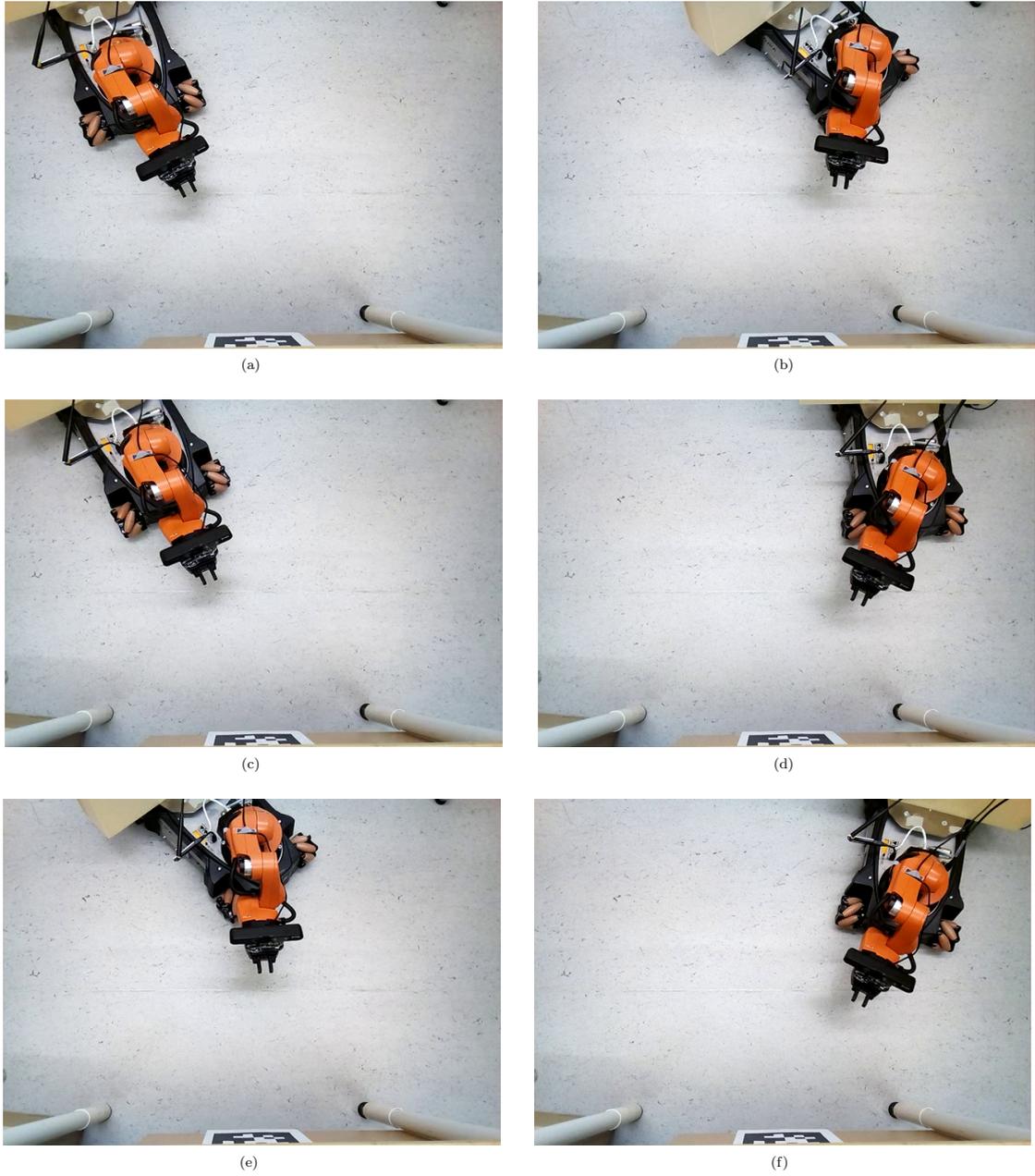


Figure 3.13: Example of self-reconfiguration of the robot along the transverse plane during the execution of constant latitude $\phi=-15.0^\circ$ and orbit distance $\rho=0.5m$ task, with longitude λ varying from 30.0° to -30.0° . Image frames are organized by column, from top to bottom and left to right. As observed on frames (a), (c) and (e), the initial robot motion is driven by the primary task goal, moving sideways in a circle trajectory without preserving the base alignment with respect the x -axis of the end-effector. When the misalignment grows and the primary task error decreases, as observed in frames (b) and (d), the alignment constraint induces a self-reconfiguration by coupling the velocity \dot{q}_1 of the first joint of the arm together with the base angular velocity ω . Asymptotically, the robot reach the final pose shown in frame (f). The symmetric task, with longitude λ varying from -30.0° to 30.0° presents the symmetric behaviour, implying that the proposed approach leads to repeatable solutions.

Chapter 4

Robust collision detection in articulated tracked vehicles¹

Articulated tracked vehicles are widely used in contexts where terrain conditions are difficult and unpredictable, or the environment is hazardous for humans, examples are referred in [12, 69, 59]. For better traction on harsh terrains these robotic platforms can either increase or decrease the tracks contact area with the ground by actuating their active sub-tracks, namely *flippers*, placed at the end of each track. Flippers ensure a good contact with the ground and several operating configurations, according to their positions: stretched, walking, parking, and so on. For example, in a stair-climbing task, flippers have to assume a stretched configuration to both ensures the contact with the ridges of the stairs and prevent the robot from tip-over. Conversely, within narrow passages, flippers have to be lifted to enhance the robot mobility, minimizing the contact surface with the ground. To overcome gaps and holes or to cross over different shaped objects, flipper configuration should be used to avoid bumps at the negative side of the obstacle.

Today, an increasing number of approaches is focusing on autonomous control of these configurations or for assisting remote control (see for example [52, 80, 102, 78, 38, 42]), this is still an open problem and the above mentioned approaches provide case studies and not yet feasible solutions. Indeed, terrain adaptability depends not only on the mechanical design but mainly on the controller ability to accurately adapt the active parts of the tracks. Even if the terrain surface is well approximated on the basis of vision and point cloud modeling, a compliance inter-

¹This is an adaptation of the original conference article “*Terrain contact modeling and classification for ATVs*”, included in the Proceedings of the IEEE International Conference on Robotics and Automation (ICRA) [43]

action between active parts of the tracks and the terrain needs to be established. The main difficulty with tracked robots is both a lack of proximity sensors, on the flippers, for the underfoot terrain profile and the impact of the tracks with the terrain inducing a continuous vibration of the robot body, therefore the contact cannot be considered in terms of a phenomenon characterized by short duration and rapid energy dissipation [44] as for example in the case of legged robots and manipulators [3]. In synthesis, unlike wheeled mobile robots, deriving an explicit dynamic model, articulated tracked vehicles have both to deal with the continuous contact with a changing surface and to account for any dynamic effect generated by the track-terrain interaction. This problem, for tracked robots is defined as the *traversability problem*, which requires both a knowledge of the terrain and knowledge of all the internal and external forces acting on the robot.

In this work we consider some specific tracked robot model in which the active parts of the tracks are flippers, and present a method that estimates the flippers terrain contact event. Since any contact event can be associated to an (unexpected) collision between the flippers and the environment, by considering any flipper unit as part of a tree-shaped open kinematic chain rigidly attached to a mobile base, we can unveil the contact event by resorting to state-of-the-art collision detection strategies for robot manipulators, based on unexpected transient perturbations of the dynamics of the failure signal or *residual*. The idea is to consider the robot locomotion and the unexpected collisions generating the contact as the only disturbances affecting the flippers dynamics. Here, by collision we refer to two different types of events. The first, in which some object of the environment exerts force over the flippers for a relatively short time. The second, in which the object may prevent further movement of the flippers. In both cases, we assume that the interaction forces are bounded and don't damage the robot.

Real-time detection of collisions for manipulators has been widely studied in the literature [96, 44, 55, 22, 48, 20] and several approaches have been proposed, based either on comparison with nominal torques on desired motion of the robotic arm or on the parallel simulation of robot dynamics. Approaches of the second type are typically based on an accurate dynamic model of the manipulator, and in most systems this is complemented with a sensory apparatus measuring the presence of collision forces that produce work at the contact. In particular, we consider the generalized momenta fault detection and isolation (FDI) scheme [21, 22, 48]. This method estimates the residual signal from a non-linear observer of the manipulator generalized momenta and relies only on both the kinematics measurements provided by the encoders and the torque values applied to the actuators.

From a geometric viewpoint, FDI cannot be achieved on one particular input channel in the presence of non structured disturbances acting on the same channel [21, 26]. That is, any unexpected collision of the flipper with the environment introduce disturbances, which resemble precisely those generated by the locomotion of the mobile platform. However, under the hypothesis that the residual signal presents different patterns in correspondence to each disturbance source, we can apply a classification method to discriminate the contact event between the flipper and the environment. To this end the wavelet packet transform is used to both decompose the signal and generate from different sub-bands a feature space. Finally, a sparse support vector machine (SVM), based on feature selection [97], is used to discriminate the contact signal.

In this work we focus on the collision detection of a single flipper with the terrain. The approach introduces explicitly the uncertainty deriving from the approximations of the dynamic effects of track-soil interaction, and discriminates the contact within the residual signal. The need to resort to learning methods, in the presence of uncertain signal with no clear noise contribution, was already highlighted in [65]. Our approach is based on the idea that the main theoretical principles at the basis of collision detection for robot manipulation [22] can be embedded within a statistical learning framework.

The rest of the chapter is organized as follows. Section 4.1 introduces the proposed model for the residual signal of the flippers. Section 4.2 describes the classification method based on wavelet expansion and the feature selection scheme. Section 4.3 discusses the experimental evaluation, and finally, the conclusion of the paper are found in section 4.4.

4.1 Problem statement

Based on the fault detection and isolation (FDI) scheme proposed in [21], we develop the hypothesis that unknown generalized forces or torques disturbing the manipulator dynamics can be represented by a linear mixing of basis functions that can be identified and classified to discriminate between the disturbance sources. Given the standard manipulator dynamic model

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{g}(\mathbf{q}) + \mathbf{F}_v\dot{\mathbf{q}} + \mathbf{F}_s\text{sgn}\{\dot{\mathbf{q}}\} = \boldsymbol{\tau} - \boldsymbol{\tau}_D \quad (4.1)$$

where \mathbf{q} and $\dot{\mathbf{q}}$ are the measured joint positions and velocities of the manipulator, $\mathbf{M}(\mathbf{q})$ is the positive definite symmetric inertia matrix, $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}}$ is the Coriolis and centrifugal vector, $\mathbf{g}(\mathbf{q})$ is the gravity vector, F_v and F_s are, respectively, the pos-

itive diagonal viscous and friction matrices, $\boldsymbol{\tau}$ the commanded nominal torques and $\boldsymbol{\tau}_D$ the unknown disturbance torques. Under the assumptions that nominal input torques $\boldsymbol{\tau}$, a measurement of full manipulator's state $(\mathbf{q}, \dot{\mathbf{q}})$ and an identified dynamic model of the manipulator are available, the generalized momentum FDI framework states that the residual signal \mathbf{r} is governed by

$$\mathbf{r} = \mathbf{K} \left[\int (\boldsymbol{\tau} - \boldsymbol{\alpha}(\mathbf{q}, \dot{\mathbf{q}}) - \mathbf{r}) dt - \mathbf{p} \right] \quad (4.2)$$

where $\boldsymbol{\alpha}(\mathbf{q}, \dot{\mathbf{q}})$ depends on the system dynamics, \mathbf{K} is a positive definite diagonal matrix and \mathbf{p} is the generalized momentum². Furthermore, \mathbf{r} corresponds to an exponentially stable linear filter driven by the unknown disturbance $\boldsymbol{\tau}_D$, that satisfies

$$\dot{\mathbf{r}} = -\mathbf{K}\mathbf{r} + \mathbf{K}\boldsymbol{\tau}_D \quad (4.3)$$

Assuming that the moving base link undergoes small accelerations, the disturbance torques $\boldsymbol{\tau}_D$ can be rewritten as

$$\boldsymbol{\tau}_D = \boldsymbol{\tau}_M + \boldsymbol{\tau}_U \quad (4.4)$$

$\boldsymbol{\tau}_M$ specify the disturbance due to the non modeled dynamics of the robot's locomotion and $\boldsymbol{\tau}_U$ the unexpected disturbances induced by collisions between the robot and the environment. Since the locomotion dynamics depends on the unknown shape of the terrain, we cannot provide an explicit model of the disturbance $\boldsymbol{\tau}_M$. However, we can assume that the disturbance patterns of the manipulator dynamics (and thus, those of the residual \mathbf{r}), generated by the robot's motion, are directly related to both the pattern fluctuations of the terrain and the bounded control inputs generating the motion. Therefore, assuming that the terrain is defined by a compact sufficiently smooth surface, and that control velocity inputs are bounded in amplitude and frequency, we can approximate the disturbance $\boldsymbol{\tau}_M$ as a linear combination of a number of basis functions $\boldsymbol{\phi}_i$, $i=1, \dots, n_M$. Analogously, we can assume that the collision of the manipulator with the environment can be approximated by a linear combination of another set of basis functions $\boldsymbol{\varphi}_j$, $j=1, \dots, n_U$. Therefore we can write the disturbances in the form

$$\boldsymbol{\tau}_M = \sum_{i=1}^{n_M} a_i \boldsymbol{\phi}_i, \quad \boldsymbol{\tau}_U = \sum_{j=1}^{n_U} b_j \boldsymbol{\varphi}_j \quad (4.5)$$

²The generalized momentum of a dynamical system is defined as $\mathbf{p} = \mathbf{M}(\mathbf{q})\dot{\mathbf{q}}$.

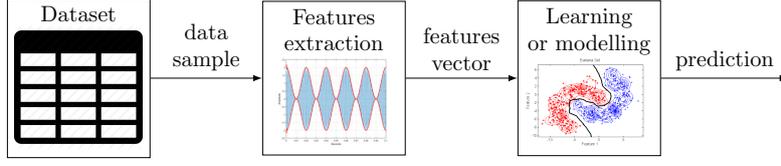


Figure 4.1: General learning framework.

Let $\Delta\phi \triangleq \text{span}\{\phi_1, \dots, \phi_{n_M}\}$, $\Delta\varphi \triangleq \text{span}\{\varphi_1, \dots, \varphi_{n_U}\}$ be the vector spaces spanned, respectively, by the set of basis functions ϕ_i and by the set of basis functions φ_j . We make the reasonable hypothesis that a pattern τ_M , spanning $\Delta\phi$, cannot be expressed as a linear combination of a basis in $\Delta\varphi$. Likewise, a pattern τ_U cannot lie entirely in $\Delta\phi \cap \Delta\varphi$. According to these hypotheses, from (4.3), the residual evolution can be expressed as

$$\dot{\mathbf{r}} = -\mathbf{K}\mathbf{r} + \mathbf{K} \left(\sum_{i=1}^{n_M} a_i \phi_i + \sum_{j=1}^{n_U} b_j \varphi_j \right) = -\mathbf{K}\mathbf{r} + \mathbf{K} \sum_{k=1}^{n_D} c_k \gamma_k \quad (4.6)$$

where the basis functions γ_k , by hypothesis, span the space $\Delta\gamma \triangleq \Delta\phi \cup \Delta\varphi$ and $n_D \leq n_U + n_M$. Now, given a residual dynamic, the computation of the coefficients c_k can lead to the identification (classification) of the disturbance source, under the proviso that the basis functions γ_k can discriminate the subspace $\Delta\phi \cap \Delta\varphi$. Indeed, we show that the contact sensing problem can be addressed via the identification of the unknown subset of basis functions γ_k not spanning the subspace $\Delta\phi \cap \Delta\varphi$.

4.2 Statistical learning framework

In general terms, classical learning framework can be decomposed into three main blocks, as shown on Figure 4.1. The starting point is a solid dataset describing the phenomena under study, followed by a features extraction block able to compute relevant information from samples and, based on this information, feed and train a learning model either for classification or regression.

Within our FDI context, the dataset block is generated through controlled experiments where desired motions, together with controlled disturbances (either due to locomotion or collisions), are assigned to the manipulator. The dataset is populated with different dynamic patterns of the residual together with its associated labels. On the other hand, the learning/modeling block correspond to a supervised learning classifier model. The resulting block diagram for our classification problem is shown on Figure 4.2. We observe that, in our pattern classification problem, the features extraction block can be thought as a de-mixing

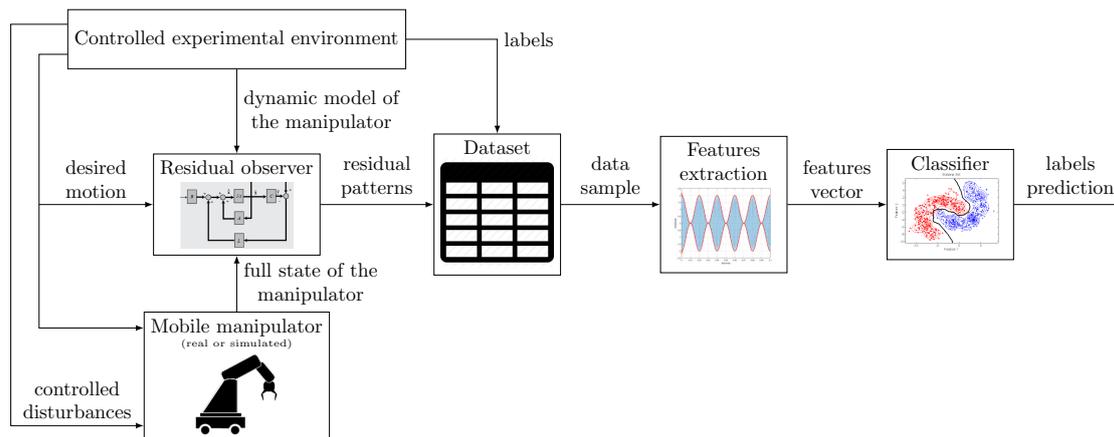


Figure 4.2: Embedding of the fault detection scheme inside the learning framework.

device, since residual patterns are associated to the superposition of the unknown disturbances due to locomotion and unexpected collisions.

At this point, the fundamental design choices of the learning framework are associated to the identification of optimal features extraction algorithm and the classifier model for the task. The choice of features extraction is highly related to the previous knowledge of and understanding of the signal under study. Hand-crafted or engineered features can be an optimal choice if there is an underlying well-known model relating the signal and the information carried by it. Instead, if no modeling basis are available, manual selection of features may be avoided in favor of automatically learned features, computed through unsupervised learning procedures. Regarding the identification of the most appropriate learning algorithm, it is worth noticing that provided a well-designed set of features, the choice of the final machine learning method may only play a minor role on the classification outcomes. In other words, if the feature space properly represents the meaningful and relevant information for classification, then the choice of the learning method is less significant since the information is readily available from features. If this is not the case, the choice of the method plays a prominent role in the classification outcomes, since intermediate or internal features representations maybe necessary to extract the desired information.

It turns out, that the fundamental issue in any learning framework is to properly extract and represent the necessary and sufficient information to perform classification. Therefore, one important step in the framework’s design is given by the evaluation of the quality of features representation. Such evaluation can be performed by measuring the relevance of individual components or subsets of the feature vector in the discrimination and prediction of labels, either by feature selection [46, 103] or relevance learning [86]. We note that such “relevance measuring”

techniques are often embedded directly in the learning algorithms [85, 97, 70, 27]. With all this considerations in mind, we introduce our design choice for the learning framework.

As stated in Section 4.1, the problem falls in the context of *spectral methods* [50, 99, 8, 100], that is, the numerical resolution of differential equations through functional approximation. Such functional approximations are defined as linear combinations of orthogonal basis functions. Two real functions f and g are said to be orthogonal if the scalar product

$$\langle g, f \rangle = \int_0^t f(s)g(s)ds$$

is zero. Standard families of basis functions include piecewise linear functions, polynomials, periodic functions (Fourier series) and wavelets [57]. The selection between one family or the other depends of on the a priori knowledge of the signal or function under study. When no knowledge is available, it is possible to compute a set of orthogonal bases based on data measurements [16, 66, 71, 104].

