UNIVERSITY OF VERONA

DEPARTMENT OF
Computer Science

GRADUATE SCHOOL OF
Natural Sciences and Engineering
DOCTORAL PROGRAM IN
Computer Science
XXXIV cycle (2018)

Collision Avoidance and Dynamic Modeling for Wheeled Mobile Robots and Industrial Manipulators

S.S.D. ING - INF/05

Coordinator: **Prof. Massimo Merro**

Advisor: **Prof. Riccardo Muradore**
Co-Advisor: **Prof. Luca Di Persio**

Candidate: **Federico Vesentini**

Contents

Chapter 1

Introduction

Collision Avoidance and Dynamic Modeling are key topics for researchers dealing with mobile and industrial robotics. A wide variety of algorithms, approaches and methodologies have been exploited, designed or adapted to tackle the problems of finding safe trajectories for mobile robots and industrial manipulators, and of calculating reliable dynamics models able to capture expected and possible also unexpected behaviors of robots. The knowledge of these two aspects and their potential is important to ensure the efficient and correct functioning of Industry 4.0 plants such as automated warehouses, autonomous surveillance systems and assembly lines.

Collision avoidance is a crucial aspect to improve automation and safety, and to solve the problem of planning collision-free trajectories in systems composed of multiple autonomous agents such as unmanned mobile robots and manipulators with several degrees of freedom. Collision avoidance approaches belong to two main classes: centralized and decentralized.

In centralized collision avoidance approaches for multi-agent systems, safe trajectories are calculated by a global controller, able to access all the information related to the state of the robots that constitute the multi-agent system in order to calculate the commands guaranteeing collision-free paths. In decentralized collision avoidance, every autonomous agent is a decision-making entity that constantly adjust its velocity vector on the basis of the information it perceives from the surrounding dynamic environment, in order to perform collision-free maneuvers. In particular, approaches assume that agents do not communicate with each other.

In certain scenarios such as automated warehouses, the problem of calculating safe trajectories for all agents moving in a dynamic environment can be solved by combining the aforementioned two types of approaches: the first is useful to calculate collision-free maneuvers with respect to expected fixed obstacles (i.e., global planner), e.g. the shelves of the warehouse, while the second handles collision avoidance with respect to other autonomous agents or in presence of unexpected obstacles (i.e., local planner).

The problem of calculating collision-free trajectories for manipulators is slightly different and possibly more complex due to their structural nature: unlike mobile robots, they cannot be approximated as mass-points, therefore

the trajectory planner must take into account the links' size of the robotic arms even for simple kinematic structures.

A rigorous and accurate model explaining the dynamics of robots, is necessary to tackle tasks such as simulation, torque estimation, reduction of mechanical vibrations and design of control law. From the mathematical point of view, the starting point for the derivation of a rigorous dynamic model can be the Newton-Euler equations, Euler-Lagrange model or other more specific theories such as the Lie Group Theory. However, i) a rigorous mathematical model, although accurate, could be very complicated and time-consuming and therefore not suitable for on-board microcontrollers or feedback control laws with small sample times, and ii) in real-life applications there may be effects which cannot be mathematically modelled in advance, whose contributions must be included into the model by exploiting parameter identification techniques a run-time. A proper integration of first-principle model and data-driven models should be carefully analysed.

1.1 Thesis Outline

This thesis is divided into three parts: the first is about path planning while the other two are about dynamic modeling. Figure 1.1 shows the organigram of the thesis.