In our case, we have only *partial* knowledge of residual signal, since it depends on the dynamics of the flippers, model uncertainties, robot's locomotion and unexpected interactions with the environment. As a consequence, defining one particular family of basis functions could be harmful, since there's no way to guarantee that the functional approximation allow to extract the desired information. On the other hand, resorting to a pure empirical analysis implies a waste of information. One important remark is that our goal is to recognize faulty transients on the signal evolution.

With these ideas in mind, a convenient approach for the basis functions definition is the multi-resolution signal decomposition [2]. This approach introduces a space-frequency decomposition of the signal parametrized in terms of a family of wavelet functions [64]. In particular, we define our feature space from different sub-bands of the wavelet packed decomposition (WPD) [17] of the signal. Then, a sparse support vector machine (SVM), with enforced sparsity constraints for automatic feature selection [45, 70, 97], is used to discriminate the contact signal.

4.2.1 Wavelet packed decomposition (WPD)

Given the residual evolution in (4.6), we look for a decomposition of the signal able to determine those bases γ_k and the associated coefficients c_k that are meaningful

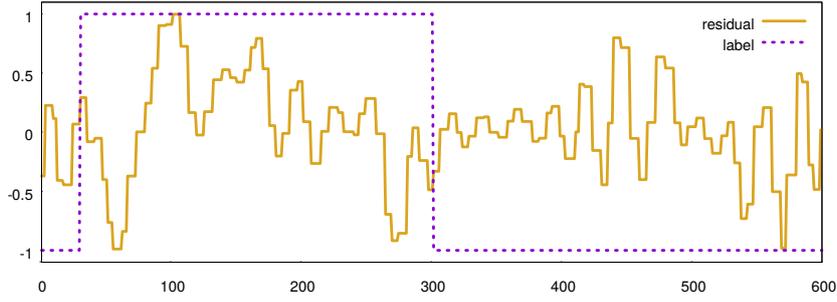


Figure 4.3: Typical residual patterns of a single joint obtained by simulations.

for the identification of contact, that is

$$\gamma_k \in H = \Delta\gamma \setminus (\Delta\phi \cap \Delta\varphi)$$

Let's us define $n \in \{1, \dots, 2^N\}$, with $N \in \mathbb{N}$, and consider a single channel r of the residual signal \mathbf{r} . Moreover, let $r[n]$ be a collected set of tagged samples of length L , with $L=2^N$, obtained by sampling at frequency 2^{-N} the residual channel r . To obtain a decomposition of the signal we resort to a multi-resolution technique based on WPD.

This technique can be applied to represent discrete signals as linear combinations of discrete wavelet packet bases of $l^2(\mathbb{Z})$, namely the Hilbert space of finite energy sequences. We assume that

$$r[n] \in \mathbb{W}_N \subset l^2(\mathbb{Z})$$

Under this assumption, we can apply the WPD transform, which recursively decomposes $r[n]$ in smoothed approximations and details at coarser resolutions by a cascade of discrete convolution and sub-sampling operations, using a pair of conjugate mirror filters. This recursive decomposition generates a binary tree [17] where each node n_k , at height h , with $k=2^h+j-1$, $j \in \{1, \dots, 2^h\}$, is associated with a set of orthonormal basis functions

$$\left\{ \psi_k[n - m2^{N-h}] \mid m \in \{1, \dots, 2^{N-h}\} \right\}$$

obtained by scaling and translating by 2^{N-h} a wavelet function ψ . Each set of bases defines a sub-space $\mathbb{W}_k \subset \mathbb{W}_N$ such that

$$\mathbb{W}_k \triangleq \text{span} \left\{ \psi_k[n - 2^{N-h}], \psi_k[n - 2^{N-h+1}], \dots, \psi_k[n - 2^{2N-2h}] \right\}$$

Let $\mathcal{K}_h = \{k \mid k=2^h+j-1, j=1, \dots, 2^h\}$ be the set of the indexes of the leaf nodes of

a wavelet packet binary tree of depth h . The spaces \mathbb{W}_k , with $k \in \mathcal{K}_h$ are mutually orthogonal, hence

$$\mathbb{W}_N = \bigoplus_{k \in \mathcal{K}_h} \mathbb{W}_k$$

where \bigoplus denotes the direct sum of spaces. It follows that the union of the corresponding wavelet packet bases defines an orthogonal basis of \mathbb{W}_N . Since by hypothesis $r[n] \in \mathbb{W}_N$, let

$$d_k[n] = \langle r[n], \psi_k[n - m2^{N-h}] \rangle \quad (4.7)$$

with $\langle \cdot, \cdot \rangle$ denoting the inner product on $l^2(\mathbb{Z})$. Given a WPD of height h , we can approximate the discrete signal $r[n]$ as the linear combination of discrete wavelet packet bases as follows

$$r[n] = \sum_{\substack{m=1 \\ k \in \mathcal{K}_h}}^{2^{N-h}} d_k[n] \psi_k[n - m2^{N-h}] \quad (4.8)$$

The problem now is to identify the reconstructing discrete wavelets bases in (4.8) that discriminate the collision. As consequence, we define the coefficients (4.7) as features and feed a sparse SVM classifiers. Still, this definition of features leaves several issues opened, due to the parameters N and h used by the WPD to approximate the residual component $r[n]$. In particular, the choice of N affects both the length L of the observations and the tagging of the samples. That is, to apply the WPD decomposition, each observation groups together L samples of $r[n]$ under a single label. Moreover, the choice of the best level decomposition h affects the quality of the approximation. We have tested the effects of such parameters during the experimental evaluation and reported on Section 4.3.

4.2.2 Classification

The mapping of the signal $r[n]$ into features is obtained by selecting the 2^h leaves nodes of the perfect binary tree of height h computed through the WPD. By construction, any sample of length L has a features vector of length $L' > L$, therefore, it may be convenient to introduce a variable selection procedure by enforcing a sparsity constraint [45, 46, 70]. We shortly recall that, given a training set $(\mathbf{x}_i, y_i)_{i=1, \dots, k}$ with \mathbf{x}_i the observations (features) vector, $y_i \in \{-1, 1\}$ the labels, SVM training

amounts to learning a decision function

$$g_w(\mathbf{x}) = \mathbf{w}^\top \mathbf{x} - b$$

by solving the ℓ_2 -norm problem

$$\min_{\mathbf{w}} \left\{ \|\mathbf{w}\|_2^2 + C \sum_{i=1}^k \mathcal{L}(-y_i g_w(\mathbf{x}_i)) \right\}$$

with C a regularization parameter and \mathcal{L} a convex loss function. Most of the approaches to variable selection, by enforcing sparsity, have considered either the zero-norm $\|\mathbf{w}\|_0$ [108], whose minimization is NP-hard, or the ℓ_1 norm $\|\mathbf{w}\|_1$ [113], through the average hinge loss on the training data. Here we apply the approach of [97], which facilitates the two steps process of doing variable selection by linear SVMs and the final nonlinear SVM inference, since they both employ the same loss function. This approach is based on the idea of rescaling the input \mathbf{x} , element-wise multiplying it by a selection vector $\mathbf{s}=(s_1, \dots, s_k)^\top$, such that

$$\sum_i s_i \leq B, \quad s_i \in \{0, 1\}$$

and B controlling the sparsity of the decision hyperplane

$$\mathbf{w}^\top \mathbf{x} = (\mathbf{w} \odot \mathbf{s})^\top \mathbf{x}$$

where \odot is the element-wise matrix product. The associated optimization problem can then be formulated as follows, given S , the space of all selection vectors \mathbf{s} satisfying the constraints:

$$\begin{aligned} \min_{\mathbf{s} \in S} \min_{\mathbf{w}, \xi, \rho} & \left\{ \frac{1}{2} \left(\|\mathbf{w}\|_2^2 + C \sum_{i=1}^k \xi_i^2 - \rho \right) \right\} \\ \text{s.t.} & \quad y_i \mathbf{w}^\top (\mathbf{x}_i \odot \mathbf{s}) \geq \rho - \xi_i, \quad i=1, \dots, k. \end{aligned} \tag{4.9}$$

The outcome of the method is the vector \mathbf{s} choosing the best features. More details can be found in [97]. The whole computation for the tree expansion, variable selection and classification is illustrated in Algorithm 1.

Algorithm 1: Algorithm of the proposed approach

Data: ;

$r[n]$: Samples from single channel of residual vector \mathbf{r} ;

$\ell[n]$: Tags for each sample of $r[n]$;

$N \in \mathbb{N}$: Parameter for each data-set observation length;

h : Height of the wavelet packed tree;

ψ : Wavelet function;

\mathcal{L} : Loss function;

C_1 : Regularization parameter set;

C_2 : Penalty parameters set;

G : Kernel function \mathcal{K} parameters set;

k : Number of folds;

p : Percentage for test-set.

Result: ;

acc : classification accuracy;

\mathcal{M}^* : best model;

\mathcal{I}^* : indices of best features

```

1   $acc \leftarrow 0$ ;
2   $\mathcal{I}^* \leftarrow \emptyset$ ;
3   $\mathcal{M}^* \leftarrow 0$ ;
4   $\mathcal{D} \leftarrow \text{createDataSet}(r[n], \ell[n], N)$ ;
   /*  $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^M$   $x_i \in \mathbb{R}^{2^N}$   $y_i \in \{-1, 1\}$  */
5   $[\mathcal{J}, K_h, z] \leftarrow \text{extractFeatures}(\mathcal{D}, h, \psi)$ ;
   /*  $\mathcal{J} = \{(d_k[n], y_k)\}_{k=1}^M$   $d_k[n] \in \mathbb{R}^{n|K_h|}$  */
   /*  $z$  number of coefficients for each node  $n_k$  with  $k \in K_h$  */
6   $[\mathcal{D}_1, \mathcal{D}_2] \leftarrow \text{splitData}(\mathcal{J}, p)$ ;
   /*  $\mathcal{D}_1$  training set */
   /*  $\mathcal{D}_2$  test set */
7   $B \leftarrow \{n, 2n, \dots, \kappa n\}$ ;
   /* the elements of the set  $B$  controls the choice of the features associated with nodes;
       $\kappa > 1$  */
8  for  $i=1$  to  $\text{length}(B)$  do
9    for  $j=1$  to  $\text{length}(C_1)$  do
10      $\mathcal{I} \leftarrow \text{extractIndices}(\mathcal{D}_1, B(i), C_1(j), \mathcal{L})$ ;
        /* indices  $\mathcal{I}$  of selected features */
11      $\mathcal{M} \leftarrow \text{crossValid}(\mathcal{D}_1, \mathcal{I}, k, \mathcal{K}, C_2, G)$ ;
12      $a \leftarrow \text{test}(\mathcal{D}_2, \mathcal{I}, \mathcal{M})$ ;
13     if  $a \geq acc$  then
14        $acc \leftarrow a$ ;
15        $\mathcal{I}^* \leftarrow \mathcal{I}$ ;
16        $\mathcal{M}^* \leftarrow \mathcal{M}$ ;
17     end
18   end
19 end

```

4.3 Experimental evaluation

We now describe the experiments carried out to estimate the parameters of the model (4.2) and to validate the hypotheses (4.6). All experiments were done on a simulated environment developed with the rigid body dynamic simulation engine ODE. We simulated the motion of the robot, with different terrain shapes, suitably setting masses, inertia and friction forces. The simulator not only allows us to generate samples of \mathbf{r} , but also to tag each sample with a label, denoting the contact of the flippers with the obstacles within the simulated environment, through the collision checking engine.

The experiment setup was designed to evaluate the sensitivity with respect of the parameters N , height h of the WPD and B controlling the features selection. In particular, N is chosen to vary between 5 and 9 (first column of Table 4.1), to capture collisions of different physical nature. The choice of the range of h is based on the level of granularity of the bandwidth sub-division of the signals under consideration and the degree of approximation of the decomposition. We choose to vary h between 3 and 9 (second column of Table 4.1). The values of B have been chosen to be multiple of the number of coefficients of a leaf node at depth h , since the number of leaves nodes depend on h , while the number of coefficients by node varies with respect N . In these experiments we use the wavelet Daubechies 5 (db5), which is asymmetric, orthogonal and bi-orthogonal. We set $k=5$, $p=70\%$ and

$$\begin{aligned} C_1 &= \left\{ c_{1i} \mid c_{1i} = \log_2(\alpha_1), \alpha_1 \in \{-5, \dots, 5\} \right\} \\ C_2 &= \left\{ c_{2i} \mid c_{2i} = \log_2(\alpha_2), \alpha_2 \in \{-10, \dots, 10\} \right\} \\ G &= \left\{ g_i \mid g_i = \log_2(\alpha_3), \alpha_3 \in \{-10, \dots, 10\} \right\} \end{aligned}$$

Note that, these parameters have been chosen to follow the dyadic property of the decomposition.

For a sequence of tagged samples of a single channel of the residual vector \mathbf{r} and a fixed N , we build sub-sequences $r[n]$ of length 2^N . We denote with $\mathbf{x}_i \in \mathbb{R}^{2^N}$ the vector associated to this sub-sequence. Each vector \mathbf{x}_i is labeled with a label $y_i \in \{-1, 1\}$, computed by averaging the tags $\ell[n]$ and by comparing the average with a fixed threshold. With Algorithm 1 we generate different data-sets $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^M$ for each value in the range N (line 4). For each \mathcal{D} , a new data-set $\mathcal{J} = \{(d_k[n], y_k)\}_{k=1}^M$ is build (line 5), where the WPD is applied to each \mathbf{x}_i , for any value in the range of h . Fifth column of Table 4.1 reports the different sizes of

the data-set \mathcal{J} , as the parameter N changes. Sixth column of Table 4.1 reports the different dimensions of the feature spaces of \mathcal{J} , as both the parameters N and h change. Upon a data-set \mathcal{J} is splitted, to create training \mathcal{D}_1 and test set \mathcal{D}_2 , respectively, the algorithm, at line 10, extracts the best feature indices \mathcal{I} , given both the set B which controls the choice of the features associated with nodes and the set C_1 of the regularization parameters. Table 4.3 shows the number of features selected by the sparse SVM classifier as both N and h change. This set of indices, together with \mathcal{J} is given as input to the cross validation procedure, which computes both the best parameters of the model classifier \mathcal{M} . This model is then used for testing on the remaining data-set \mathcal{D}_2 . At each iteration, the best model \mathcal{M}^* , together with the best indices are stored on the basis of the accuracy on the test set.

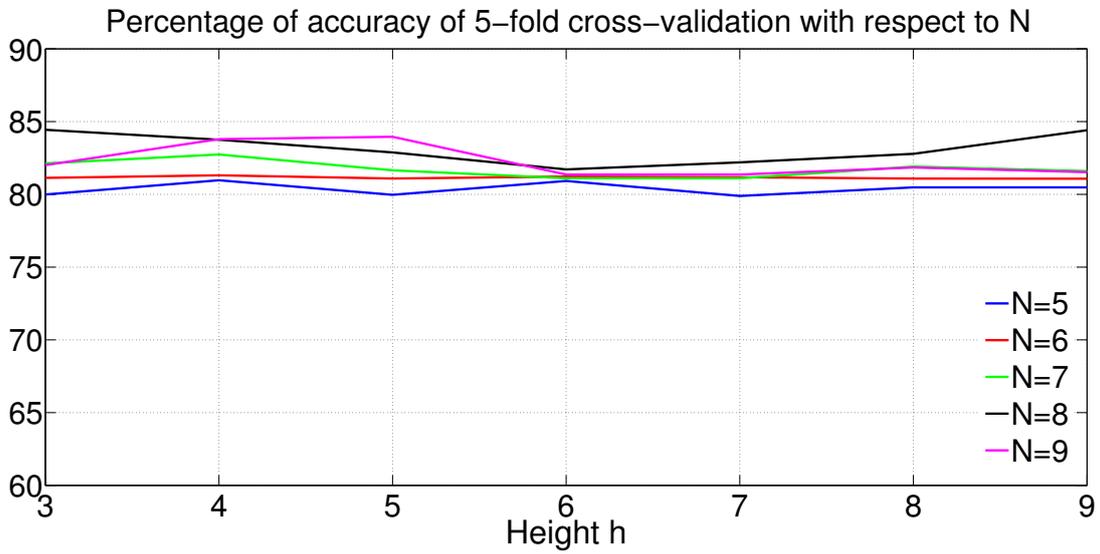
Figure 4.4(a) shows the trends of the cross validation accuracies, for each value of h , with respect to the parameter N . Conversely, Figure 4.4(b) shows the cross validation accuracies for each value of N , with respect to the h . On the other hand, Figure 4.5(a) depicts the accuracies on the test-set, for each value of h , with respect to the parameter N . Analogously, Figure 4.5(b) shows the trends of the accuracies on the test-set, for each value of N , with respect to the parameter h . Table 4.2 reports the overall results obtained during all the process of learning the classifier for the flipper contact, as well as the best parameters of the model. These results highlight that we obtain the best accuracy of 84.31% on the test-set, for $N=8$ and $h=5$. Moreover, for these values 96 features out of 512 have been selected. This implies that, the analysis of the residual is not enough to identify the contact of the flipper. In other words, the residual does not accurately model the track-soil interaction, since it is not possible to estimate with a high degree of confidence whether the contact of the flipper occurred or not. Additional features, representing the properties of the terrain should be also taken into account to improve the performance of the classification. However, the proposed model can be applied to real tracked robots, to obtain a preliminary estimate of the contact, since it turned out from the experiments that the WPD takes, in average, 0.1145 sec. to extract the features from the residual, and can then be applied in real-time applications. To conclude, in Figure 4.6 we show the wavelet packet binary tree associated with the best value of the accuracy on the test set, for $N=8$ and $h=5$. Thin circles around the leafs, denote the nodes at depth 5 which have been selected by the extract indices procedure. These nodes represents the signal bands in which frequencies of the contact event lie.

Table 4.1: Data-set organization

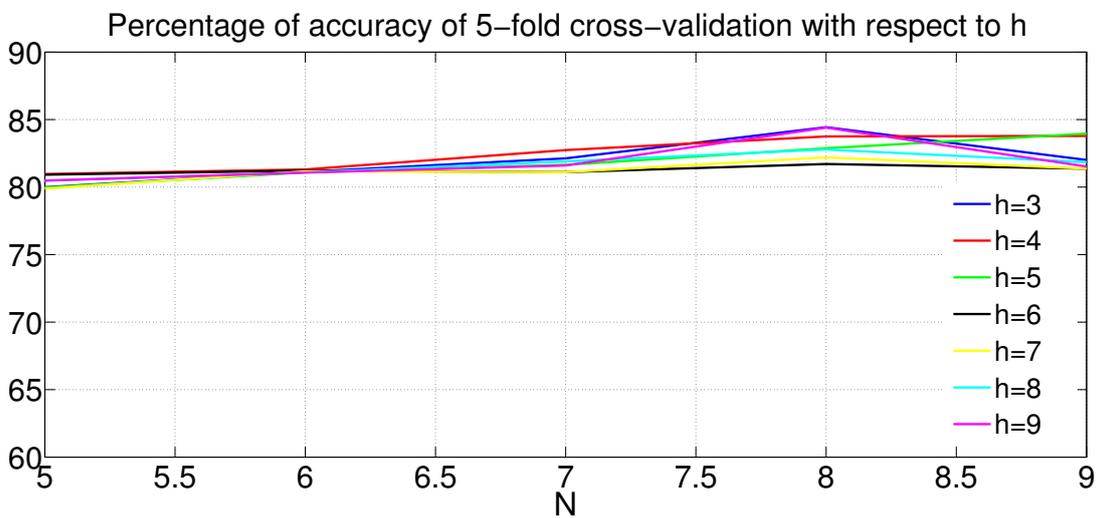
N	h	$\#f$	$\#n$	Data-set dim.	Feat. dim.	B
5	3	11	8	3878	88	11:11:22
5	4	10	16	3878	160	10:10:40
5	5	9	32	3878	288	9:9:72
5	6	9	64	3878	576	9:9:162
5	7	9	128	3878	1152	9:9:234
5	8	9	256	3878	2304	9:9:576
5	9	9	512	3878	4608	9:9:1530
6	3	15	8	2279	120	15:15:30
6	4	12	16	2279	192	12:12:48
6	5	10	32	2279	320	10:10:100
6	6	9	64	2279	576	9:9:189
6	7	9	128	2279	1152	9:9:288
6	8	9	256	2279	2304	9:9:576
6	9	9	512	2279	4608	9:9:1530
7	3	23	8	1466	184	23:23:46
7	4	16	16	1466	256	16:16:64
7	5	12	32	1466	384	12:12:60
7	6	10	64	1466	640	10:10:120
7	7	9	128	1466	1152	9:9:297
7	8	9	256	1466	2304	9:9:576
7	9	9	512	1466	4608	9:9:1530
8	3	39	8	1028	312	39:39:78
8	4	24	16	1028	384	24:24:48
8	5	16	32	1028	512	16:16:128
8	6	12	64	1028	768	12:12:144
8	7	10	128	1028	1280	10:10:390
8	8	9	256	1028	2304	9:9:288
8	9	9	512	1028	4608	9:9:576
9	3	71	8	617	568	71:71:142
9	4	40	16	617	640	40:40:80
9	5	24	32	617	768	24:24:120
9	6	16	64	617	1024	16:16:96
9	7	12	128	617	1536	12:12:144
9	8	10	256	617	2560	10:10:270
9	9	9	512	617	4608	9:9:477

Table 4.2: Best parameters

N	h	acc	cv	α_2	α_3	B	C_1
5	3	81.05	79.98	0	-2	22	32
5	4	80.38	80.96	1	-3	30	32
5	5	80.05	79.96	2	-4	54	4
5	6	80.27	80.91	-1	-2	45	0.125
5	7	81.01	79.88	3	3	63	8
5	8	80.84	80.47	10	9	27	32
5	9	80.71	80.47	4	-1	72	16
6	3	83.11	81.13	0	-3	15	32
6	4	82.49	81.30	2	-4	12	0.0313
6	5	82.39	81.08	2	-5	10	0.0313
6	6	82.81	81.21	-1	10	81	32
6	7	82.91	81.17	2	3	117	8
6	8	82.96	81.08	2	-4	279	0.125
6	9	83.06	81.07	4	-3	63	0.0313
7	3	82.64	82.12	2	-4	23	0.0313
7	4	82.48	82.74	1	-4	32	32
7	5	81.52	81.65	9	-9	12	32
7	6	83.43	81.10	10	10	30	32
7	7	83.43	81.10	10	-10	27	32
7	8	82.63	81.90	-1	8	99	0.0625
7	9	82.58	81.60	8	4	459	8
8	3	83.63	84.43	1	-5	39	0.0625
8	4	83.18	83.75	1	-5	24	0.0313
8	5	84.31	82.87	2	-6	32	0.5
8	6	82.5	81.71	1	-5	12	0.125
8	7	82.72	82.19	7	-10	30	0.5
8	8	82.95	82.78	3	-9	252	32
8	9	83.62	84.40	4	-3	207	0.5
9	3	81.88	82	-1	-6	71	0.0313
9	4	83.01	83.79	2	-6	40	0.125
9	5	77.35	83.95	1	-6	24	8
9	6	81.50	81.36	1	-6	16	0.625
9	7	83.01	81.36	1	-7	108	0.5
9	8	79.24	81.84	10	7	160	32
9	9	78.11	81.52	3	-7	63	0.0312

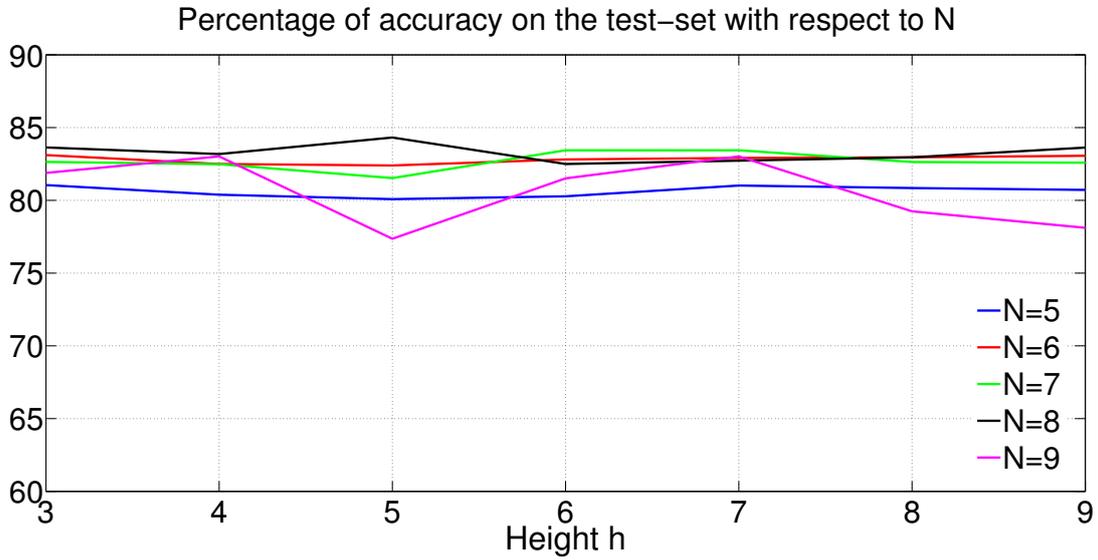


(a) Training accuracy, for each value of h , with respect to the parameter N .

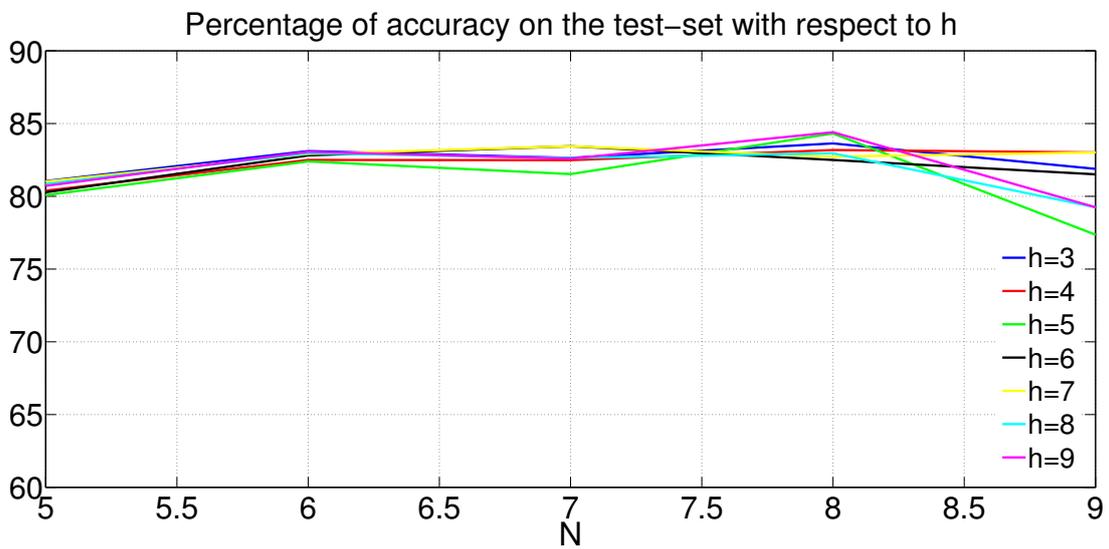


(b) Training accuracy, for each value of N , with respect to the h .

Figure 4.4: 5-fold cross validation accuracy



(a) Testing accuracy, for each value of h , with respect to the parameter N .



(b) Testing accuracy, for each value of N , with respect to the parameter h .

Figure 4.5: Testing accuracy

Table 4.3: Number of features selected out of total number of features

h	N				
	5	6	7	8	9
3	33/88	30/120	46/184	112/312	142/568
4	40/160	24/192	80/256	72/384	120/640
5	45/288	30/320	36/384	96/512	120/768
6	36/576	108/576	70/640	36/768	80/1024
7	27/1152	81/1152	63/1152	60/1280	264/1536
8	27/2304	27/2304	63/2304	432/2304	420/2560
9	63/4608	45/4608	387/4608	81/4608	162/4608

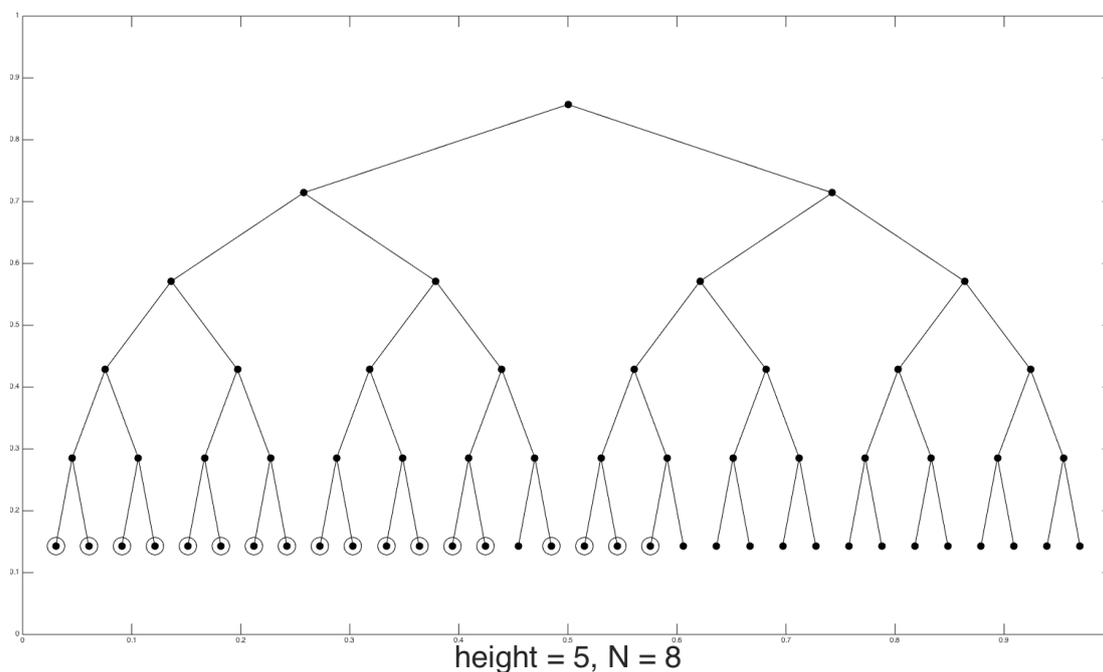


Figure 4.6: WPD tree with best testing accuracy. Circles highlight selected leaf nodes by the sparse SVM classifier.

4.4 Conclusions and future work

In this work we described a model for real-time detection of the contact between the flippers of an articulated tracked vehicle and the terrain. Contact identification is a crucial aspect for autonomous safe navigation. In fact, different control strategies can be designed on the basis of the model providing as feedback the occurrence of the flipper contacts [114], enhancing not only the motion capabilities of the robot but also the understanding or perception of the environment in the near surroundings. The proposed model extends the well-known FDI scheme of robot manipulators to kinematic chains with multiple non structured induced disturbances through the non inertial reference frame. In particular, we consider the case of flippers rigidly attached to a mobile platform. This extension is characterized by the analysis of the transient perturbations patterns of the residual signal evolution, disregarding its amplitude bounds. The method proves that it is possible to obtain terrain contact information even from sensorless flippers. To perform such analysis we propose a classification framework based on features extracted from the WPD of the residual signals and a SVM classifier. In particular, we define a feature space based on the WPD of the residual and use a sparse SVM to discriminate the contact event. The correctness of the model, on the application under consideration, has been demonstrated through simulations. It is worth noticing that the validity of the spectral model is clearly independent of the particular choice of the features extraction and classification techniques.

One practical limitation of our approach is the features selection procedure. WPD recursively decomposes a signal in smoothed approximations and details at coarser resolutions by a cascade of discrete convolutions and sub-sampling operations, using a pair of conjugate mirror filters. The result of this recursive decomposition is a binary tree, whose nodes have associated an orthonormal basis function, defined by the translation and scaling of a wavelet function in terms of the node's height [17]. WPD features can be obtained from the projection of the input signal (at the corresponding resolution) into the linear spaces spanned by the orthonormal functions of a set of nodes. In terms of empirical evaluation, one has to face the selection of the parameter N (defining the length $L=2^N$ of batch segments of the signal decomposition), the maximum height h of the decomposition tree and the set of nodes that will be employed for the definition of features. In the developed work, we had chosen optimal values of N and h based on classification outcomes and selected the set of all leaf nodes of a fully expanded tree decomposition for features definition. Our choice in defining the set of features is purely heuristic and the possibility of combining nodes from different heights

inside the tree has not been tested.

Moreover, by considering that WPD recursively decomposes a signal by a cascade of discrete convolutions and sub-sampling operations, it is possible to observe that such procedure greatly resembles the basic layering structure of a deep convolutional neural network (DCNN), with two fundamental differences. In the case of DCNN we can define arbitrary batch lengths L and the optimal sub-sampling strategy can be learned from data (WPD instead uses a fixed sub-sampling factor). With this in mind, the deep network alternative seems to provide also methodological advantages over the WPD, since it learns from data not only the optimal sub-sampling rule but also the shape of the convolution kernels, the equivalent of the wavelet function underlying the signal decomposition. Moreover, by choosing a priori the desired number of features, we know that the unsupervised features learning algorithm extract the most relevant information that can be encoded in such a feature vector. It is a clear advantage with respect to the WPD of the residual signal, since we are forced to take the whole set of node's features because we have no idea how relevant information is distributed among them.

By taking the projection coefficients of all leaf nodes of the WPD tree, the dimension of the feature space becomes very large and the need for a feature selection or relevance learning step becomes critical, not only for feature validation but also for reduction of the computational effort of the learning algorithm. We tried the feature selective SVM algorithm of [97]. The great limitation we found on feature selection is that we have no idea of the relevance of single components of the feature vector, but only a binary label telling us if the component was used or not by the classification algorithm. Relevance learning [5, 86, 87] is a valid alternative to overcome this limitation, since a confidence measure (formally a *metric*) weighting the significance of each feature is obtained as part of the classification outcome. That is, a continuous measure for each feature, proportional to its relevance with respect the other components of the feature vector. Feature selection can be considered as a particular instance of relevance learning, where a binary relevance measure is computed for each feature (zero-one or useless-useful).

Chapter 5

Coplanarity constrained monocular visual odometry

5.1 Introduction

Visual odometry (VO) is the process of estimating the egomotion of an agent using only the input of a single or multiple camera sensors attached to it [83]. In contrast to the general structure from motion (SfM) system [33], the idea of the VO is to provide a locally consistent estimation of the agent's motion by sequential (i.e. real-time, frame by frame) analysis of the input data. In particular, monocular VO refers to the special case when only one camera sensor is employed. Monocular VO is highly susceptible to drift, not only because errors introduced by the sequential motion estimation accumulate over time, but also because the absolute scale of the agent's trajectory cannot be uniquely determined. Therefore, the computation of accurate drift aware motion estimates relies on the availability of additional geometric constraint necessary to improve the posedness of the problem.

For example, in the case of vehicles with constrained motions, different cues have been exploited like the nonholonomic constraints of motion [77] or the constant sensor height by robust estimation of the ground plane [92]. A general alternative is to take advantage of the trinocular vision [75] to overcome the scale ambiguity between successive frames. Despite, many state of the art monocular SLAM systems still rely on a VO core based on the stereo geometry arising from the sensor's motion, followed by a local bundle adjustment optimization over a small subset of camera poses [30, 68, 31]. Mostly because the simplicity of the stereo geometry of a calibrated device, good trade-off between accuracy and implementation complexity of the feature matching or tracking, handling of applications with or without motion constraints, and the growing family of differential geomet-

ric algorithms that have been developed for fast and accurate computation of the associated essential matrices [62, 51, 94].

We can rework the above ideas by noting that unconstrained optimization approaches are usually preferred, in the sense that frame-to-frame geometries are estimated independently. As consequence, most differential geometric optimization or “on-manifold” schemes are exclusively designed for the general unconstrained estimation problem, where the compound sensor motion is assumed to lie on the manifold \mathcal{M} given by the product of the pair-wise motion manifolds \mathcal{M}_{ij} , i.e. $\mathcal{M}=\mathcal{M}_{12}\times\mathcal{M}_{23}\times\dots$. However, simple geometrical facts [49] suffice to show that the compound sensor motion does not lie on the entire product manifold but on a sub-manifold of it, i.e. $\mathcal{M}\subset\mathcal{M}_{12}\times\mathcal{M}_{23}\times\dots$. For example, the calibrated geometry between 3 views is given by 12 degrees of freedom, while the independent pair-wise stereo analysis requires the estimation of 15 parameters.

Within this context, the present work aims to produce a reliable and accurate constrained on-manifold algorithm that can jointly recover the pair-wise stereo geometries between a small subset of views by considering the spatial coplanarity of pair-wise baselines between three different calibrated views as the optimization constraint. In particular, our research efforts have been focused in extending the essential manifold optimization algorithm derived by Helmke et al. [51] for the two-views case.

The advantage of using the coplanarity between baselines to constraint the stereo geometry estimation is that it does not require the knowledge of the true shape of the scene nor the relative scale between pair-wise geometry estimates (note that the coplanarity between three vectors depends only on their spatial orientation and not on their lengths) For a single moving sensor, this implies that the knowledge of the true lengths of the camera translations are not necessary to impose the constraint, and thus, neither to refine the joint estimation of the pair-wise epipolar geometries, which by definition can only be known up to scale. Moreover, a novel measure of consistency between essential matrices [11] is introduced without requiring the a priori knowledge of the baselines.

The rest of the chapter is structured as follows. Section 5.2 provides a review of the stereo geometry of calibrated views, including the essential manifold parametrization and the derivation of the iterative optimization algorithm proposed in [51]. Next, in Section 5.3 the formal algebraic description of coplanarity constraint is introduced. After that, Section 5.4 describes two novel optimization approaches for pair-wise stereo geometry exploiting the coplanarity constraint. Finally, in Section 5.5 we provide some concluding.