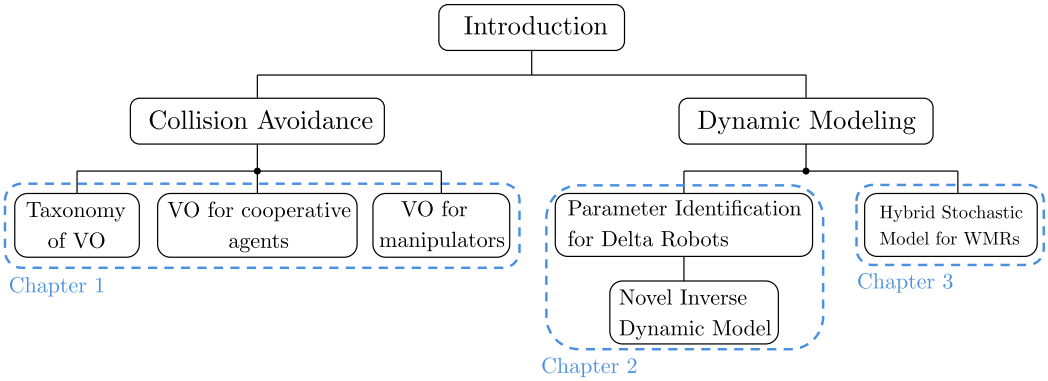


Figure 1.1: Organigram of the thesis.

In particular

- Chapter 2 deals with the trajectory planning and the collision avoidance problem for cooperative holonomic mobile robots and planar manipulators with two revolute degrees of freedom, using the Velocity Obstacle paradigm;
- Chapter 3 describes two different inverse dynamic models for Delta Robots with 3 degrees of freedom, obtained through the application of the *Lagrange D'Alambert principle* exploiting two different ways of treating the kinematic chains that constitute the physical structure. The models are experimentally validated with data coming from a real Delta Robot, made available by the company SIPRO Srl, located in Verona, Italy;

- Chapter 4 deals with a novel hybrid stochastic dynamic model for mobile robots of cart-like type. We started from a basic hybrid two-state deterministic version that allows lateral slippages due to losses of grip with the ground and we developed our proposal, called *Hybrid Stochastic Brownian-Markov* (HSBM) model, by modelling unknown friction and inertial forces with independent Brownian motion. Transitions from *grip state* to *slip state* are driven by a Markov chain, whose transition probabilities depend on the lateral velocity;

- Chapter 5 draws the conclusions and possible future works.

1.1.1 Velocity Obstacle for Mobile Robots and Manipulators

The Velocity Obstacle (VO) paradigm [43, 44], together with its improved versions, has gained popularity during the last two decades. It is a decentralized geometric method for the planning of collision-free trajectories of swarm of unmanned mobile robots, based on the construction of *cones of forbidden velocities* called *Velocity Obstacles* (Fig. 1.2). As a first contribution, we propose a taxonomy of the research field which aims to highlight the evolution of the paradigm in its alternative versions [2, 111, 112, 125, 126, 127] over the years from 1998. The aim is to provide a complete overview of the VO methods for addressing the decentralized collision avoidance problem of multi-agent systems.

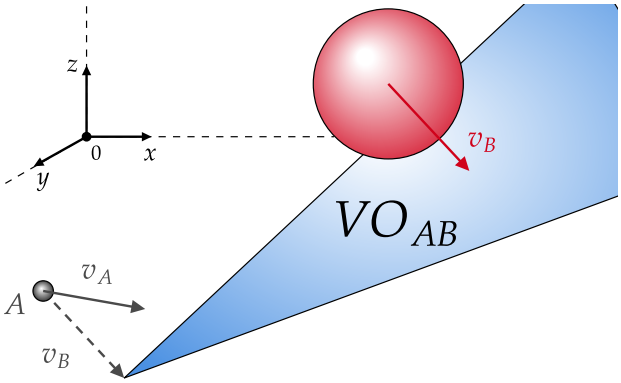


Figure 1.2: The three-dimensional Velocity Obstacle VO_{AB} of agent A induced by the moving obstacle B: the cone in $\mathbb{R}^3$ of forbidden velocities causing collisions between A and B.

As a second contribution, we adapted the *Optimal Reciprocal Collision Avoidance* (ORCA) paradigm [127] to the case of two holonomic planar mobile robots that need to cooperate in order to deliver a long and possibly heavy payload. In particular, we simulated an automated warehouse in which the two constrained agents move from an initial to a final configuration while avoiding collisions with obstacles and other agents. The planner ensuring collision-free trajectories is made of two components: Voronoi Diagrams [94] for planning

trajectories using a map of environment (global planner) and ORCA paradigm for avoiding collision with unexpected obstacles and/or other agents (local planner).