5.2 Stereo geometry revisited

5.2.1 Preliminaries

Assuming that we have two calibrated views \mathcal{I}_1 and \mathcal{I}_2 of some 3-dimensional static scene, taken with the same camera whose internal parameters \mathbf{K} are known. We would like to compute the relative rigid transformation between the camera poses used to acquire such images. All homogeneous point correspondences $\mathbf{x}_1 \leftrightarrow \mathbf{x}_2$ are assumed well defined, known and already *normalized* or back-projected with respect the calibration matrix

$$\mathbf{x}_i = \mathbf{K}^{-1} \tilde{\mathbf{x}}_i \quad i=1, 2$$

where the tilde $\tilde{\cdot}$ means that the coordinates are given in projective (homogeneous pixel) coordinates. In the following paragraphs, the rigid transformation between views is defined by the transformation from the second view frame to the first, denoted by the unknown rotation matrix \mathbf{R} and the unknown translation vector \mathbf{t} , both with respect the first view frame. The epipolar geometry can be then defined by the essential matrix \mathbf{E} such that

$$\lambda_1 \mathbf{x}_1 = \lambda_2 \mathbf{R} \mathbf{x}_2 + \mathbf{t}$$

$$\lambda_1 \Omega \mathbf{x}_1 = \lambda_2 \Omega \mathbf{R} \mathbf{x}_2$$

$$0 = \lambda_2 \mathbf{x}_1^\top \Omega \mathbf{R} \mathbf{x}_2$$

$$0 = \mathbf{x}_1^\top \Omega \mathbf{R} \mathbf{x}_2$$

$$0 = \mathbf{x}_1^\top \mathbf{E} \mathbf{x}_2$$

where the positive scale factors λ_i represents the depth of the of the 3-dimensional point associated to the (back-projected) ray \mathbf{x}_i . As we observe, the epipolar constraint is linear with respect the essential matrix \mathbf{E} , thus we re-write the above expression as the following linear system

$$\mathbf{x}_1^\top \mathbf{E} \mathbf{x}_2 = \mathbf{x}_1^\top \sum_{i=1}^3 x_{2,i} \mathbf{E}_i \tag{5.1}$$

$$= x_{2,1} \mathbf{x}_1^\top \mathbf{E}_1 + x_{2,2} \mathbf{x}_1^\top \mathbf{E}_2 + x_{2,3} \mathbf{x}_1^\top \mathbf{E}_3 \tag{5.2}$$

$$= \text{vec}^\top \left\{ \begin{pmatrix} x_{2,1} \mathbf{x}_1 & x_{2,2} \mathbf{x}_1 & x_{2,3} \mathbf{x}_1 \end{pmatrix} \right\} \text{vec} \{ \mathbf{E} \} \tag{5.3}$$

$$= \text{vec}^\top \left\{ \mathbf{x}_1 \mathbf{x}_2^\top \right\} \text{vec} \{ \mathbf{E} \} \tag{5.4}$$

One important property of the essential matrices is given by the characterization theorem, which states that any essential matrix has one zero singular value and two coincident non-zero singular values. This property can be easily proved by recalling that any (3×3) skew-symmetric matrix

$$\mathbf{A} = \mathbf{U}(\lambda\mathbf{Z})\mathbf{U}^\top = \mathbf{U} \underbrace{\begin{pmatrix} 0 & -\lambda & 0 \\ \lambda & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}}_{\lambda\mathbf{Z}} \mathbf{U}^\top$$

can be decomposed as

$$\mathbf{U} \underbrace{\begin{pmatrix} 0 & -\lambda & 0 \\ \lambda & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}}_{\lambda\mathbf{Z}} \mathbf{U}^\top = \mathbf{U} \underbrace{\begin{pmatrix} \lambda & 0 & 0 \\ 0 & \lambda & 0 \\ 0 & 0 & 0 \end{pmatrix}}_{\lambda\mathbf{E}_0} \underbrace{\begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}}_{\mathbf{W}^\top} \mathbf{U}^\top = \mathbf{U}(\lambda\mathbf{E}_0\mathbf{W}^\top)\mathbf{U}^\top$$

with

$$\lambda = \frac{\|\mathbf{A}\|^2}{2}$$

where $\|\mathbf{A}\|$ denotes the Frobenius norm and $\pm\lambda i$ are the nonzero eigenvalues of the skew-symmetric matrix \mathbf{A} . Based on such decomposition, we readily obtain that

$$\mathbf{E} = \mathbf{U}(\lambda\mathbf{E}_0)(\mathbf{W}\mathbf{U}^\top\mathbf{R})$$

which equals the above stated singular value decomposition of the essential matrix. This proof gives us a way to compute the relative translation and rotation associated to a given normalized essential matrix, that is such that $\lambda=1$, or $\|\mathbf{E}\|=\|\Omega\|=\sqrt{2}$ or equivalently $\|\mathbf{t}\|=1$

$$\mathbf{E} = \mathbf{U}\mathbf{E}_0\mathbf{V}^\top = \mathbf{U}(\mathbf{Z}\mathbf{W})\mathbf{V}^\top = (\mathbf{U}\mathbf{Z}\mathbf{U}^\top)(\mathbf{U}\mathbf{W}\mathbf{V}^\top) = \Omega\mathbf{R}$$

Note that the above solution of the relative pose is not unique, because we still have a sign ambiguity. Indeed, there are two possible translation candidates $\pm\mathbf{U}\mathbf{Z}\mathbf{U}^\top$ together with two rotation candidates $\mathbf{U}\mathbf{W}\mathbf{V}^\top$ and $\mathbf{U}\mathbf{W}^\top\mathbf{V}^\top$, for a total of four different pose hypotheses. The true hypothesis can be identified by testing the sign of the depth of the imaged points, known as the chirality test.

In general, the singular value decomposition does not provide rotation matrices but unitary ones, i.e., with determinant ± 1 , however we can freely flip the sign of

the last row or column of the matrices \mathbf{U} and \mathbf{V} without altering the value of \mathbf{E} , to obtain a decomposition based on two proper rotation matrices belonging to the special group $SO(3)$. With this observation in mind, we note that any normalized essential matrix \mathbf{E} belongs to the following non-linear subset of $\mathbb{R}^{3 \times 3}$, known as the *essential manifold*

$$\mathcal{E} = \left\{ \Omega \mathbf{R} \mid \Omega \in \mathfrak{so}(3), \mathbf{R} \in SO(3), \|\Omega\|^2 \equiv \text{tr}\{\Omega^\top \Omega\} = 2 \right\} \quad (5.5)$$

$$\mathcal{E} = \left\{ \mathbf{U} \mathbf{E}_0 \mathbf{V}^\top \mid \mathbf{U}, \mathbf{V} \in SO(3) \right\} \quad (5.6)$$

Based on (5.5), it is clear that any matrix $\mathbf{E} \in \mathcal{E}$ has only five degrees of freedom, while with (5.6) seems to have six instead of five. However, this is not the case because the pair of matrices (\mathbf{U}, \mathbf{V}) that gives rise to the essential matrix \mathbf{E} is not unique, there is a complete equivalence class of elements in $SO(3) \times SO(3)$ defining such matrix. We can identify the underlying equivalence relation by specifying the structure of the matrix $\mathbf{G} \in SO(3)$ such that $\mathbf{E}(\mathbf{U}, \mathbf{V}) = \mathbf{E}(\mathbf{U}\mathbf{G}, \mathbf{V}\mathbf{G})$, that is

$$\mathbf{U} \mathbf{E}_0 \mathbf{V}^\top = \mathbf{U} \mathbf{G} \mathbf{E}_0 \mathbf{G}^\top \mathbf{V}^\top \implies \mathbf{E}_0 = \mathbf{G} \mathbf{E}_0 \mathbf{G}^\top \implies \mathbf{G} = \begin{pmatrix} \Gamma & 0 \\ 0 & \det\{\Gamma\} \end{pmatrix} \quad (5.7)$$

with $\Gamma \in O(2)$. In other words, \mathbf{G} is a pure rotation about the z axis, a transformation with a single degree of freedom. Therefore, the equivalence class of (\mathbf{U}, \mathbf{V}) is the set

$$\left\{ (\mathbf{A}, \mathbf{B}) \in SO(3) \times SO(3) \mid (\mathbf{A}, \mathbf{B}) = (\mathbf{U}\mathbf{G}, \mathbf{V}\mathbf{G}), \mathbf{G} \text{ given by (5.7)} \right\} \quad (5.8)$$

5.2.2 Tangent space of the essential manifold \mathcal{E}

Given a smooth curve

$$\gamma(t) = \mathbf{U}(t) \mathbf{E}_0 \mathbf{V}^\top(t)$$

on \mathcal{E} , starting at $\mathbf{E}(0) = \mathbf{U} \mathbf{E}_0 \mathbf{V}^\top$, its tangent vector at time $t \in \mathbb{R}^+$ is defined by¹

$$\dot{\gamma} = \frac{d}{dt} \left\{ \mathbf{U} \mathbf{E}_0 \mathbf{V}^\top \right\} = \dot{\mathbf{U}} \mathbf{E}_0 \mathbf{V}^\top + \mathbf{U} \mathbf{E}_0 \dot{\mathbf{V}}^\top = \Omega_{\mathbf{U}} \mathbf{U} \mathbf{E}_0 \mathbf{V}^\top - \mathbf{U} \mathbf{E}_0 \mathbf{V}^\top \Omega_{\mathbf{V}}$$

where $\Omega_{\mathbf{U}}$ and $\Omega_{\mathbf{V}}$ are skew-symmetric matrices.

Let's suppose now that the pair (\mathbf{U}, \mathbf{V}) is given by the smooth mapping $\varphi(\tilde{\mathbf{U}}, \tilde{\mathbf{V}})$,

¹The explicit dependence on time will be omitted to improve the readability of the formulae contained in the present paragraph.

with $(\tilde{\mathbf{U}}, \tilde{\mathbf{V}})$ local coordinates in $SO(3) \times SO(3)$, centered at the point (\mathbf{U}, \mathbf{V})

$$\varphi(\tilde{\mathbf{U}}, \tilde{\mathbf{V}}) = (\mathbf{U}\tilde{\mathbf{U}}, \mathbf{V}\tilde{\mathbf{V}}) \implies \varphi(\mathbf{I}, \mathbf{I}) = (\mathbf{U}, \mathbf{V}) \quad (5.9)$$

Using the new coordinates (5.9) the tangent vector $\dot{\gamma}$ assumes the following form

$$\dot{\gamma} = \frac{d}{dt} \left\{ \mathbf{U}\tilde{\mathbf{U}}\mathbf{E}_0\tilde{\mathbf{V}}^\top\mathbf{V}^\top \right\} = \mathbf{U}[\Omega_{\tilde{\mathbf{U}}}\tilde{\mathbf{U}}\mathbf{E}_0\tilde{\mathbf{V}}^\top - \tilde{\mathbf{U}}\mathbf{E}_0\tilde{\mathbf{V}}^\top\Omega_{\tilde{\mathbf{V}}}] \mathbf{V}^\top$$

at time $t=0$ we have that $(\tilde{\mathbf{U}}(0), \tilde{\mathbf{V}}(0)) = (\mathbf{I}, \mathbf{I})$, then

$$\dot{\gamma}(0) = \left. \frac{d}{dt} \left\{ \mathbf{U}\tilde{\mathbf{U}}\mathbf{E}_0\tilde{\mathbf{V}}^\top\mathbf{V}^\top \right\} \right|_{t=0} = \mathbf{U}[\Omega_{\tilde{\mathbf{U}}}(0)\mathbf{E}_0 - \mathbf{E}_0\Omega_{\tilde{\mathbf{V}}}(0)] \mathbf{V}^\top$$

As consequence, the tangent space $\mathcal{T}_{\mathbf{E}}\mathcal{E}$ at \mathbf{E} of the essential manifold \mathcal{E} , with respect to a local coordinates chart centered at \mathbf{E} , is the set of all tangent vectors $\dot{\gamma}(0)$ such that

$$\mathcal{T}_{\mathbf{E}}\mathcal{E} = \left\{ \mathbf{U}[\Omega_{\mathbf{U}}\mathbf{E}_0 - \mathbf{E}_0\Omega_{\mathbf{V}}] \mathbf{V}^\top \mid \Omega_{\mathbf{U}}, \Omega_{\mathbf{V}} \in \mathfrak{so}(3) \right\} \quad (5.10)$$

To conclude, let's assume that $\Omega_{\mathbf{U}} = [\omega]_{\times}$ and $\Omega_{\mathbf{V}} = [\psi]_{\times}$, with $\omega, \psi \in \mathbb{R}^3$. Then we have that

$$\mathbf{U}[\Omega_{\mathbf{U}}\mathbf{E}_0 - \mathbf{E}_0\Omega_{\mathbf{V}}] \mathbf{V}^\top = \mathbf{U} \begin{pmatrix} 0 & \psi_3 - \omega_3 & -\psi_2 \\ \omega_3 - \psi_3 & 0 & \psi_1 \\ -\omega_2 & \omega_1 & 0 \end{pmatrix} \mathbf{V}^\top$$

which implies that $\omega = \psi = (0, 0, x)^\top$, $x \in \mathbb{R}$, spans the null space of the tangent space $\mathcal{T}_{\mathbf{E}}\mathcal{E}$. This is an expected result, since it reflects the equivalence relation defined in (5.8). This imply that any vector field assigning to some point \mathbf{E} of \mathcal{E} a tangent vector lying on its null space, then such vector field will emanates no flow from \mathbf{E} .

5.2.3 Essential manifold parameterization

A smooth local parametrization, or chart, of an open subset \mathcal{S} of the smooth n -dimensional manifold \mathcal{M} is defined by a diffeomorphic map between some open subset \mathcal{U} of \mathbb{R}^n centered at the origin (e.g. a linear space) and \mathcal{S} . Since any element \mathbf{E} of the essential manifold \mathcal{E} have (at most) five different degrees of freedom, any small open $\mathcal{S} \subset \mathcal{E}$ is diffeomorphic to some small open $\mathcal{U} \subset \mathbb{R}^5$. Let assume one more time that the pair (\mathbf{U}, \mathbf{V}) is given by the aforementioned smooth mapping $\varphi(\tilde{\mathbf{U}}, \tilde{\mathbf{V}})$.

Exploiting the global exponential parametrization of the elements of $\text{SO}(3)$, we can substitute the pair $(\tilde{\mathbf{U}}, \tilde{\mathbf{V}})$ by $(\exp \Omega_{\mathbf{U}}(t), \exp \Omega_{\mathbf{V}}(t))$ with $\Omega_{\mathbf{U}}(t)$ and $\Omega_{\mathbf{V}}(t)$ elements of $\mathfrak{so}(3)$, leading to the following parametrization of the essential matrix \mathbf{E}

$$\mathbf{E}(t) = \mathbf{U} e^{\Omega_{\mathbf{U}}(t)} \mathbf{E}_0 e^{\Omega_{\mathbf{V}}(t)} \mathbf{V}^\top \quad (5.11)$$

However, we cannot yet guarantee that such mapping is diffeomorphic between an small open subset of \mathcal{E} containing \mathbf{E} and some small open subset of \mathbb{R}^5 . In fact, we need specify the structure of the skew-symmetric exponents. By thinking $\mathbf{E}(t)$ as a one parameter flow emanated from \mathbf{E} , we implicitly assume that the vector field generating the flow does not assign, at \mathbf{E} , a vector lying in the kernel of $\mathcal{T}_{\mathbf{E}}\mathcal{E}$. Thus, by simple computations, we obtain that a smooth local parametrization of some small open $\mathcal{S} \subset \mathcal{E}$ centered at $\mathbf{E} \in \mathcal{S}$, is given by (5.11) with

$$\begin{aligned} \Omega_{\mathbf{U}}(t) &= \left[\begin{array}{c} x_1(t) \\ x_2(t) \\ x_3(t) \end{array} \right]_{\times} \\ \Omega_{\mathbf{V}}(t) &= \left[\begin{array}{c} x_4(t) \\ x_5(t) \\ -x_3(t) \end{array} \right]_{\times} \end{aligned} \quad (5.12)$$

where $x_i(t)$, $i=1, \dots, 5$, are the coordinates of parametric point $\mathbf{x}(t) \in \mathcal{U}$, where \mathcal{U} is a small open subset of \mathbb{R}^5 containing the origin. Note that the time parametrization is not strictly necessary to define the mapping $\mathcal{U} \rightarrow \mathcal{S}$, however offers a clear, intuitive and geometric interpretation of the parametrization.

5.2.4 Optimization on the essential manifold

Based on the linearity of the epipolar constraint with respect the entries of the essential matrix \mathbf{E} (see Eq. 5.4), given a set of N point correspondences $\mathbf{x}_1^{(i)} \leftrightarrow \mathbf{x}_2^{(i)}$ between images \mathcal{I}_1 and \mathcal{I}_2 , we define our objective function as the following quadratic expression

$$\begin{aligned} f(\mathbf{E}) &= \frac{1}{2N} \sum_{i=1}^N \left(\text{vec}^\top \{ \mathbf{M}^{(i)} \} \text{vec} \{ \mathbf{E} \} \right)^2 \\ &= \frac{1}{2} \text{vec}^\top \{ \mathbf{E} \} \left(\frac{1}{N} \sum_{i=1}^N \text{vec} \{ \mathbf{M}^{(i)} \} \text{vec}^\top \{ \mathbf{M}^{(i)} \} \right) \text{vec} \{ \mathbf{E} \} \\ &= \frac{1}{2} \text{vec}^\top \{ \mathbf{E} \} \mathcal{M} \text{vec} \{ \mathbf{E} \} \end{aligned}$$

$$= \frac{1}{2} \left\| \text{vec}\{\mathbf{E}\} \right\|_{\mathcal{M}}^2$$

where $\mathbf{M}^{(i)} = \mathbf{x}_1^{(i)} \mathbf{x}_2^{(i)\top}$ is the compact representation of the i -th point correspondence. We note that the definition of the data matrix \mathcal{M} is possible only under the assumption that the set of point correspondences is free of outliers.

The geometric optimization scheme introduced by Helmke et al. [51] is based on the minimization of $f(\mathbf{E})$ using interleaved iterations of the Newton and Gauss-Newton methods over the essential manifold \mathcal{E} . The algorithm is derived from the second order Taylor approximation of the cost $f(\mathbf{E})$ in terms of the parametrization vector $\mathbf{x} \in \mathcal{U}$, $\mathcal{U} \subset \mathbb{R}^5$, defined by equations (5.11) and (5.12)

$$\begin{aligned} f \circ \varphi(\mathbf{x}) &= \frac{1}{2} \left\| \text{vec}\left\{ \mathbf{U} e^{\Omega_{\mathbf{U}}} \mathbf{E}_0 e^{\Omega_{\mathbf{V}}} \mathbf{V}^{\top} \right\} \right\|_{\mathcal{M}}^2 \\ &= \frac{1}{2} \left\| (\mathbf{V} \otimes \mathbf{U}) \text{vec}\left\{ e^{\Omega_{\mathbf{U}}} \mathbf{E}_0 e^{\Omega_{\mathbf{V}}} \right\} \right\|_{\mathcal{M}}^2 \end{aligned}$$

The second order Taylor expansion of $f(\mathbf{x})$ around zero is given by

$$f \circ \varphi(\mathbf{x}) \approx f \circ \varphi(0) + \nabla^{\top}(f \circ \varphi)(0) \mathbf{x} + \mathbf{x}^{\top} \mathbf{H}_{f \circ \varphi}(0) \mathbf{x}$$

where

$$\begin{aligned} \nabla^{\top}(f \circ \varphi)(0) \mathbf{x} &= \text{vec}^{\top}\{\mathbf{E}\} \mathcal{M} (\mathbf{V} \otimes \mathbf{U}) \text{vec}\{\Omega_{\mathbf{U}} \mathbf{E}_0 - \mathbf{E}_0 \Omega_{\mathbf{V}}\} \\ &= \text{vec}^{\top}\{\mathbf{E}\} \mathcal{M} (\mathbf{V} \otimes \mathbf{U}) \left[\begin{array}{cc} (\mathbf{E}_0 \otimes \mathbf{I}) \underline{\text{vec}\{\Omega_{\mathbf{U}}\}} & - (\mathbf{I} \otimes \mathbf{E}_0) \underline{\text{vec}\{\Omega_{\mathbf{V}}\}} \\ \text{vec}\{\Omega_{\mathbf{U}}\} \triangleq \mathbf{Q}_{\mathbf{U}} \mathbf{x} & \text{vec}\{\Omega_{\mathbf{V}}\} \triangleq \mathbf{Q}_{\mathbf{V}} \mathbf{x} \end{array} \right] \\ &= \text{vec}^{\top}\{\mathbf{E}\} \mathcal{M} (\mathbf{V} \otimes \mathbf{U}) \left[\underline{(\mathbf{E}_0 \otimes \mathbf{I}) \mathbf{Q}_{\mathbf{U}} - (\mathbf{I} \otimes \mathbf{E}_0) \mathbf{Q}_{\mathbf{V}}} \right] \mathbf{x} \\ &\quad \underline{\mathbf{J} \triangleq (\mathbf{V} \otimes \mathbf{U}) [(\mathbf{E}_0 \otimes \mathbf{I}) \mathbf{Q}_{\mathbf{U}} - (\mathbf{I} \otimes \mathbf{E}_0) \mathbf{Q}_{\mathbf{V}}]} \\ &\triangleq \text{vec}^{\top}\{\mathbf{E}\} \mathcal{M} \mathbf{J} \mathbf{x} \end{aligned}$$

and, by analogous computations,

$$\mathbf{H}_{f \circ \varphi}(0) = \mathbf{J}^{\top} \mathcal{M} \mathbf{J} + \begin{pmatrix} \mathbf{Q}_{\mathbf{U}}^{\top} & \mathbf{Q}_{\mathbf{V}}^{\top} \end{pmatrix} \begin{pmatrix} -(\mathbf{D} \mathbf{E}_0 \otimes \mathbf{I}) & (\mathbf{D} \otimes \mathbf{E}_0) \\ (\mathbf{D}^{\top} \otimes \mathbf{E}_0) & -(\mathbf{E}_0 \mathbf{D} \otimes \mathbf{I}) \end{pmatrix} \begin{pmatrix} \mathbf{Q}_{\mathbf{U}} \\ \mathbf{Q}_{\mathbf{V}} \end{pmatrix}$$

with

$$\text{vec}\{\mathbf{D}\} = (\mathbf{V}^{\top} \otimes \mathbf{U}^{\top}) \mathcal{M} \text{vec}\{\mathbf{E}\}$$

The Newton and Gauss-Newton directions for optimizing $f \circ \varphi(\mathbf{x})$, respectively \mathbf{x}_N^* and \mathbf{x}_{GN}^* , are computed as the solution of following linear systems

$$-\nabla^\top(f \circ \varphi)(0) = \mathbf{H}_{f \circ \varphi}(0) \mathbf{x}_N^* \quad (5.13)$$

$$-\nabla^\top(f \circ \varphi)(0) = \mathbf{J}^\top \mathcal{M} \mathbf{J} \mathbf{x}_{GN}^* \quad (5.14)$$

The complete algorithm apply the Newton method on a sufficiently small neighborhood of the set of local minima of $f(\mathbf{x})$, where the Hessian $\mathbf{H}_{f \circ \varphi}(0)$ is full rank. Outside of such a neighborhood, where the Hessian matrix may become either nearly degenerate or indefinite, the algorithm switches to Gauss-Newton iterations, thus enlarging the domain of attraction of the local minima. At each optimization step k , matrices \mathbf{U}^{k+1} and \mathbf{V}^{k+1} are computed incrementally by means of the exponential parametrization introduced in (5.11), using the optimal parameters \mathbf{x}^* calculated either with the Newton or Gauss-Newton scheme

$$\begin{aligned} \mathbf{U}^{k+1} &= \mathbf{U}^k e^{\Omega_{\mathbf{U}}^*} \\ \mathbf{V}^{k+1} &= \mathbf{V}^k e^{\Omega_{\mathbf{V}}^*} \end{aligned} \quad (5.15)$$

The geometric optimization of the essential matrix is performed until the norm of the gradient $\nabla^\top(f \circ \varphi)(0)$ become smaller than some predefined small threshold.