Besides collision avoidance for *unmanned ground vehicles* (UGVs) or *unmanned aerial vehicles* (UAVs), a third contribution of Chapter 2 consists of adapting the Velocity Obstacle paradigm for planning collision-free maneuvers for manipulators, for which other methodologies are commonly used such as *Artificial Potential Fields* (APFs) [70] or Dynamic Movement Primitives (DMPs) [53]. Specifically, we study a simulated RR planar two-link manipulator that has to move without colliding with a moving obstacle in its workspace. The base joint, elbow joint and end-effector of the manipulator are modeled as three holonomic planar agents: the base agent is fixed at the origin of the world reference frame, the elbow agent is forced to moving along a circumference whose center is the origin of the world frame and with radius equal to the length of the first link and the end-effector is able to move in the space. The end-effector has to reach a target position avoiding collisions between the manipulator and an obstacle in the workspace.

1.1.2 Dynamic Modeling for Delta Robots

Delta Robots belong to the class of parallel robots widely used in industrial production processes, mostly for pick-and-place operations. The most relevant characteristics are the high speed and the extremely favorable ratio between the maximum payload and the weight of the robot itself. A reliable dynamic model is needed to implement torque controllers that reduce unnecessary high accelerations and so mechanical vibrations. In Chapter 3 we explain how to derive the inverse dynamic model for the 3-DoF Delta Robot model D3-1200 (Fig. 1.3) made available by SIPRO Srl, a company located in Verona (Italy) active in the field of industrial robotics.

The objective of the research was to use this model for the accurate estimation of the torques to be applied to the robot according to a known trajectory and feed-forward them within the control architecture. Our proposed model enhances the standard inverse dynamic model available in literature [74, 75, 122]. The latter is obtained by a drastic approximation of the kinematic structure of the robot which, on one hand, allows to obtain a simpler mathematical formulation but, on the other hand, causes an underestimation of inertia which implies underestimated torques commanded to the D3-1200

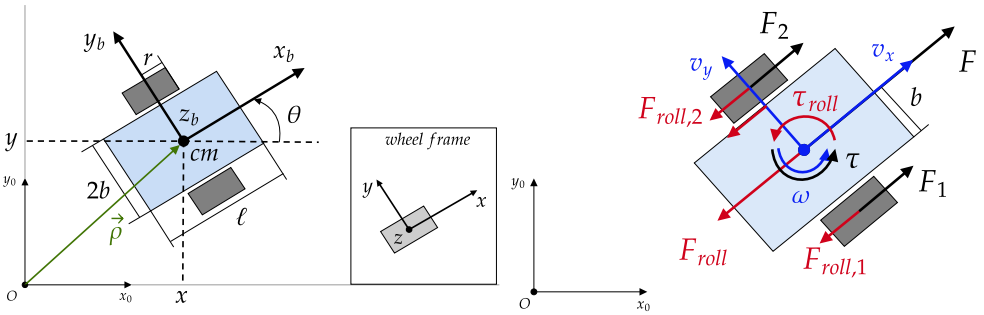
Figure 1.3: A 3-DoF Delta Robot. Courtesy of SIPRO Srl `http://www.sipro.vr.it`.

as well. We mitigate this inconvenience by performing an offline parameter identification, to estimate two parameters $\rho_1^\star$ and $\rho_2^\star$ to be included in the standard model that allow to compensate for most of the missing inertia and gravity. In addition, we proposed a real-time method that estimates the weight of a payload grabbed by the robot, by comparing the estimated torques and the real ones.

In the second part of Chapter 3 we proposed a novel inverse dynamic model without any simplification on the kinematic structure of the robot. We express the passive joints angles as function of the extended generalized coordinates. This mathematical formulation of the inverse model has been compared with the ones available in the literature by comparing accuracy and computational complexity.

1.1.3 Stochastic Modeling of Mobile Robots

Wheeled mobile robots are used in a wide range of applications from automated warehouses to patrolling. A rigorous and accurate model describing their dynamics is important for tackling regulation and tracking problems. In Chapter 4 we provide a novel dynamic model for differential-drive wheeled mobile robots (Fig. 1.4) (WMRs) taking into account displacements along the transverse direction due to losses of adhesion with the ground.



(a) (b)

Figure 1.4: (a) Reference frames of the robot; and (b) force diagram of the robot describing all the forces and torques acting on the WMR.