5.3 Coplanarity constraint

5.3.1 Mixed algebraic matrix constraints

Suppose now that we have three different views instead of two, and that some initial estimate of the pair-wise stereo geometry between views have been already computed, i.e. the essential matrices \mathbf{E}_{12} , \mathbf{E}_{23} and \mathbf{E}_{31} ² are known. Moreover, let also assume that the corresponding camera matrices have been previously disambiguated. Such camera poses can be represented by the following homogeneous transformations³

$$\begin{aligned} \mathbf{M}_{12} &= \begin{pmatrix} \mathbf{R}_{12} & \lambda_1 \mathbf{t}_{12} \\ \mathbf{0}^\top & 1 \end{pmatrix} & \mathbf{R}_{12} &= \mathbf{U}_{12} \mathbf{W}_{12} \mathbf{V}_{12}^\top \\ \mathbf{M}_{23} &= \begin{pmatrix} \mathbf{R}_{23} & \lambda_2 \mathbf{t}_{23} \\ \mathbf{0}^\top & 1 \end{pmatrix} & \mathbf{R}_{23} &= \mathbf{U}_{23} \mathbf{W}_{23} \mathbf{V}_{23}^\top \end{aligned}$$

² \mathbf{E}_{ij} represents the essential matrix between views i and j , with i as the reference view.

³ \mathbf{M}_{ij} represents the homogeneous transformation from view j to view i with respect frame i .

$$\mathbf{M}_{31} = \begin{pmatrix} \mathbf{R}_{31} & \lambda_3 \mathbf{t}_{31} \\ \mathbf{0}^\top & 1 \end{pmatrix} \quad \mathbf{R}_{31} = \mathbf{U}_{31} \mathbf{W}_{31} \mathbf{V}_{31}^\top$$

where \mathbf{W}_{ij} can be either \mathbf{W} or \mathbf{W}^\top (defined at Section 5.2.1) and $\lambda_i \in \mathbb{R}_{\neq 0}$ are the unknown lengths of the unitary translations $\mathbf{t}_{ij} = \mathbf{U}_{ij} \mathbf{z}$, with \mathbf{z} representing the unitary vector along the z -axis.

As we know, any triplet of points defines a plane, conversely, all lines joining a triplet of points are coplanar. In the case of three views, we have that the set of lines connecting the (camera) centers are coplanar. With respect to the first view frame, this condition can be written as the triple product

$$\left(\mathbf{R}_{31}^\top \mathbf{t}_{31} \right)^\top \cdot (\mathbf{t}_{12} \times \mathbf{R}_{12} \mathbf{t}_{23}) = 0$$

which, in terms of the essential matrix parametrization, can be expressed as

$$\begin{aligned} 0 &= \left(\mathbf{R}_{31}^\top \mathbf{t}_{31} \right)^\top [\mathbf{t}_{12}]_\times \mathbf{R}_{12} \mathbf{t}_{23} \\ &= \mathbf{t}_{31}^\top \mathbf{R}_{31} [\mathbf{t}_{12}]_\times \mathbf{R}_{12} \mathbf{t}_{23} \\ &= \left(\mathbf{z}^\top \mathbf{U}_{31}^\top \right) \left(\mathbf{U}_{31} \mathbf{W}_{31} \mathbf{V}_{31}^\top \right) [\mathbf{U}_{12} \mathbf{z}]_\times \left(\mathbf{U}_{12} \mathbf{W}_{12} \mathbf{V}_{12}^\top \right) (\mathbf{U}_{23} \mathbf{z}) \\ &= \left(\mathbf{z}^\top \mathbf{U}_{31}^\top \right) \left(\mathbf{U}_{31} \mathbf{W}_{31} \mathbf{V}_{31}^\top \right) \left(\mathbf{U}_{12} [\mathbf{z}]_\times \mathbf{U}_{12}^\top \right) \left(\mathbf{U}_{12} \mathbf{W}_{12} \mathbf{V}_{12}^\top \right) (\mathbf{U}_{23} \mathbf{z}) \\ &= \mathbf{z}^\top \mathbf{W}_{31} \mathbf{V}_{31}^\top \mathbf{U}_{12} [\mathbf{z}]_\times \mathbf{W}_{12} \mathbf{V}_{12}^\top \mathbf{U}_{23} \mathbf{z} \\ &= \mathbf{z}^\top \left[\mathbf{V}_{31}^\top \mathbf{U}_{12} (\pm \mathbf{E}_0) \mathbf{V}_{12}^\top \mathbf{U}_{23} \right] \mathbf{z} \end{aligned} \quad (5.16)$$

where $\mathbf{E}_0 = \text{diag}\{1, 1, 0\}$ as defined in Section 5.2.1. Following analogous steps, we can find equivalent expressions for the same coplanarity constraint with respect the second and third views

$$0 = \mathbf{z}^\top \left[\mathbf{V}_{12}^\top \mathbf{U}_{23} (\pm \mathbf{E}_0) \mathbf{V}_{23}^\top \mathbf{U}_{31} \right] \mathbf{z} \quad (5.17)$$

$$0 = \mathbf{z}^\top \left[\mathbf{V}_{23}^\top \mathbf{U}_{31} (\pm \mathbf{E}_0) \mathbf{V}_{31}^\top \mathbf{U}_{12} \right] \mathbf{z} \quad (5.18)$$

Now, defining the matrices

$$\mathbf{A} = \mathbf{V}_{31}^\top \mathbf{U}_{12} = \begin{pmatrix} \mathbf{A}_{22} & \mathbf{a} \\ \boldsymbol{\alpha}^\top & a \end{pmatrix}$$

$$\mathbf{B} = \mathbf{V}_{12}^\top \mathbf{U}_{23} = \begin{pmatrix} \mathbf{B}_{22} & \mathbf{b} \\ \boldsymbol{\beta}^\top & b \end{pmatrix}$$

$$\mathbf{C} = \mathbf{V}_{23}^\top \mathbf{U}_{31} = \begin{pmatrix} \mathbf{C}_{22} & \mathbf{c} \\ \boldsymbol{\gamma}^\top & c \end{pmatrix}$$

we can compactly rewrite equations (5.16), (5.17) and (5.18) as

$$\begin{aligned} 0 &= \mathbf{z}^\top [\mathbf{A}(\pm \mathbf{E}_0) \mathbf{B}] \mathbf{z} \\ 0 &= \mathbf{z}^\top [\mathbf{B}(\pm \mathbf{E}_0) \mathbf{C}] \mathbf{z} \\ 0 &= \mathbf{z}^\top [\mathbf{C}(\pm \mathbf{E}_0) \mathbf{A}] \mathbf{z} \end{aligned} \tag{5.19}$$

Simple algebraic manipulations allow to show that the coplanarity constraint with respect the first view implies that

$$0 = \pm \boldsymbol{\alpha}^\top \mathbf{b} \quad \implies \quad \boldsymbol{\alpha} \perp \mathbf{b}$$

Analogously, with respect the remaining views, we also obtain that

$$\begin{aligned} \boldsymbol{\beta} &\perp \mathbf{c} \\ \boldsymbol{\gamma} &\perp \mathbf{a} \end{aligned}$$

as consequence, the camera motion between three views does not lie on the product manifold $\mathcal{E}_{12} \times \mathcal{E}_{23} \times \mathcal{E}_{31}$, given by the pair-wise essential manifolds, but on a constrained sub-manifold of it. Our goal is to understand how the above orthogonality conditions can be exploited so as to characterize and possibly parametrize such manifold.

5.3.2 Algebraic insights

The on-manifold optimization algorithm proposed in [51] depends on the parametrization vector $\mathbf{x}(t)$, belonging to the subspace $\mathcal{U} \subset \mathbb{R}^5$ diffeomorphic to the tangent space $\mathcal{T}_{\mathbf{E}} \mathcal{E}$, therefore we need to derive an alternative parametric expression describing the above perpendicularity constraints that can be effectively used for our minimization purposes. In particular, we observe that the orthogonality conditions $\boldsymbol{\alpha} \perp \mathbf{b}$, $\boldsymbol{\beta} \perp \mathbf{c}$ and $\boldsymbol{\gamma} \perp \mathbf{a}$ are holonomic, therefore, we require an equivalent integrable differential form to be inserted inside the optimization scheme.

We start by considering the \mathbf{A} , \mathbf{B} and \mathbf{C} matrices, instead of considering the underlying rotation matrices \mathbf{U}_{ij} and \mathbf{V}_{ij} used to compute them. Let represent them in terms of the ZYZ Euler angles, that is, as the composition of three elementary

rotations of the form

$$\begin{aligned}
\mathbf{A} &= \mathbf{R}_z(\phi_{\mathbf{A}})\mathbf{R}_y(\theta_{\mathbf{A}})\mathbf{R}_z(\psi_{\mathbf{A}}) \\
\mathbf{B} &= \mathbf{R}_z(\phi_{\mathbf{B}})\mathbf{R}_y(\theta_{\mathbf{B}})\mathbf{R}_z(\psi_{\mathbf{B}}) \\
\mathbf{C} &= \mathbf{R}_z(\phi_{\mathbf{C}})\mathbf{R}_y(\theta_{\mathbf{C}})\mathbf{R}_z(\psi_{\mathbf{C}})
\end{aligned} \tag{5.20}$$

Let's now substitute (5.20) into equations (5.19). After few algebraic manipulations, we obtain

$$0 = \sin(\theta_{\mathbf{A}}) \cos(\psi_{\mathbf{A}} + \phi_{\mathbf{B}}) \sin(\theta_{\mathbf{B}}) \tag{5.21}$$

$$0 = \sin(\theta_{\mathbf{B}}) \cos(\psi_{\mathbf{B}} + \phi_{\mathbf{C}}) \sin(\theta_{\mathbf{C}}) \tag{5.22}$$

$$0 = \sin(\theta_{\mathbf{C}}) \cos(\psi_{\mathbf{C}} + \phi_{\mathbf{A}}) \sin(\theta_{\mathbf{A}}) \tag{5.23}$$

Equation (5.21) is satisfied whenever

$$\theta_{\mathbf{A}} = \frac{k\pi}{2} \quad \text{or} \quad \theta_{\mathbf{B}} = \frac{k\pi}{2} \quad \text{or} \quad \psi_{\mathbf{A}} + \phi_{\mathbf{B}} = \frac{k\pi}{2}$$

for any $k \in \mathbb{Z}_{\neq 0}$. Since all angular quantities defining \mathbf{A} and \mathbf{B} as the composition of elementary rotations are arbitrary, the only feasible constraint is given by the mixed term $\psi_{\mathbf{A}} + \phi_{\mathbf{B}} = \frac{k\pi}{2}$. Extending this result to equations (5.22) and (5.23), we finally obtain that

$$\begin{aligned}
\psi_{\mathbf{A}} + \phi_{\mathbf{B}} &= \frac{k\pi}{2} \\
\psi_{\mathbf{B}} + \phi_{\mathbf{C}} &= \frac{k\pi}{2} \\
\psi_{\mathbf{C}} + \phi_{\mathbf{A}} &= \frac{k\pi}{2}
\end{aligned}$$

This is an interesting result, since the spatial coplanarity constraint has been translated into a set of angular constraints. However, using this representation we cannot be directly translate the constraint in terms of the underlying \mathbf{U}_{ij} and \mathbf{V}_{ij} matrices.

Let now recall the orthogonality conditions (5.16), (5.17) and (5.18). To improve the readability of the paragraph we relax the notation of matrices \mathbf{M}_{ij} as \mathbf{M}_i , also we identify the k row vector of \mathbf{M}_i as ${}^i\mathbf{m}^k$ and the l column vector as ${}^i\mathbf{m}_l$. The

orthogonality conditions after some manipulations can be written as

$$\begin{aligned}
\begin{pmatrix} {}^1\mathbf{u}_1^\top \\ {}^1\mathbf{u}_2^\top \end{pmatrix} {}^3\mathbf{v}_3 &= \pm \begin{pmatrix} -{}^1\mathbf{v}_2^\top \\ {}^1\mathbf{v}_1^\top \end{pmatrix} {}^2\mathbf{u}_3 \\
\begin{pmatrix} {}^2\mathbf{u}_1^\top \\ {}^2\mathbf{u}_2^\top \end{pmatrix} {}^1\mathbf{v}_3 &= \pm \begin{pmatrix} -{}^2\mathbf{v}_2^\top \\ {}^2\mathbf{v}_1^\top \end{pmatrix} {}^3\mathbf{u}_3 \\
\begin{pmatrix} {}^3\mathbf{u}_1^\top \\ {}^3\mathbf{u}_2^\top \end{pmatrix} {}^2\mathbf{v}_3 &= \pm \begin{pmatrix} -{}^3\mathbf{v}_2^\top \\ {}^3\mathbf{v}_1^\top \end{pmatrix} {}^1\mathbf{u}_3
\end{aligned} \tag{5.24}$$

This set of equations reflects that the orthogonality condition constraints the columns and rows of the matrices \mathbf{U}_{ij} and \mathbf{V}_{ij} to satisfy a set of bilinear relations.

5.4 Constrained geometric optimization

5.4.1 Three-view soft-constrained optimization

Following the same notation and assumptions introduced at the beginning of the previous Section, in the case of three views, we can refine each pair-wise stereo geometry through the unconstrained optimization scheme defined by Helmke et al. [51]. However, by simple algebraic manipulations, we can define a single parameter space $\mathcal{U} \subset \mathbb{R}^{15}$, by stacking together the three independent pair-wise parameters \mathbf{x}_{12} , \mathbf{x}_{23} and \mathbf{x}_{31} , and a single cost function $f(\mathbf{x})$ such that all stereo geometries are independently refined. We will denote this problem as the *unconstrained* three view optimization. Again, to improve the readability of the paragraph we relax the notation of matrices \mathbf{M}_{ij} as \mathbf{M}_i and vectors \mathbf{x}_{ij} as \mathbf{x}_i . For this unconstrained problem, the cost function $f(\mathbf{E}_1, \mathbf{E}_2, \mathbf{E}_3)$ is given as the sum of the single pair-wise stereo geometry costs

$$f(\mathbf{E}_1, \mathbf{E}_2, \mathbf{E}_3) = \sum_{i=1}^3 \text{vec}^\top\{\mathbf{E}_i\} \mathcal{M}_i \text{vec}\{\mathbf{E}_i\} \tag{5.25}$$

Let now stack all parameters into the vector $\mathbf{x} = (\mathbf{x}_1^\top \ \mathbf{x}_2^\top \ \mathbf{x}_3^\top)^\top$ and rewrite the cost function in term of these

$$f \circ \varphi(\mathbf{x}) = \frac{1}{2} \sum_{i=1}^3 \left\| (\mathbf{V}_i \otimes \mathbf{U}_i) \text{vec}\left\{ e^{\Omega_{v_i}} \mathbf{E}_0 e^{\Omega_{v_i}} \right\} \right\|_{\mathcal{M}_i}^2 \tag{5.26}$$

with

$$\begin{aligned}\Omega_{U_1} &= \left[\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \right]_{\times} & \Omega_{V_1} &= \left[\begin{pmatrix} x_4 \\ x_5 \\ -x_3 \end{pmatrix} \right]_{\times} \\ \Omega_{U_2} &= \left[\begin{pmatrix} x_6 \\ x_7 \\ x_8 \end{pmatrix} \right]_{\times} & \Omega_{V_2} &= \left[\begin{pmatrix} x_9 \\ x_{10} \\ -x_8 \end{pmatrix} \right]_{\times} \\ \Omega_{U_3} &= \left[\begin{pmatrix} x_{11} \\ x_{12} \\ x_{13} \end{pmatrix} \right]_{\times} & \Omega_{V_3} &= \left[\begin{pmatrix} x_{14} \\ x_{15} \\ -x_{13} \end{pmatrix} \right]_{\times}\end{aligned}$$

Following analogous algebraic computations as those presented on the Section 5.2.4, it is easily to show that the compound gradient of this unconstrained problem correspond to the stack of the single gradients, while the Hessian corresponds to a block diagonal matrix with diagonal elements equal to the single pair-wise Hessians. In formal notation we have

$$\underbrace{\nabla^{\top}(f \circ \varphi)(0)}_{15 \times 1} = \left(\underbrace{\nabla^{\top}(f_1 \circ \varphi_1)(0)}_{5 \times 1} \quad \underbrace{\nabla^{\top}(f_2 \circ \varphi_2)(0)}_{5 \times 1} \quad \underbrace{\nabla^{\top}(f_3 \circ \varphi_3)(0)}_{5 \times 1} \right) \quad (5.27)$$

$$\underbrace{H_{f \circ \varphi}(0)}_{15 \times 15} = \begin{pmatrix} \underbrace{H_{f_1 \circ \varphi_1}(0)}_{5 \times 5} & 0 & 0 \\ 0 & \underbrace{H_{f_2 \circ \varphi_2}(0)}_{5 \times 5} & 0 \\ 0 & 0 & \underbrace{H_{f_3 \circ \varphi_3}(0)}_{5 \times 5} \end{pmatrix} \quad (5.28)$$

We stress that the unconstrained optimization of the stacked problem is exactly equivalent to optimizing each pair-wise stereo geometry independently, reason why the simple block structure of the differential forms. In the remainder of the chapter, we will develop algorithm based exclusively on the Gauss-Newton scheme, therefore the computation of both gradient and Hessian depends uniquely on the definition of the first differential form.