We started from a deterministic model which consists of a two-state hybrid system [54]: the *grip state*, determined by a set of ordinary differential equations (ODEs) that does not allow lateral motion, and the *slip state*, governed by a set of differential equations allowing lateral motion. Afterwards, we transformed the ODEs characterizing both states of the hybrid system into the corresponding set of Stochastic Differential Equations (SDEs) with nonlinear drift and diffusion terms, driven by independent Brownian motions. Transitions from one state to the other happen according to a homogeneous Markov chain, with transition probabilities depending on the state of the robot (e.g., linear and angular velocities). Brownian noises model the uncertainty due to the action of external forces which, usually, are not known with precision.

Chapter 2

Velocity Obstacle for Mobile Robots and Manipulators

2.1 Related Works

The problem of planning collision-free trajectories for autonomous robots is a crucial aspect of the design of control architectures for multi-agent systems such as automated warehouses, surveillance systems based on unmanned ground or aerial vehicles, automatic transportation lines or, in general, for the safe navigation of autonomous robots. Collisions between autonomous agents or with obstacles moving in their workspace could cause serious problems, e.g. blocking of production lines, workers injuries, pileups, etc. In order to achieve the best levels of production efficiency and safety for human workers, every autonomous entity should be able to accomplish programmed tasks by following collision-free trajectories with respect to other agents moving in the same environment and to unexpected fixed or moving obstacles.

In the current literature, there are two main types of methods that guarantee collision-free trajectories for multi-robot systems in which the agents operate in possibly crowded dynamic environments: *Centralized multi-robot navigation* and *Decentralized multi-robot navigation* methods. In centralized motion planning, all the agents are able to communicate with a common processing unit, sharing information about their state (e.g. current position and velocity) at every instant and also other types of information such as actual goal positions. Several works covering different aspects about multi-robot systems have been proposed, like task allocation [26, 118], formation control of robots [10, 27, 35, 119], and transportation of objects [1, 65, 83, 124]. Recently, [108] proposed a method based on Satisfiability Modulo Convex Optimization (SMC), a combination of satisfiability solving (SAT) and convex programming. The global communication among every agent and the central controller guar-

This chapter is based on the following publications:
▷ N.Piccinelli, F.Vesentini, R.Muradore. *Planning with Real-Time Collision Avoidance for Cooperating Agents under Rigid Body Constraints*, IEEE Design, Automation and Test in Europe Conference and Exhibition (DATE) (2019), pp. 1261-1264.
▷ F.Vesentini, R.Muradore *Velocity Obstacle-based Trajectory Planner for Two-Link Planar Manipulators*, European Control Conference (ECC) 2021, Rotterdam, Netherlands, pp. 687–692.
▷ F.Vesentini, R.Muradore, P.Fiorini *A Survey on Velocity Obstacle Paradigm*, (under review).

antees optimal and safe trajectories [81, 106, 121, 136].

The major advantage of this kind of approaches is that collision-free trajectories are computed offline from the beginning to the end and re-planning is done in an optimal way. Unfortunately, they suffer from several drawbacks: i) they become computationally expensive when dealing with a large number of agents, thus they do not scale well in crowded scenarios, ii) the global communication assumption can be unfeasible in real-world applications, since it would require an exceptionally complex as well as expensive communication network between the global controller and every agent, iii) the global communication system or the central global controller may be affected by disturbances, and finally iv) they assume the perfect knowledge of the environment, so they cannot be applied in presence of unexpected obstacles (e.g. human workers or other objects that may interrupt the navigation of the agents).

Decentralized Collision Avoidance methods has been extensively studied during the last decades. The most important and successful approaches to the problem are represented by *Potential Fields*, [70], *Social Force*, [62], *Dynamic Window*, [46], *Model Predictive Control*, (MPC) [68] and *Velocity Obstacle* [44].

Collision avoidance in Potential Fields, also known as "Artificial Potential Fields" (APFs), is based on representing the robot as a point in the configuration space and then constructing a potential field associated to artificial forces. In such a way the agent is repulsive within a certain neighborhood of every obstacle and globally attracted to its goal position. The original method is designed for fixed obstacles and for robots with two degrees of freedom, however Barraquand and Latombe [12] proposed adapted APFs to robots with several degrees of freedom; and authors in [49, 100, 101] managed to extend the approach also to moving obstacles. The Social Force method can be seen as an adaptation of APFs approach applied to pedestrians , whose dynamics satisfies additional constraints such as the *private sphere* [105].

The Dynamic Window is an approach specifically designed for synchro-drive mobile robots, i.e., a particular type of differential-drive robots with three degrees of freedom. It is based on the optimization of a gain function of the forward and angular velocities of the robot, over a three-dimensional space of velocities that satisfy constraints on both admissibility and time availability of the current velocities of the robot. Brock and Khatib [21] generalized the approach proposing the *Global Dynamic Window*, that allows the execution of high velocity motion for non-holonomic robots in unknown and dynamic environments. Furthermore, Ogren and Leonard [86] addressed the convergence properties of the method, in order to solve known deadlock situations.

The Model Predictive Control approach consists in solving a series of constrained optimal control problems over discrete time intervals. It minimizes a convex cost function over the space of inputs of an autonomous robot. The constraints of these minimization problems are given by the dynamic model of the robot and possibly other limitations about its state. Remarkable works about sufficient conditions ensuring the stability of the closed-loop control system can be found in [82, 83].

These approaches are in general relatively simple from a theoretical point

of view and highly adaptable to dynamic scenarios and agents' model. Their major drawbacks are essentially the difficulty to handle heterogeneous agents and, since in some cases the behaviours of the agents are regulated by a cost function (e.g, a function used to prioritize stopping policies in place of avoidance maneuvers), they may require extra and possibly offline non-trivial calculations in order to tune the related parameters. Keviczky et al. [68] used an MPC technique for formation control and collision avoidance of UAVs, see Figure 2.1a, forced to fly at certain fixed altitude. Park et al. [90] proposed an obstacle avoidance scheme for UGVs, see Figure 2.1b, where safe trajectories are generated by adopting a non-linear MPC approach, in order to better approximate the dynamics of each vehicle. Similarly, Jiang et al. [67] designed an MPC scheme based on convex quadratic programming (CQP) that provides safe trajectories for autonomous vehicles which dynamics is described by a linear time-varying differential equation, obtained by performing a linear approximation of a non-linear time-varying model.

(a) (b)

Figure 2.1: An octa-copter Unmanned Aerial Vehicle (UAV) (a) and a *Jackal* Unmanned Ground Vehicle by Clearpath Robotics (b) `https://clearpathrobotics.com/` .

Another popular decentralized method for collision avoidance in multi-agent scenarios is represented by the Velocity Obstacle (VO) paradigm, [42, 44] and its enhanced formulations. VO is a geometric approach for the navigation of a single agent that has to avoid collisions with several moving agents and obstacles, arising from the concept of "Configuration Space Obstacle" by [80] extended to dynamic environments. The later formulations have been designed to extend the paradigm to dynamic multi-agent environments, [125], to implement optimal velocity selection for every agent via linear optimization, [127] to include a constraint of maximum acceleration for the agents, [126], to model sensors uncertainty, [112] and to extend the paradigm to non-holonomic car-like agents [2]. Generally speaking, VO-based approaches assume that agents do not communicate, but only detect each other thanks to their on-board sensing system. Furthermore, each agent is characterized by five values: current position, current velocity, radius, initial and goal position. The first three can be sensed by every agent, and the collision avoiding maneuvers are computed using these measurements at every cycle of sensing-and-acting of the algorithm.

2.2 Velocity Obstacle paradigm

The Velocity Obstacle (VO) paradigm [44] is a geometric construction used to determine which velocity vectors cause collisions between a robot and other moving obstacles or agents. VO has been formalized in 1998 and its popularity grew significantly over the next decades, among researchers who deal with online decentralized collision avoidance for autonomous robots.