Based on constrains (5.16), (5.17) and (5.18), we would like to define a set of regularizing terms form unconstrained problem previously described. To this end, observe that each coplanarity constraint can be written in the following vector

form

$$\mathbf{z}^\top \left[\mathbf{V}_k^\top \mathbf{U}_i (\pm \mathbf{E}_0) \mathbf{V}_i^\top \mathbf{U}_j \right] \mathbf{z} = \text{vec}^\top \{ \mathbf{z} \mathbf{z}^\top \} \left[\mathbf{V}_k^\top \mathbf{U}_i (\pm \mathbf{E}_0) \mathbf{V}_i^\top \mathbf{U}_j \right] = 0 \quad (5.29)$$

with $i \neq j \neq k$ and $i, j, k \in \{1, 2, 3\}$. Now, defining the (9×9) matrix

$$\mathbf{Z} = \text{vec} \{ \mathbf{z} \mathbf{z}^\top \} \text{vec}^\top \{ \mathbf{z} \mathbf{z}^\top \}$$

the associated cost of the constraint (5.29) correspond to

$$\begin{aligned} f_{c_i}(\mathbf{x}) &= \frac{1}{2} \left(\mathbf{z}^\top \left[\mathbf{V}_k^\top \mathbf{U}_i (\pm \mathbf{E}_0) \mathbf{V}_i^\top \mathbf{U}_j \right] \mathbf{z} \right)^2 \\ &= \frac{1}{2} \text{vec}^\top \{ \mathbf{V}_k^\top \mathbf{U}_i \mathbf{E}_0 \mathbf{V}_i^\top \mathbf{U}_j \} \mathbf{Z} \text{vec} \{ \mathbf{V}_k^\top \mathbf{U}_i \mathbf{E}_0 \mathbf{V}_i^\top \mathbf{U}_j \} \end{aligned}$$

we observe that the structure of the constraint resemble and generalize the structure of the cost function itself. In particular, the matrix

$$\mathbf{E}_{c_i} \triangleq \mathbf{V}_k^\top \mathbf{U}_i \mathbf{E}_0 \mathbf{V}_i^\top \mathbf{U}_j$$

depends on four different elements of $\text{SO}(3)$, as belongs to some manifold \mathcal{N} . The expression of the tangent space is given by

$$\mathcal{T}_{\mathbf{E}_{c_i}} \mathcal{N} = \left\{ \mathbf{E}_{c_i} \Omega_{\mathbf{U}_j} - \Omega_{\mathbf{V}_k} \mathbf{E}_{c_i} + \mathbf{V}_k^\top \mathbf{U}_i \left[\Omega_{\mathbf{U}_i} \mathbf{E}_0 - \mathbf{E}_0 \Omega_{\mathbf{V}_i} \right] \mathbf{V}_i^\top \mathbf{U}_j \mid \Omega_{\mathbf{U}_i}, \Omega_{\mathbf{U}_j}, \Omega_{\mathbf{V}_i}, \Omega_{\mathbf{V}_k} \in \mathfrak{so}(3) \right\}$$

The directional derivative and the Gauss-Newton approximation of the Hessian matrix result

$$\begin{aligned} \nabla^\top (f_{c_i} \circ \varphi)(0) &= \text{vec}^\top \{ \mathbf{E}_{c_i} \} \mathbf{Z} \text{vec} \{ \mathbf{E}_{c_i} \Omega_{\mathbf{U}_j} - \Omega_{\mathbf{V}_k} \mathbf{E}_{c_i} + \mathbf{V}_k^\top \mathbf{U}_i (\Omega_{\mathbf{U}_i} \mathbf{E}_0 - \mathbf{E}_0 \Omega_{\mathbf{V}_i}) \mathbf{V}_i^\top \mathbf{U}_j \} \\ &\triangleq \underbrace{\text{vec}^\top \{ \mathbf{E}_{c_i} \}}_{1 \times 9} \underbrace{\mathbf{Z}}_{9 \times 9} \underbrace{\mathbf{J}_{c_i}}_{9 \times 15} \underbrace{\mathbf{x}}_{15 \times 1} \end{aligned} \quad (5.30)$$

$$\mathbf{H}_{f_{c_i} \circ \varphi}(0) = \mathbf{J}_{c_i}^\top \mathbf{Z} \mathbf{J}_{c_i} \quad (5.31)$$

where

$$\mathbf{J}_{c_i} \triangleq (\mathbf{I} \otimes \mathbf{E}_{c_i}) \mathbf{Q}_{\mathbf{U}_j} - (\mathbf{E}_{c_i} \otimes \mathbf{I}) \mathbf{Q}_{\mathbf{V}_k} + (\mathbf{V}_i^\top \mathbf{U}_j \otimes \mathbf{V}_k^\top \mathbf{U}_i) \left[(\mathbf{E}_0 \otimes \mathbf{I}) \mathbf{Q}_{\mathbf{U}_i} - (\mathbf{I} \otimes \mathbf{E}_0) \mathbf{Q}_{\mathbf{V}_i} \right]$$

Algorithm 2: Soft-constrained optimization of pair-wise stereo geometry between three views

Data:

K: camera calibration matrix;
 $\mathbf{X}_{12}, \mathbf{X}_{23}, \mathbf{X}_{31}$: point correspondences between views;
 ϵ : desired accuracy;
 $\lambda_1, \lambda_2, \lambda_3$: desired regularization weights;

Result:

$\mathbf{E}_{12}, \mathbf{E}_{23}, \mathbf{E}_{31}$: pair-wise stereo geometry;

Step 0:

linear initialization of \mathbf{E}_{ij} (e.g., using the 8p [49] or Fiore [35] algorithm);
computation of \mathbf{U}_{ij} and \mathbf{V}_{ij} (i.e., projection of \mathbf{E}_{ij} into the essential manifold);

Step 1:

projection of \mathbf{U}_{ij} and \mathbf{V}_{ij} into the coplanarity constraint;

Step 2: regularized Gauss-Newton iterations;

do

 compute the unconstrained gradient (5.27) and Hessian (5.28) of $f(\mathbf{x})$;
 compute the gradients (5.30) and Hessians (5.31) of each $f_{c_i}(\mathbf{x})$;
 compute the regularized gradient $\tilde{\nabla}(f \circ \varphi)(0)$ and Hessian $\tilde{\mathbf{H}}(f \circ \varphi)(0)$;
 get the regularized Gauss-Newton optimal motion $\tilde{\mathbf{x}}_{GN}^*$;
 update \mathbf{U}_{ij} and \mathbf{V}_{ij} using the exponential map (eq. 5.15);

while $\epsilon \geq \|\tilde{\nabla}(f \circ \varphi)(0)\|$;

The soft constrained or regularized cost function is thus defined as

$$\tilde{f}(\mathbf{x}) = f(\mathbf{x}) + \lambda \sum_{i=1}^3 f_{c_i}(\mathbf{x}), \text{ with } \lambda > 0$$

To derive a Gauss-Newton iterative optimization algorithm over the stacked parameters \mathbf{x} , we observe that both the gradient and the Hessian of this soft-constrained optimization problem, correspond to the sum of the unconstrained counterparts (5.27) and (5.28) together with the constraints gradients (5.30) and Hessians (5.31), respectively, weighted by the regularization term λ . As before, the update rule for each rotation matrix involved in the minimization is given by the incremental contribution computed from the optimal parameters \mathbf{x}^* using the exponential map. The proposed algorithm is summarized in Algorithm 2.

5.4.2 Experimental results

The soft constrained or regularized three view optimization algorithm has been tested on a synthetic dataset. This dataset was created through a simulated camera model, calibrated with OpenCV using a virtual calibration pattern (chessboard). Camera model, chessboard pattern and sensor motion are all simulated using OpenGL. The dataset is composed by 30 different views of the chessboard pattern, ground truth motion of the sensor, estimated intrinsic parameters and the set of point correspondences given by the tracked pattern corners. To evaluate the improvement of the joint estimate of pair-wise stereo geometry, we analyze three different error quantities. The first, \mathbf{e}_e , proportional to the symmetric epipolar error and equivalent to (5.25). The second

$$\mathbf{e}_c = \sum_{i=1}^3 f_{c_i}(\mathbf{x})$$

accounting for the soft coplanarity constraint. The third, measuring the rotational consistency between the pair-wise rotation estimates

$$\mathbf{e}_r = \frac{1}{2}(3 - \text{tr}\{\mathbf{R}_3\mathbf{R}_1\mathbf{R}_2\})^2$$

Figures 5.1 and 5.2 present sixteen different random configuration and the optimization results provided by both the unconstrained and the soft constrained algorithms. In all cases, the coplanarity constraint provides an interesting improvement on the consistency between the estimates of the three stereo geometries. We observe that the coplanarity error \mathbf{e}_c is always decreased with the respect the unconstrained algorithm, for at least, four orders of magnitude. Instead, the rotational error \mathbf{e}_r is never worsened and in the vast majority of cases is also notably improved. A key contradiction comes from the epipolar term \mathbf{e}_e , which is always worsened within the constraint. This is, in fact, the expected behavior because the algorithm plays with the trade off between over fitting of the images data for independent pair-wise geometry estimation and keeping a consistent structure between all stereo geometries.

Despite the improved accuracy, this regularized optimization schemes present some drawbacks. First of all, outcomes fully depend on the value of the regularization parameter λ , which it is arbitrarily defined, and an optimal choice requires off-line parameters tuning, which in general is not desired for a visual odometry algorithm that usually works on real-time. During the empirical evaluation we observed that a good choice for the regularization parameter is the mean baseline

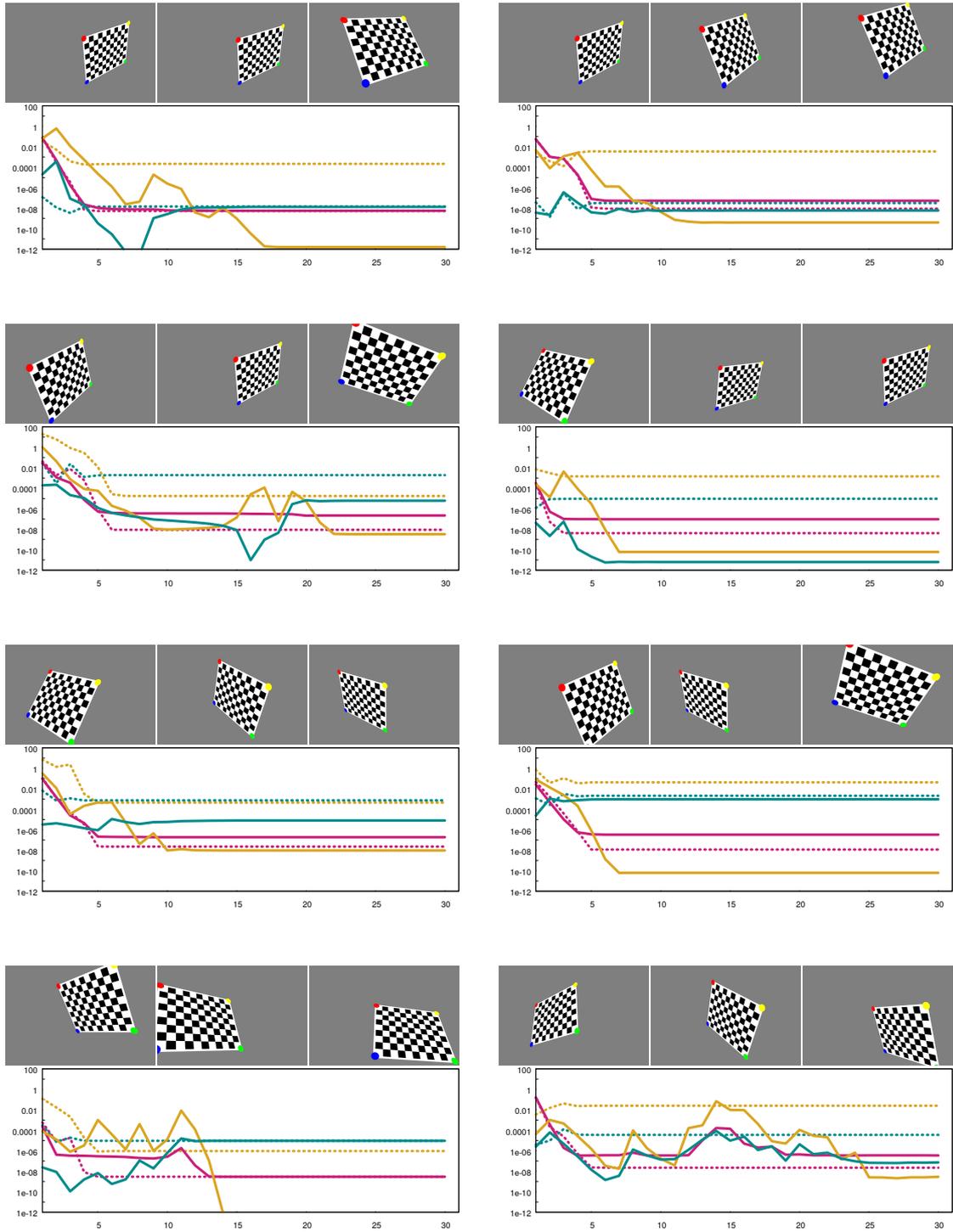


Figure 5.1: Experimental evaluation of the soft-constrained algorithm over synthetic data. On top of each figure are present the synthetic views of the virtual chessboard used to estimate the pair-wise stereo geometries, from left (first view) to right (third view). On bottom the associated errors: e_e (magenta), e_c (yellow) and e_r (cyan); dashed lines correspond to the unconstrained optimization while continuous lines to the soft constrained or regularized algorithm.

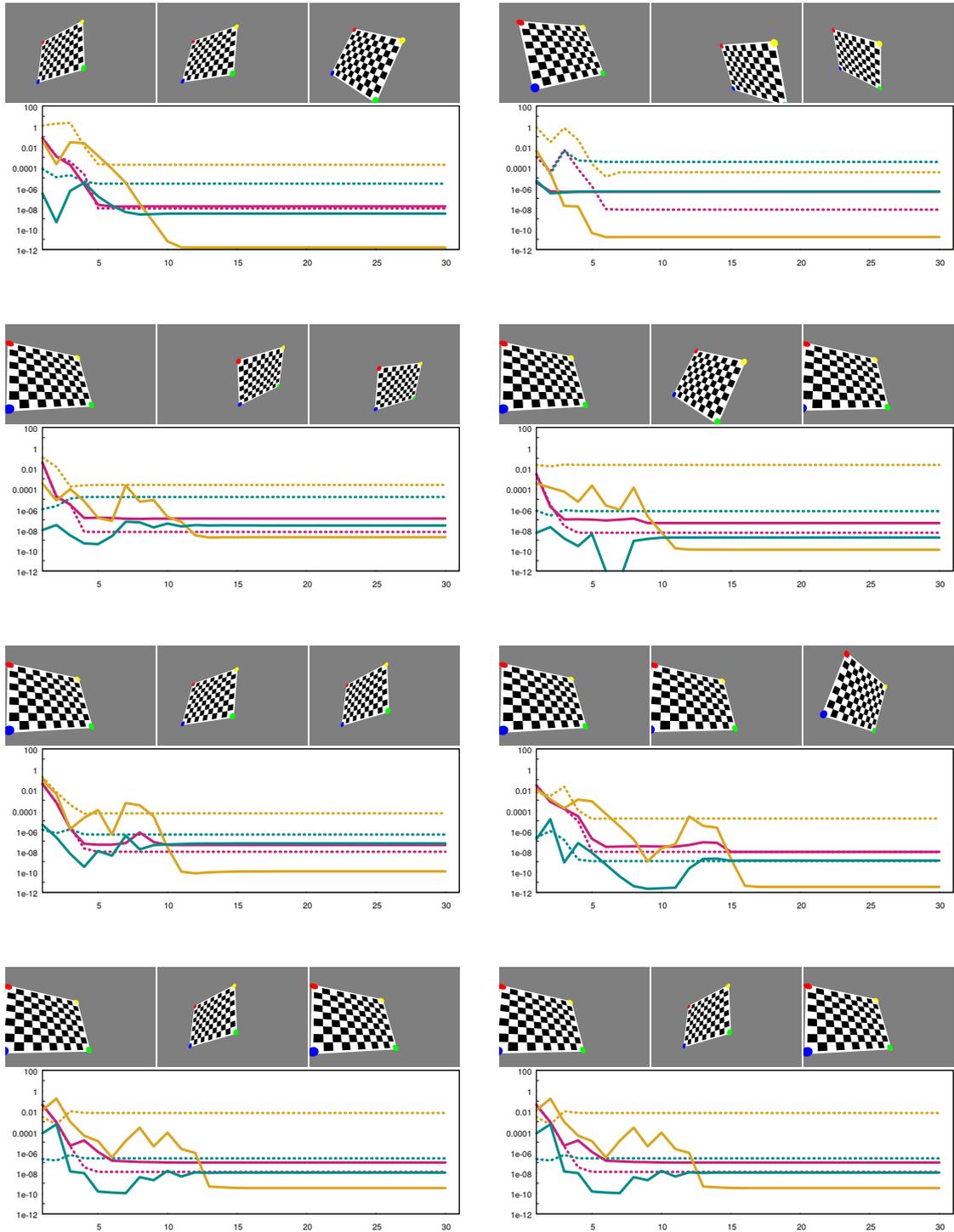


Figure 5.2: Experimental evaluation of the soft-constrained algorithm over synthetic data. On top of each figure are present the synthetic views of the virtual chessboard used to estimate the pair-wise stereo geometries, from left (first view) to right (third view). On bottom the associated errors: e_e (magenta), e_c (yellow) and e_r (cyan); dashed lines correspond to the unconstrained optimization while continuous lines to the soft constrained or regularized algorithm.

length of the whole agent’s motion. We can justify this heuristic by noting that the magnitude of the coplanarity error linearly depends on the true scale, which in turns depends in the true length of the baselines; by approximating the mean baseline length, we also provide a mean scale estimate. Second, a direct consequence of the soft-constrained approach is that the monotonic convergence of the Helmke et al. [51] algorithm can be lost, depending on the values of the regularization parameter. In fact, an estimate of the error reduction cannot be anymore computed in terms of the number of iterations, since during convergence there may be either transient oscillatory behaviors or premature arrival to a false minimum. Thus, the number of iterations necessary for convergence may be dramatically increased with respect the unconstrained counterpart. Some examples of this issues are presented on Figure 5.3.

5.4.3 Three-view hard-constrained optimization

Until now we have introduced the coplanarity constraint as a regularization term for the unconstrained optimization problem. The idea now is to map the set of constraints 5.29 into the parameter space $\mathcal{U} \subset \mathbb{R}^{15}$, to reduce the dimensionality of the problem by projecting parameters motions into a feasible subspace that preserves coplanarity during the algorithm’s iterations. We start by recalling the definition of the matrix J_{c_i} previously defined on equation 5.30

$$J_{c_i} = (\mathbf{I} \otimes \mathbf{E}_{c_i}) \mathbf{Q}_{U_j} - (\mathbf{E}_{c_i} \otimes \mathbf{I}) \mathbf{Q}_{V_k} + (\mathbf{V}_i^\top \mathbf{U}_j \otimes \mathbf{V}_k^\top \mathbf{U}_i) [(\mathbf{E}_0 \otimes \mathbf{I}) \mathbf{Q}_{U_i} - (\mathbf{I} \otimes \mathbf{E}_0) \mathbf{Q}_{V_i}]$$

Now, by computing the equivalent kinematic constraint, that is, by taken the temporal derivative of the geometrical constraints 5.29, we have that

$$0 = \text{vec}^\top \{ \mathbf{z} \mathbf{z}^\top \} J_{c_i} \mathbf{x} = \mathbf{j}_{c_i}^9 \mathbf{x}, \quad \mathbf{x} \in \mathcal{U}$$

since the row vector $\text{vec}^\top \{ \mathbf{z} \mathbf{z}^\top \}$ is zero except for the last element, then the product $\text{vec}^\top \{ \mathbf{z} \mathbf{z}^\top \} J_{c_i}$ is equivalent to the ninth (last) row $\mathbf{j}_{c_i}^9$ of J_{c_i} . Now, we can stack the above vector constraints to form a (3×15) matrix \mathbf{A} holding all kinematic constraints of feasible motions in the parameters space at the given state⁴.

$$\mathbf{A} = \begin{pmatrix} \mathbf{j}_{c_1}^9 \\ \mathbf{j}_{c_2}^9 \\ \mathbf{j}_{c_3}^9 \end{pmatrix} \implies 0 = \mathbf{A} \mathbf{x} \implies \mathbf{x} \in \mathcal{N}(\mathbf{A}) \subset \mathcal{U} \quad (5.32)$$

⁴By state we refer to the current estimates of the set of matrices \mathbf{U}_{ij} and \mathbf{V}_{ij} under optimization.