The dynamic environment in VO consists of an agent A and moving obstacle B_i, $i = 1, ..., m$, assumed to be holonomic and disc-shaped. The agent is characterized by its position x_A, radius r_A and velocity v_A, and is able to detect position x_{B_i}, radius r_{B_i} and velocity v_{B_i} of every obstacle around it. Velocity Obstacle represents the dynamic environment into the velocity space that is the tangent bundle $T\mathbb{R}^2$ of $\mathbb{R}^2$. A motion in the velocity space, for agent A, is given by the pair

$$(x_A(t), v_A(t)) \in T\mathbb{R}^2 \simeq \mathbb{R}^2 \times \mathbb{R}^2.$$

For every couple (A, B_i), let

$$v_{AB_i} = v_A - v_{B_i} \quad i = 1, ..., m,$$

be the relative velocity of A with respect to B_i.

The corresponding relative trajectory in the velocity space is defined as

$$\mathrm{trj}_{AB_i} = \{(x, \dot{x}) : x(t) = x_{AB_i}, \dot{x}(t) = v_{AB_i}\}.$$

At the core of Velocity Obstacle there is this statement [43, 44]: *A collision between A and B_i occurs if and only if the relative velocity v_{AB_i} does not change and* $\mathrm{trj}_{AB_i} \cap B_i \neq \emptyset$. The set of all relative velocities for which the statement is satisfied is defined as

$$CC_{AB_i} \triangleq \{v_{AB_i} : \mathrm{trj}_{AB_i} \cap B_i \neq \emptyset\} \tag{2.1}$$

and called *Relative Collision Cone* of A induced by B_i. Geometrically, it can be constructed on the two-dimensional space by reducing the agent A to a point $\hat{A}$ and enlarging the obstacle B_i by the radius of A, obtaining $\hat{B}_i$.

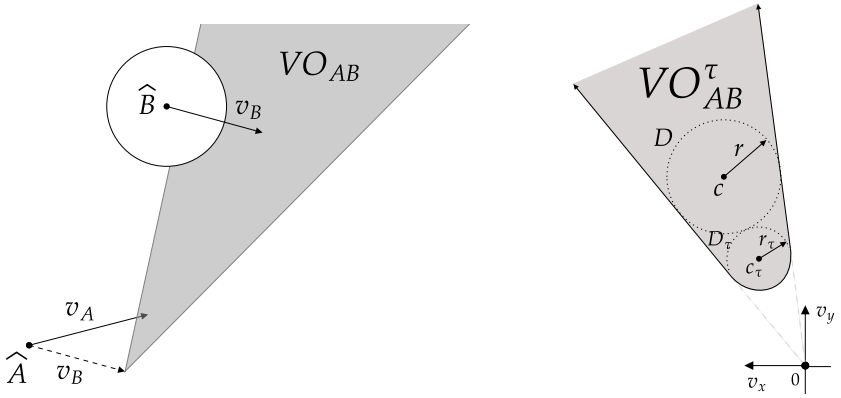
The Relative Collision Cone is delimited by two straight lines tangent to $\hat{B}$ whose intersection point is $\hat{A}$. The *Absolute Collision Cone* of A with respect to B_i is defined as follows

$$CC_{A_i} \triangleq \{v_A : v_{AB_i} \in CC_{AB_i}\} \tag{2.2}$$

and it is obtained by translating the relative collision cone CC_{AB_i} by v_A via Minkowski sum

$$CC_{A_i} = CC_{AB_i} \oplus v_{B_i}.$$

Agent A does not collide with B_i as long as its velocity vector remains outside CC_{A_i}. The absolute collision cone can be thought as Velocity Obstacle VO_A for A induced by B_i, see Figure 2.2a.



(a) (b)

Figure 2.2: The Velocity Obstacle of A induced by B in the original paradigm (a) and the truncated equivalent cone (b).

In case of multiple obstacles, the union of all CC_{A_i} defines the *Multiple Velocity Obstacle* for agent A

$$MVO_A \triangleq \bigcup_{i=1}^{m} CC_{A_i}. \tag{2.3}$$

Agent A must select a velocity vector v_A^{new} such that

$$v_A^{new} = \underset{v \in V \setminus MVO_A}{\arg\min} \; ||v_A^{pref} - v||_2,$$

where V is the set of admissible velocities for agent A and v_A^{pref} is the preferred velocity, i.e., the maximum velocity leading A to its goal along a straight line (shortest path). This optimization has to be performed at every sensing-and-acting cycle.

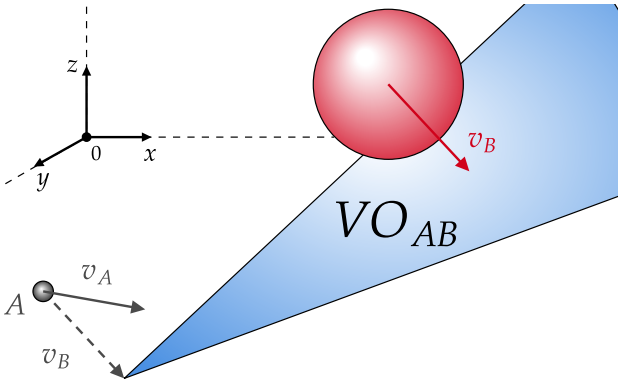


Figure 2.3: Three-dimensional Velocity Obstacle VO_{AB} of A induced by B.

VO guarantees collision-free trajectories also for mobile robots able to move in the three-dimensional space, where a motion for agent A is represented by the pair

$$(x_A(t), v_A(t)) \in T\mathbb{R}^3 \simeq \mathbb{R}^3 \times \mathbb{R}^3.$$

Robot A and obstacles B_i are represented as spheres of radius r_A and r_{B_i} instead of circles. The velocity cones in (2.1), (2.2) and (2.3) are now three-dimensional cones of forbidden velocities for A, see Figure 2.3.

Modern formulations of VO are based on truncated cones, denoted with $VO^{\tau}_{A_i}$ where $\tau \in \mathbb{R}_{>0}$ is called *truncation factor*. The agent A that selects a velocity vector v^{new}_A such that $v^{new}_A \notin VO^{\tau}_{A_i}$ avoids collisions with the moving obstacle B_i until time τ [127]. The original formulation of VO endowed with truncated cones takes the name of *Finite-Time Velocity Obstacle* (FVO) [59]. In Figure 2.2b it is possible to see a representation of VO^{τ}_{AB}. The circle D, with center $c = x_B - x_A$ and radius r, is B represented in the velocity space of A, while D_τ is the cut-off circle of radius $r_\tau = r/\tau$ and center $c = (x_B - x_A)/\tau$. Note that

$$\lim_{\tau \to \infty} VO^{\tau}_{A_i} = VO_{A_i}$$

since FVO converges to VO for $\tau \to \infty$, because the cut-off circle D_τ collapses to a single point.

2.3 Taxonomy of Velocity Obstacle methods

The major contributions that extended the original Velocity Obstacle paradigm over the last 25 years are shown in Figure 2.4. Starting from the original Velocity Obstacle (VO, [44]), around 2008-2009 there is the first cluster of important extensions given by *Reciprocal Velocity Obstacle* (RVO, [125]), *Finite-Time Velocity Obstacle* (FVO, [59]) and *Optimal Reciprocal Collision Avoidance* (ORCA, [127]). The first work [125] introduced the reciprocal collision avoidance, i.e., the autonomous robots share responsibility in performing collision avoidance maneuvers with respect to each other, improving the algorithm performance in multi-agent systems. The second work [59] focused on avoiding sub-optimal collision-free maneuvers between agents by introducing the concept of truncated collision cones, that allow to impose a policy among agents and moving obstacles. The last work [127] of the group incorporates reciprocal collision avoidance and truncated cones with linear constraints optimization for collision-free velocity optimal selection. The *Generalized Velocity Obstacle* (GVO,[130]) in the same year extended the original paradigm to car-like robots, to overcome the holonomic robots assumption. The *Non-Linear Velocity Obstacle* (NLVO,[107]) is a direct extension of the original paradigm based on warped collision cones designed specifically to deal with obstacles that follow non-linear known trajectories. The *Hybrid Reciprocal Velocity Obstacle* (HRVO, [112]) extends the RVO in order to solve the problem of reciprocal dances, i.e., an undesired phenomenon that induces two agents to perform oscillatory trajectories while trying to avoid each other. The method is based on enlarged collision cones that allow a robot to identify which is the most natural direction to take while avoiding collision with another robot. The HRVO paradigm solved the reciprocal dances, but the obtained trajectories may be non-smooth in crowded dynamic environments. The ORCA paradigm, on this reguard, is able to produce smooth and non-oscillating trajectories also

in crowded environments, but it has the problem of deadlocks, i.e. an event for which the linear optimization that characterized the method leads an autonomous agent to select a collision-free velocity vector equal to zero, forcing him to stop and never reach the target.