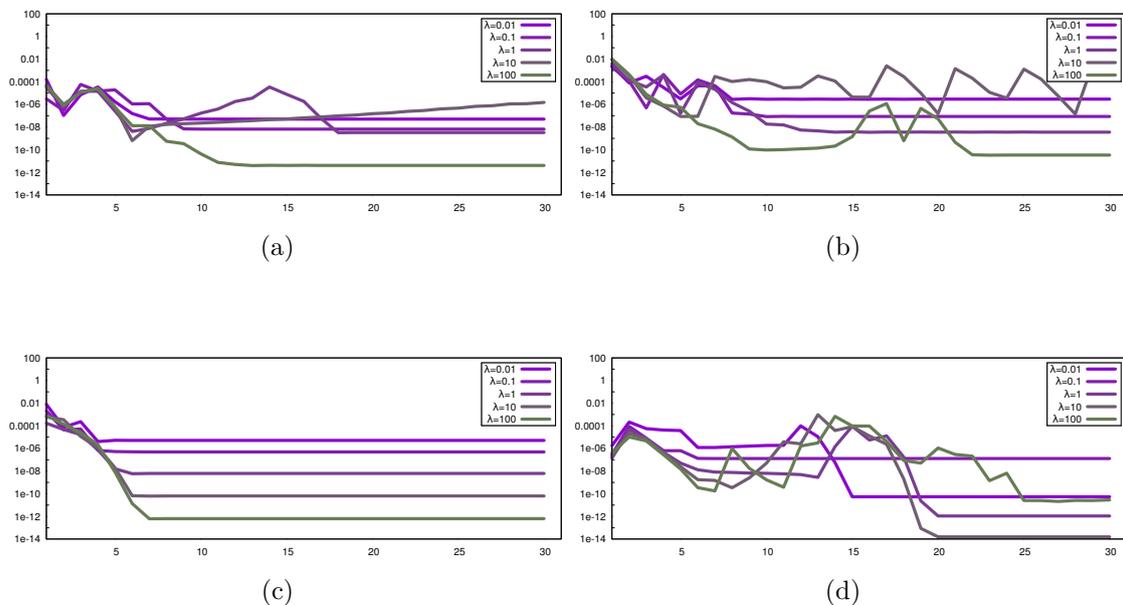


Figure 5.3: Soft-constrained optimization issues. As we observe, the monotonic behavior of the unconstrained algorithm may be potentially lost when using regularization. The above Figures show the coplanarity error evolution during optimization for different values of the regularization parameter λ . Each graph is associated to a particular set of three randomly chosen views from the synthetic dataset. A nice behavior is shown on Figure (c), where monotonic convergence is kept and also linearly accelerated for increasing values of the regularization parameter λ . However, this is not the only behavior that we can obtain, as shown on the rest of images. In Figure (a) we can observe that convergence is not linearly associated to the regularization parameter, since the worst result is obtained for a middle value of λ ; we note that with respect to the number of iterations considered in the experiments, the worst behavior is characterized by a local divergence of the error. Another example of this issue is provided in Figure (b), where the worst result is defined by an oscillatory behavior for a middle value of λ ; moreover, we observe that the behavior of the best solution seems to reach a premature local minimum before reaching a stable convergence. Finally, in Figure (d), we can observe how the premature minimum trap affects almost all regularized solutions together with transient oscillations until convergence.

where $\mathcal{N}(\mathbf{A})$ represents the null space of \mathbf{A} .

At this point, the objective of our hard-constrained optimization problem is to minimize the cost function 5.26 subject to the set of feasible motions 5.32. The proposed constrained optimization scheme (Algorithm 3) is based on the iterative refinement of pair-wise stereo geometry estimates, constrained to satisfy at each iteration the coplanarity constraints. The first step consist in the linear initialization of the pair-wise stereo geometry between views and the successive projection into the set of solutions satisfying the constraints from 5.16 to 5.18, that is, the set where unitary translation vectors \mathbf{t}_{ij} are all coplanar. A simple way to satisfy this condition is

$$\mathbf{t}_{31} = \frac{-\mathbf{R}_{23}^\top [\mathbf{t}_{23} + \mathbf{R}_{12}^\top \mathbf{t}_{12}]}{\|\mathbf{t}_{23} + \mathbf{R}_{12}^\top \mathbf{t}_{12}\|}$$

The second step of the algorithm consists on the on-manifold iterative refinement based on on-manifold Gauss-Newton iterations. Let's us recall that the optimal solution \mathbf{x}_{NG}^* of the unconstrained three-view problem is obtained by solving the linear system 5.14, where gradient and Hessian matrix are given by equations 5.27 and 5.28, respectively. Now, based on the motion constraint 5.32, we know that the vector \mathbf{x} can be parameterized in terms of a lower dimensional parameters vector $\tilde{\mathbf{x}} \in \mathbb{R}^{12}$, that is $\mathbf{x} = \mathbf{G}\tilde{\mathbf{x}}$, where \mathbf{G} is a (15×12) orthonormal basis for the null space of \mathbf{A} . Therefore, by inserting this parametrization inside the Gauss-Newton system we have

$$\begin{aligned} -\tilde{\nabla}^\top(f \circ \varphi)(0) &= \tilde{\mathbf{H}}_{f \circ \varphi(0)} \tilde{\mathbf{x}}^* & \text{with} & \quad \tilde{\nabla}(f \circ \varphi) = \mathbf{G}^\top \nabla(f \circ \varphi) \\ \mathbf{x}_{GN}^* &= \mathbf{G}\tilde{\mathbf{x}}^* & & \quad \tilde{\mathbf{H}}_{f \circ \varphi} = \mathbf{G}^\top \mathbf{H}_{f \circ \varphi} \mathbf{G} \end{aligned} \quad (5.33)$$

where $\tilde{\nabla}(\cdot)$ and $\tilde{\mathbf{H}}(\cdot)$ are, respectively, the constrained gradient and Hessian. Therefore, the on-manifold iterative refinement is divided into two operations: computation of \mathbf{G} based on the current estimates and computation of the optimal motion using the constrained Gauss-Newton rule 5.33. Afterwards, the update rule 5.15 is used to calculate the new estimates.

5.4.4 Experimental results

The experimental setup, measured error quantities and outcome evaluation for the analysis of the hard constrained algorithm correspond to those described in Section 5.4.2 during the analysis of the regularized one.

As before, Figures 5.4 and 5.5 present sixteen different random configuration

Algorithm 3: Hard-constrained optimization of pair-wise stereo geometry between three views

Data:

K: camera calibration matrix;
 $\mathbf{X}_{12}, \mathbf{X}_{23}, \mathbf{X}_{31}$: point correspondences between views;
 ϵ : desired accuracy;

Result:

$\mathbf{E}_{12}, \mathbf{E}_{23}, \mathbf{E}_{31}$: pair-wise stereo geometry;

Step 0:

linear initialization of \mathbf{E}_{ij} (e.g., using the 8p [49] or Fiore [35] algorithm);
 computation of \mathbf{U}_{ij} and \mathbf{V}_{ij} (i.e., projection of \mathbf{E}_{ij} into the essential manifold);

Step 1:

projection of \mathbf{U}_{ij} and \mathbf{V}_{ij} into the coplanarity constraint;

Step 2: constrained Gauss-Newton iterations;

do

compute the motion constraint matrix \mathbf{A} (eq. 5.32);
 compute the orthonormal basis \mathbf{G} for the null space of \mathbf{A} ;
 compute $\tilde{\nabla}(f \circ \varphi)(0)$ and Hessian $\tilde{\mathbf{H}}(f \circ \varphi)(0)$ (eq. 5.33);
 get the constrained optimal motion $\tilde{\mathbf{x}}^*$ (by solving eq. 5.33);
 compute $\mathbf{x}_{GN}^* = \mathbf{G}\tilde{\mathbf{x}}^*$;
 update \mathbf{U}_{ij} and \mathbf{V}_{ij} using the exponential map (eq. 5.15);

while $\epsilon \geq \|\tilde{\nabla}(f \circ \varphi)(0)\|$;

and the optimization results provided by both the unconstrained and the hard constrained algorithms. In all cases, coplanarity and rotational errors are always decreased with respect the unconstrained algorithm and, as consequence, pair-wise stereo geometry estimates present a stronger consistency between them. With respect the outcomes of the regularized algorithm, we observe that the hard-constrained algorithm is faster, more stable and does not depend on any parameter. In fact, few iterations are need for the algorithm to converge, due to the smaller number of degrees of freedom involved in the optimization. Moreover, the stability can be qualitatively analyzed through the convergence behavior, which is almost monotone decreasing and smoother. Although the regularized algorithm can reduce the coplanarity error by two or more orders of magnitude, there is not a direct nor clear effect on the rotational error, while the hard-constrained approach apply the same order reduction to both coplanarity and rotational errors.

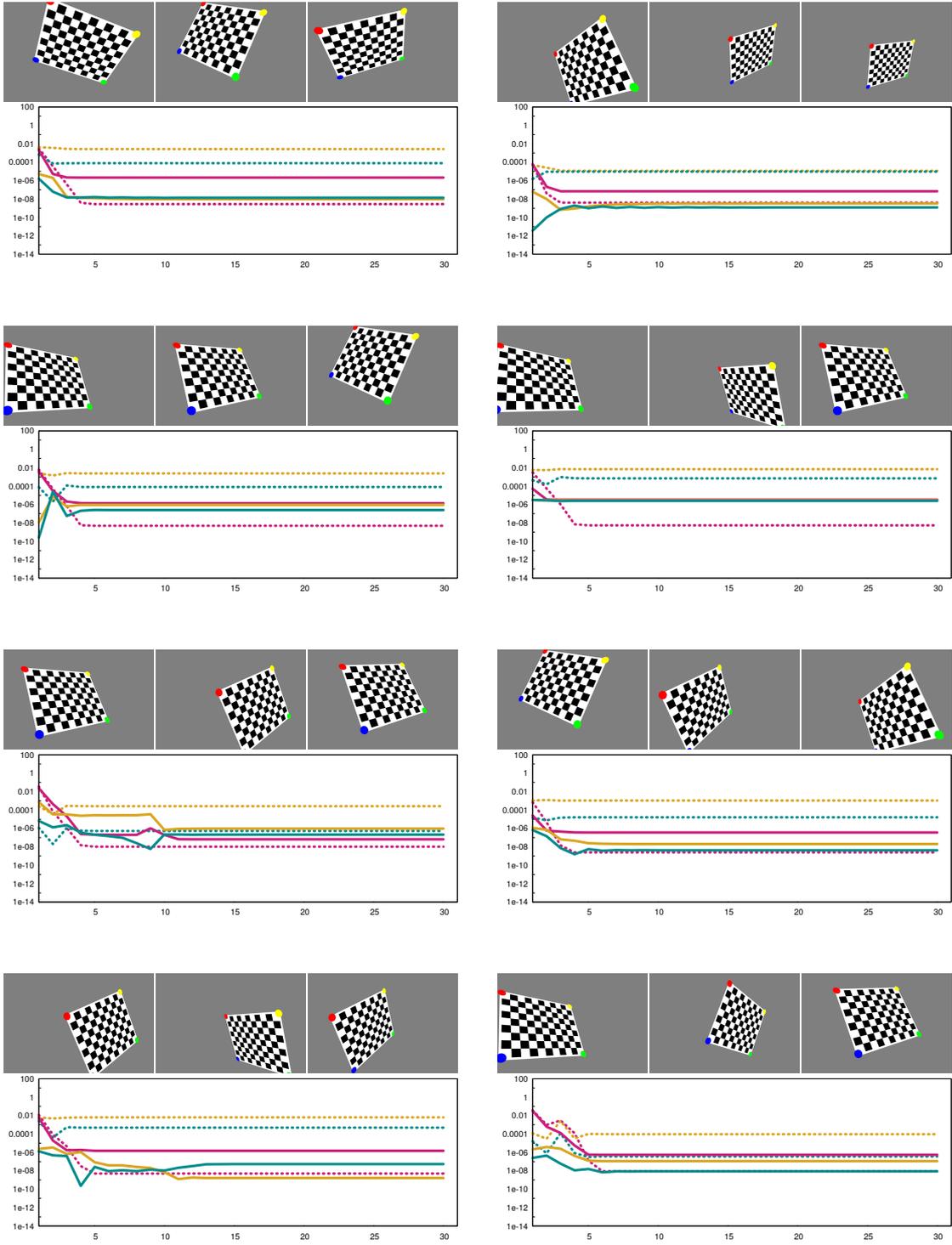


Figure 5.4: Experimental evaluation of the hard-constrained algorithm over synthetic data. On top of each figure are present the synthetic views of the virtual chessboard used to estimate the pair-wise stereo geometries, from left (first view) to right (third view). On bottom the associated errors: e_e (magenta), e_c (yellow) and e_r (cyan); dashed lines correspond to the unconstrained optimization while continuous lines to the constrained one.

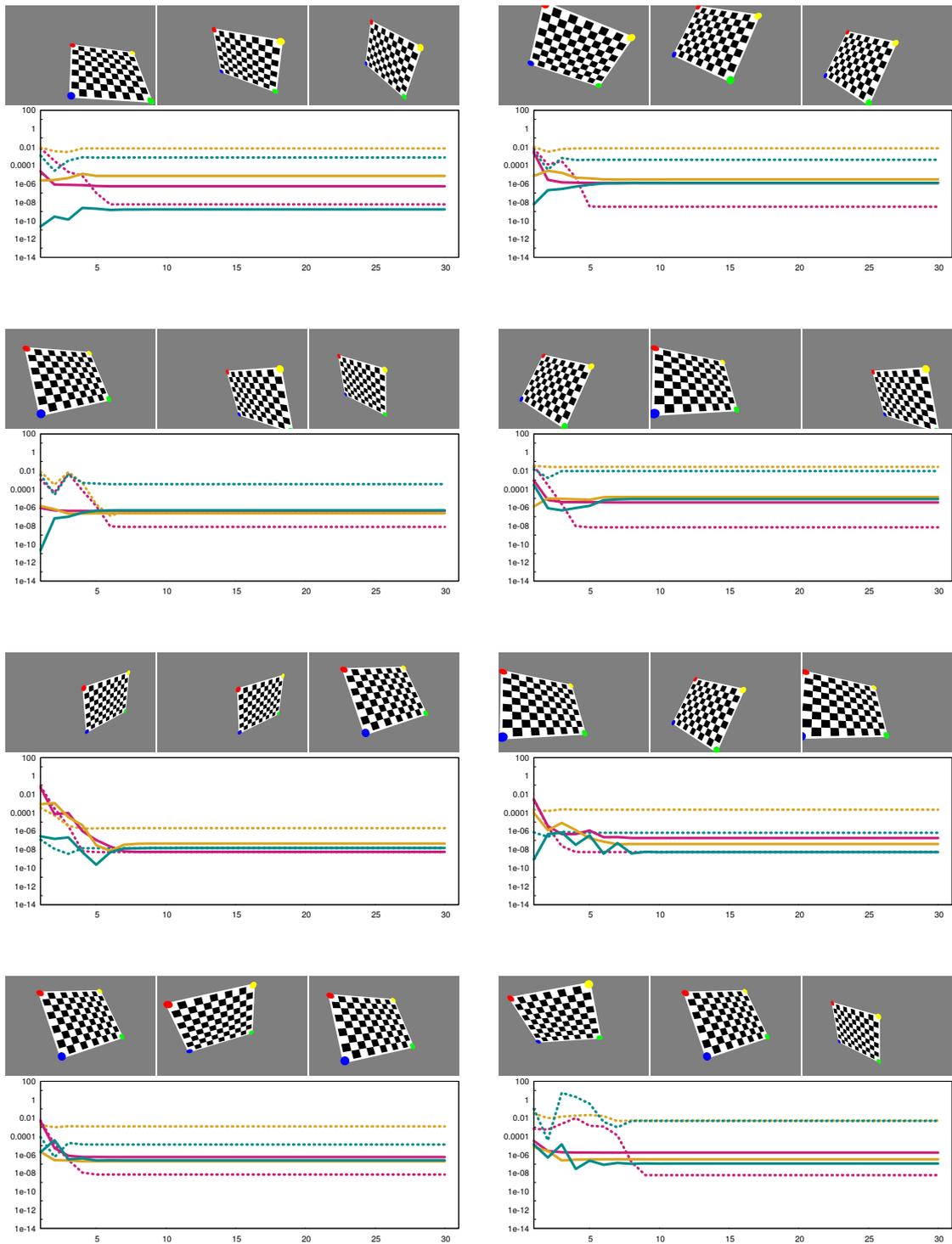


Figure 5.5: Experimental evaluation of the hard-constrained algorithm over synthetic data. On top of each figure are present the synthetic views of the virtual chessboard used to estimate the pair-wise stereo geometries, from left (first view) to right (third view). On bottom the associated errors: e_e (magenta), e_c (yellow) and e_r (cyan); dashed lines correspond to the unconstrained optimization while continuous lines to the constrained one.

5.5 Conclusions and future work

We have focused our study into the understanding of the coplanarity constraint and how it can be exploited to produce optimization algorithms able to estimate consistent pair-wise stereo geometries between three views. As a result, two different optimization algorithms have been proposed. The first, introduces the coplanarity constraint as a regularization term of the cost function. Optimal solutions are obtained by unconstrained on-manifold optimization of pair-wise stereo geometries. The accuracy of the tree-view consistency depends on the trade-off defined by the regularization term. The second, translates the coplanarity constraints into motion constraints directly in the tangent space of the manifold. Such optimal solutions are consistent with the three view geometry. The validity and correctness of the proposed algorithms has been tested and demonstrated using a synthetic calibration dataset. In particular, the constrained algorithm provides better results in term of accuracy, computation complexity (optimization on lower dimensional tangent space) and numerical stability (no parametric dependencies).

However, further experimental evaluation with standard datasets is required to assess the accuracy of the proposed algorithms. Another important issue is that computing optimal estimates alone is not sufficient [54], we must provide a measure of how reliable an optimal solution is. The presented methods are purely geometric models and completely lack a statistical measure of both the input noise, and the fidelity of the solution. This is a non negligible weakness, since outliers are ubiquitous in any computer vision inverse problem.

We have limited our analysis to the case of three views, however in the general case of multiple-views, many interesting theoretical and practical implications arise. In the case of four views, the coplanarity constraint is defined over four different baseline planes, instead of one as in the three views case. In general, for a number of views n , we observe that the number of planes is given by the combination

$${}^nC_3 = \frac{n!}{(n-3)!3!}$$

with a total number of constraints equal to $3 {}^nC_3$. On the other hand, the number of degrees of freedom required for a full pair-wise stereo analysis over the n views equals $5 {}^nC_2$. Which implies that, for $n=7$, we have a number of constraints equal to the number of degrees of freedom, and for $n>7$ the optimization becomes over determined. Therefore, it is of fundamental regard to analyze –theoretically and empirically– how coplanarity constraints can be defined over a fully-constrained

($3 < n \leq 7$) or sparsely-constrained ($n > 3$) set of views, and, in particular characterize an optimal number of coplanarity constraints that can be imposed to derive optimal bundle adjustment strategies.

As a last remark, in Section 5.3.2 we have shown that the set of coplanarity constraints represent an holonomic constraint, however we did not provide a proper analytical representation of the associated geometric object. An interesting line for future research could be to provide further insight in the intrinsic geometry of the object and to determine its relation with other multiple views representations like, for example, the *trifocal* and *quadrifocal* tensors, respectively, in the case of three and four views.

References

- [1] N. Aghakhani, M. Geravand, N. Shahriari, M. Vendittelli, and G. Oriolo. Task control with remote center of motion constraint for minimally invasive robotic surgery. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 5807–5812, 2013.
- [2] A.N. Akansu and P.A. Haddad. *Multiresolution signal decomposition: transforms, subbands, and wavelets*. Elsevier Science, 2000. ISBN 9780080512297.
- [3] J.C. Arevalo, X. Carrillo, M. Cestari, D. Sanz-Merodio, and E. Garcia. System identification applied to contact modeling: an experimental investigation. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 3699–3706, 2013.
- [4] B. Bayle, J. Fourquet, and M. Renaud. Manipulability of wheeled mobile manipulators: application to motion generation. *The International Journal of Robotics Research*, 22(7-8):565–581, 2003.
- [5] A. Bellet, A. Habrard, and M. Sebban. A survey on metric learning for feature vectors and structured data. *ArXiv e-prints*, 2013.
- [6] R. Bischoff, U. Huggenberger, and E. Prassler. KUKA youBot - a mobile manipulator for research and education. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1–4, 2011.
- [7] M. Blanke, M. Kinnaert, J. Lunze, and M. Staroswiecki. *Diagnosis and fault-tolerant control*. Springer-Verlag, 2 edition, 2006. ISBN 9783642071362.
- [8] J. Boyd. *Chebyshev and Fourier spectral methods*. Dover Publications, 2 edition, 2001. ISBN 9780486411835.
- [9] F. Båberg, Y. Wang, S. Caccamo, and P. Ögren. Adaptive object centered teleoperation control of a mobile manipulator. In *ICRA*, pages 455–461, 2016.

- [10] A. Calanca, L. Capisani, A. Ferrara, and L. Magnani. MIMO closed loop identification of an industrial robot. *IEEE Transactions on Control Systems Technology*, 19(5):1214–1224, 2011.
- [11] R. L. Carceroni, A. Kumar, and K. Daniilidis. Structure from motion with known camera positions. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 477–484, 2006.
- [12] J. Carlson and R.R. Murphy. How UGVs physically fail in the field. *IEEE Transactions on Robotics*, 21(3):423–437, 2005.
- [13] T. Chan and R. Dubey. A weighted least-norm solution based scheme for avoiding joint limits for redundant joint manipulators. *IEEE Transactions in Robotics and Automation*, 11(2):286–292, 1995.
- [14] H. Choset, K. Lynch, S. Hutchinson, G. Kantor, W. Burgard, L. Kavraki, and S. Thrun. *Principles of robot motion: theory, algorithms, and implementations*. MIT Press, 2005. ISBN 9780262033275.
- [15] J. Claret, I. Zaplana, and L. Basañez. Teleoperating a mobile manipulator and a free-flying camera from a single haptic device. In *IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, pages 291–296, 2016.
- [16] R. Coifman and M. Wickerhauser. Entropy-based algorithms for best basis selection. *IEEE Transactions on Information Theory*, 38(2):713–718, 1992.
- [17] R. Coifman, Y. Meyer, S. Quake, and M. Wickerhauser. Signal processing and compression with wavelet packets. In *Wavelets and their applications*, pages 363–379. Springer, 1994.
- [18] M. Dalvand and B. Shirinzadeh. Remote centre-of-motion control algorithms of 6-RRCRR parallel robot assisted surgery system (PRAMiSS). In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 3401–3406, 2012.
- [19] B. d’Andrea Novel, G. Bastin, and G. Campion. Dynamic feedback linearization of nonholonomic wheeled mobile robots. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 2527–2532, 1992.

- [20] A. De Luca and L. Ferrajoli. A modified newton-euler method for dynamic computations in robot fault detection and control. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 3359–3364, 2009.
- [21] A. De Luca and R. Mattone. Actuator failure detection and isolation using generalized momenta. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 634–639, 2003.
- [22] A. De Luca and R. Mattone. Sensorless Robot Collision Detection and Hybrid Force/Motion Control. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 999–1004, 2005.
- [23] A. De Luca and G. Oriolo. The reduced gradient method for solving redundancy in robot arms. *11th IFAC World Congress on Automatic Control*, 23: 133–138, 1990.
- [24] A. De Luca, A. Albu-Schaffer, S. Haddadin, and G. Hirzinger. Collision detection and safe reaction with the DLR-III lightweight manipulator arm. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1623–1630, 2006.
- [25] A. De Luca, G. Oriolo, and P. Robuffo Giordano. Kinematic modeling and redundancy resolution for nonholonomic mobile manipulators. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 1867–1873, 2006.
- [26] C. De Persis and A. Isidori. A geometric approach to nonlinear fault detection and isolation. *IEEE Transactions on Automatic Control*, 46(6):853–865, 2001.
- [27] H. de Vries, R. Memisevic, and A. Courville. Deep learning vector quantization. In *European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN)*, 2016.
- [28] W. E. Dixon, I. D. Walker, D. M. Dawson, and J. P. Hartranft. Fault detection for robot manipulators with parametric uncertainty: a prediction error based approach. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, volume 4, pages 3628–3634, 2000.
- [29] K. Doty, C. Melchiorri, E. Schwartz, and C. Bonivento. Robot manipulability. *IEEE Transactions on Robotics and Automation*, 11(3):462–468, 1995.

- [30] J Engel, J Sturm, and D Cremers. Semi-dense visual odometry for a monocular camera. In *IEEE International Conference on Computer Vision (ICCV)*, pages 1449–1456, 2013.
- [31] J Engel, T Schöps, and D Cremers. LSD-SLAM: Large-Scale Direct Monocular SLAM. In *Proceedings of the European conference on Computer vision (ECCV)*, pages 834–849, 2014.
- [32] B. Farkas and S. Wegner. Variations on Barbualat’s Lemma. *ArXiv e-prints*, 2014.
- [33] O Faugeras. *Three-dimensional computer vision: a geometric viewpoint*. MIT Press, 1993. ISBN 0262061589.
- [34] A. Ferrara and G. Incremona. Design of an integral suboptimal second-order sliding mode controller for the robust motion control of robot manipulators. *IEEE Transactions on Control Systems Technology*, 23(6):2316–2325, 2015.
- [35] P. D Fiore. Efficient linear solution of exterior orientation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 23(2):140–148, 2001.
- [36] F. Flacco, A. De Luca, and O. Khatib. Motion control of redundant robots under joint constraints: saturation in the null space. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 285–292, 2012.
- [37] F. Flacco, A. De Luca, and O. Khatib. Prioritized multi-task motion control of redundant robots under hard joint constraints. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3970–3977, 2012.
- [38] Y. Fukuoka, K. Hoshi, and R. Kurosawa. Negotiating uneven terrain with a compliant designed unmanned ground vehicle equipped with locomotive master-slave operation. *Journal of Field Robotics*, 30(3):349–370, 2013.
- [39] C. Gaz and A. De Luca. Payload estimation based on identified coefficients of robot dynamics -with an application to collision detection. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3033–3040, 2017.

- [40] C. Gaz, F. Flacco, and A. De Luca. Identifying the dynamic model used by the KUKA LWR: a reverse engineering approach. In *IEEE International Conference on Robotics and Automation ICRA*, pages 1386–1392, 2014.
- [41] A. Ghalamzan, F. Abi-Farraj, P. Robuffo Giordano, and R. Stolkin. Human-in-the-loop optimisation: mixed initiative grasping for optimally facilitating post-grasp manipulative actions. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3386–3393, 2017.
- [42] M. Gianni, F. Ferri, M. Menna, and F. Pirri. Adaptive robust three-dimensional trajectory tracking for actively articulated tracked vehicles. *Journal of Field Robotics*, 2015. ISSN 1556-4967.
- [43] M. Gianni, M. A. Ruiz Garcia, F. Ferri, and F. Pirri. Terrain contact modeling and classification for ATVs. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 186–192, 2016.
- [44] G. Gilardi and I. Sharf. Literature survey of contact dynamics modelling. *Mechanism and Machine Theory*, 37:1213–1239, 2002.
- [45] F. Girosi. An equivalence between sparse approximation and support vector machines. *Neural computation*, 10(6):1455–1480, 1998.
- [46] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182, 2003.
- [47] C. Ha, S. Park, J. Her, I. Jang, Y. Lee, G. Cho, H. Son, and D. Lee. Whole-body multi-modal semi-autonomous teleoperation of mobile manipulator systems. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 164–170, 2015.
- [48] S. Haddadin, A. Albu-Schäffer, A. De Luca, and G. Hirzinger. Collision detection and reaction: a contribution to safe physical human-robot interaction. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3356–3363, 2008.
- [49] A Harlley and A Zisserman. *Multiple view geometry in computer vision*. Cambridge University Press, 2 edition, 2006. ISBN 9780521540513.
- [50] S. Haykin. *Nonlinear methods of spectral analysis*. Springer-Verlag, 1983. ISBN 9783540707523.

- [51] U Helmke, K Hüper, P Lee, and J Moore. Essential matrix estimation using Gauss-Newton iterations on a manifold. *International Journal of Computer Vision (IJCV)*, 74(2):117–136, 2007.
- [52] D. Inoue, M. Konyo, K. Ohno, and S. Tadokoro. Contact points detection for tracked mobile robots using inclination of track chains. In *IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, pages 194–199, 2008.
- [53] T. Jung, J. Lim, H. Bae, K. Lee, H. Joe, and J. Oh. Development of the humanoid disaster response platform DRC-HUBO+. *IEEE Transactions on Robotics*, 34(1):1–17, 2018.
- [54] K. Kanatani. *Statistical optimization for geometric computation*. Dover, 1996. ISBN 9780486443089.
- [55] S. Katsura, Y. Matsumoto, and K. Ohnishi. Analysis and experimental validation of force bandwidth for force control. In *IEEE Transactions on Industrial Electronics*, volume 2, pages 796–801, 2003.
- [56] K. Katyal, C. Brown, S. Hechtman, M. Para, T. McGee, K. Wolfe, R. Murphy, M. Kutzer, E. Tunstel, P. McLoughlin, and M. Johannes. Approaches to robotic teleoperation in a disaster scenario: from supervised autonomy to direct control. In *IROS*, pages 1874–1881, 2014.
- [57] A. Kazemi Nasab, A. Kilicman, E. Babolian, and Z. Pashazadeh Atabakan. Wavelet analysis method for solving linear and nonlinear singular boundary value problems. *Applied Mathematical Modelling*, 37(8): 5876–5886, 2013.
- [58] D. Kostić, B. de Jager, M. Steinbuch, and R. Hensen. Modeling and identification for high-performance robot control: an RRR-robotic arm case study. *IEEE Transactions on Control Systems Technology*, 12(6):904–919, 2004.
- [59] G.-J.M. Kruijff, V. Tretyakov, T. Linder, F. Pirri, M. Gianni, P. Papadakis, M. Pizzoli, A. Sinha, E. Pianese, S. Corrao, F. Priori, S. Febrini, and S. Angeletti. Rescue robots at earthquake-hit Mirandola, Italy: a field report. In *IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, pages 1–8, 2012.

- [60] B. Larochelle and G. Kruijff. Multi-view operator control unit to improve situation awareness in USAR missions. In *21st IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pages 1103–1108, 2012.
- [61] J. Li, Z. Li, and K. Hauser. A study of bidirectionally telepresent teleaction during robot-mediated handover. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 2890–2896, 2017.
- [62] Y Ma, J Kosecká, and S Sastry. Optimization criteria and geometric algorithms for motion and structure estimation. *International Journal of Computer Vision (IJCV)*, 44(3):219–249, 2001.
- [63] A. Maciejewski and C. Klein. Numerical filtering for the operation of robotic manipulators through kinematically singular configurations. *Journal of Field Robotics*, 5:527–552, 1988.
- [64] S. Mallat. *A wavelet tour of signal processing, third edition: the sparse way*. Academic Press, 3rd edition, 2008.
- [65] M. Menna, M. Gianni, F. Ferri, and F. Pirri. Real-time autonomous 3D navigation for tracked vehicles in rescue environments. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 696–702, 2014.
- [66] A. Messina and V. Vittal. Extraction of dynamic patterns from wide-area measurements using empirical orthogonal functions. *IEEE Transactions on Power Systems*, 22(2):682–692, 2007.
- [67] J. Moore, K. Wolfe, M. Johannes, K. Katyal, M. Para, R. Murphy, J. Hatch, C. Taylor, R. Bamberger, and E. Tunstel. Nested marsupial robotic system for search and sampling in increasingly constrained environments. In *IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2016.
- [68] R Mur-Artal and J. D Tardós. Fast relocalisation and loop closing in keyframe-based SLAM. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 846–853, 2014.
- [69] K. Nagatani, S. Kiribayashi, Y. Okada, S. Tadokoro, T. Nishimura, T. Yoshida, E. Koyanagi, and Y. Hada. Redesign of rescue mobile robot Quince. In *IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, pages 13–18, 2011.

- [70] M. H. Nguyen and F. De la Torre. Optimal feature selection for support vector machines. *Pattern Recognition*, 43(3):584–591, 2010.
- [71] L. Nouredine, A. Kouzou, M. Guemana, and A. Hafaifa. Faults detection in squirrel cage induction generator rotor on wind turbine: Hilbert Huang transform application. In *6th International Conference on Systems and Control (ICSC)*, pages 403–408, 2017.
- [72] D. Oetomo and M. Ang. Singularity robust algorithm in serial manipulators. *Robotics and Computer-Integrated Manufacturing*, 25(1):122–134, 2009.
- [73] P. Ögren, P. Svenmarck, P. Lif, M. Norberg, and N. Söderbäck. Design and implementation of a new teleoperation control mode for differential drive UGVs. *Autonomous Robots*, 37(1):71–79, 2014.
- [74] A. Pepe, D. Chiaravalli, and C. Melchiorri. A hybrid teleoperation control scheme for a single-arm mobile manipulator with omnidirectional wheels. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1450–1455, 2016.
- [75] J Ponce, M Hebert, and H Martial. Trinocular geometry revisited. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17–24, 2014.
- [76] A. Ponce-Hinestroza, J. Castro-Castro, H. Guerrero-Reyes, V. Parra-Vega, and E. Olguín-Díaz. Cooperative redundant omnidirectional mobile manipulators: model-free decentralized integral sliding modes and passive velocity fields. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 2375–2380, 2016.
- [77] A Pretto, E Menegatti, and E Pagello. Omnidirectional dense large-scale mapping and navigation based on meaningful triangulation. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 3289–3296, 2011.
- [78] M. Reinstein, V. Kubelka, and K. Zimmermann. Terrain adaptive odometry for mobile skid-steer robots. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 4706–4711, 2013.
- [79] L. Rozo, N. Jaquier, S. Calinon, and D. Caldwell. Learning manipulability ellipsoids for task compatibility in robot manipulation. In *IEEE/RSJ*

- International Conference on Intelligent Robots and Systems (IROS)*, pages 3183–3189, 2017.
- [80] K. Sakurada, S. Suzuki, K. Ohno, E. Takeuchi, S. Tadokoro, A. Hata, N. Miyahara, and K. Higashi. Real-time prediction of fall and collision of tracked vehicle for remote-control support. In *IEEE/SICE International Symposium on System Integration (SII)*, pages 37–42, 2010.
- [81] J. Sandoval, G. Poisson, and P. Vieyres. Improved dynamic formulation for decoupled cartesian admittance control and RCM constraint. In *2016 IEEE International Conference on Robotics and Automation, (ICRA)*, pages 1124–1129, 2016.
- [82] J. Sandoval, G. Poisson, and P. Vieyres. A new kinematic formulation of the RCM constraint for redundant torque-controlled robots. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4576–4581, 2017.
- [83] D Scaramuzza and F Fraundorfer. Visual odometry [tutorial]. *IEEE Robotics and Automation Magazine*, 18(4):80–92, 2011.
- [84] D. Schinstock, T. Faddis, and R. Greenway. Robust inverse kinematics using damped least squares with dynamic weighting. In *AIAA/NASA Conference on Intelligent Robotics in Field, Factory, Service, and Space (CIRFFSS)*, volume 2, pages 861–869, 1994.
- [85] P. Schneider, M. Biehl, and B. Hammer. Adaptive relevance matrices in learning vector quantization. *Neural Computation*, 21(12):3532–3561, 2009.
- [86] A. Schulz, A. Gisbrecht, and B. Hammer. Relevance learning for dimensionality reduction. In *European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN)*, pages 165–170, 2014.
- [87] A. Schulz, B. Mokbel, M. Biehl, and B. Hammer. Inferring feature relevances from metric learning. In *IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 1599–1606, 2015.
- [88] L. Sciavicco and B. Siciliano. The augmented task space approach for redundant manipulator control. In *2nd IFAC Symposium on Robot Control (SYROCO)*, volume 21, pages 125–129, 1988.

- [89] B. Siciliano and O. Khatib, editors. *Springer handbook of robotics*. Springer, 2008. ISBN 9783540239574.
- [90] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo. *Robotics: modelling, planning and control*. Springer London, 1 edition, 2010. ISBN 9781846286414.
- [91] K. Song, S. Jiang, and M. Lin. Interactive teleoperation of a mobile manipulator using a shared-control approach. *IEEE Transactions on Human-Machine Systems*, 46(6):834–845, 2016.
- [92] S Song and M Chandaker. Robust scale estimation in real-time monocular SfM for autonomous driving. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1566–1573, 2014.
- [93] M. Spong, S. Hutchinson, and M. Vidyasagar. *Robot modeling and control*. Wiley, 2005. ISBN 9780471649908.
- [94] H Strasdat and A Davison. Scale drift-aware large scale monocular SLAM. In *Proceedings of robotics: science and systems*, 2010.
- [95] J. Swevers, C. Ganseman, D. Tükel, J. De Schutter, and H. Van Brussel. Optimal robot excitation and identification. *IEEE Transactions on Robotics and Automation*, 13(5):730–740, 1997.
- [96] S. Takakura, T. Murakami, and K. Ohnishi. An approach to collision detection and recovery motion in industrial robot. In *15th Annual Conference of IEEE Industrial Electronics Society (IECON)*, pages 421–426, 1989.
- [97] M. Tan, Li. Wang, and I. W. Tsang. Learning sparse svm for feature selection on very high dimensional datasets. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 1047–1054, 2010.
- [98] A. Torabi, M. Khadem, K. Zareinia, G. Sutherland, and M. Tavakoli. Manipulability of teleoperated surgical robots with application in design of master/slave manipulators. In *International Symposium on Medical Robotics (ISMR)*, 2018.
- [99] L. Trefethen. *Finite difference and spectral methods for ordinary and partial differential equations*, 1996.

- [100] L. Trefethen and M. Embree. *Spectra and pseudospectra: the behavior of nonnormal matrices and operators*. Princeton University Press, 2005. ISBN 9780691119465.
- [101] N. Vahrenkamp, T. Asfour, G. Metta, G. Sandini, and R. Dillmann. Manipulability analysis. In *IEEE-RAS International Conference on Humanoid Robots*, pages 568–573, 2012.
- [102] I. Vincent and Q. Sun. A combined reactive and reinforcement learning controller for an autonomous tracked vehicle. *Robotics and Autonomous Systems*, 60(4):599–608, 2012.
- [103] R. Wald, T. M. Khoshgoftaar, and J. C. Sloan. Feature selection for vibration sensor data transformed by a streaming wavelet packet decomposition. In *IEEE International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 978–985, 2011.
- [104] H. Wang, H. Liu, T. Qing, W. Liu, and T. He. An automatic fault diagnosis method for aerospace rolling bearings based on ensemble empirical mode decomposition. In *8th International Conference on Mechanical and Aerospace Engineering (ICMAE)*, pages 502–506, 2017.
- [105] J Wang and E. Olson. AprilTag 2: efficient and robust fiducial detection. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4193–4198, 2016.
- [106] Y. Wang. *Reactive Control and Coordination of Redundant Robotic Systems*. PhD thesis, School of Computer Science and Communication, KTH Royal Institute of Technology, 2016.
- [107] P. Wensing, S. Kim, and J. Slotine. Linear matrix inequalities for physically consistent inertial parameter identification: a statistical perspective on the mass distribution. *IEEE Robotics and Automation Letters*, 3(1):60–67, 2018.
- [108] J. Weston, A. Elisseeff, B. Schölkopf, and M. Tipping. Use of the zero norm with linear models and kernel methods. *Journal of Machine Learning Research*, 3:1439–1461, 2003.
- [109] M. Wrock and S. Nokleby. Decoupled teleoperation of a holonomic mobile-manipulator system using automatic switching. In *24th Canadian Conference on Electrical and Computer Engineering (CCECE)*, pages 1164–1168, 2011.

- [110] T. Yoshikawa. Manipulability of robotic mechanisms. *The International Journal of Robotics Research*, 4(2):3–9, 1985.
- [111] M. Zambelli, Y. Karayiannidis, and D. Dimarogonas. Posture regulation for unicycle-like robots with prescribed performance guarantees. *IET Control Theory Applications*, 9(2):192–202, 2015.
- [112] Y. Zhang, X. Yan, W. Li, D. Guo, and J. Li. A weighted damping coefficient based manipulability maximizing scheme for coordinated motion planning of wheeled mobile manipulators. In *Proceedings of the World Congress on Intelligent Control and Automation (WCICA)*, pages 3067–3072, 2015.
- [113] J. Zhu, S. Rosset, T. Hastie, and R. Tibshirani. 1-norm support vector machines. *Advances in neural information processing systems*, 16(1):49–56, 2004.
- [114] K. Zimmermann, P. Zuzánek, M. Reinstein, and V. Hlavác. Adaptive traversability of unknown complex terrain with obstacles for mobile robots. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 5177–5182, 2014.