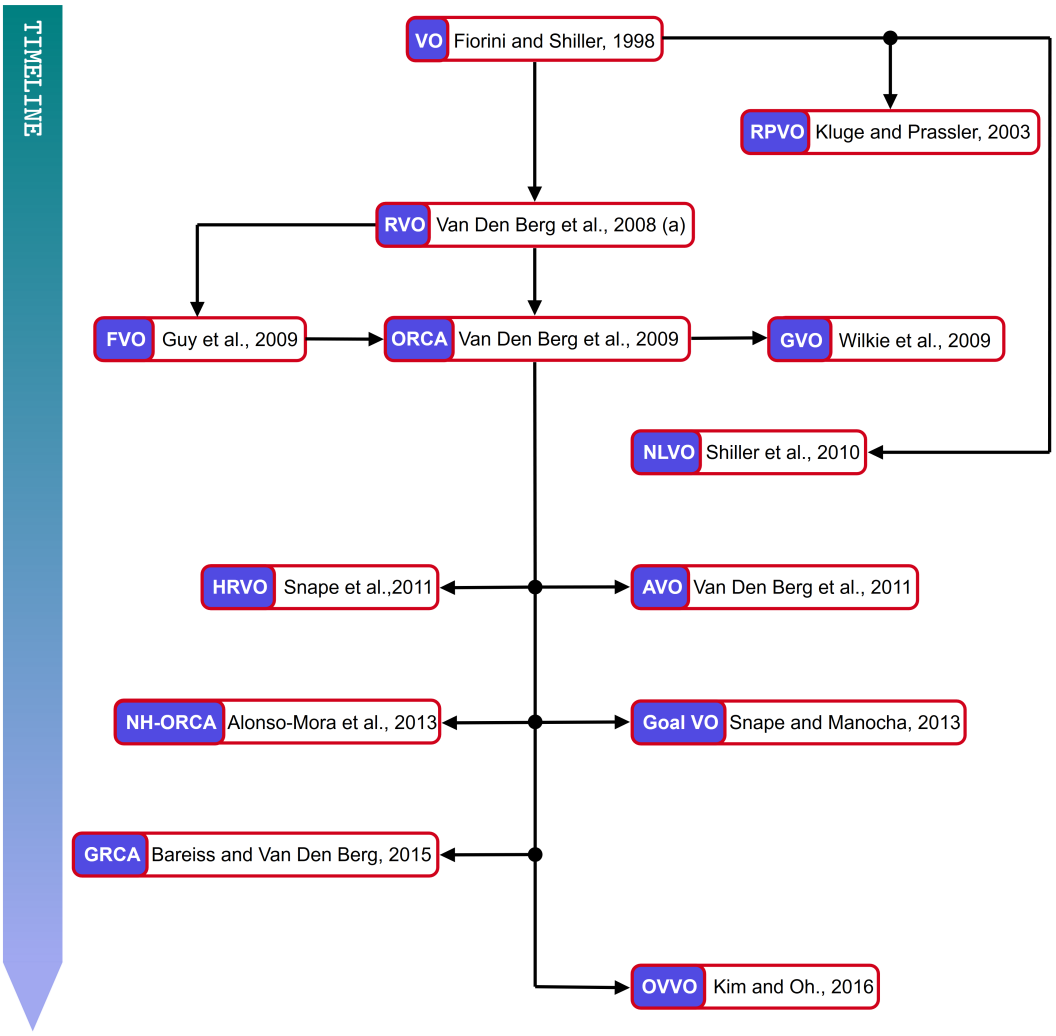


Figure 2.4: Taxonomy showing the evolution of the paradigm.

A solution to this problem, based on *dummy goals*, is proposed in [14]. *Acceleration Velocity Obstacle* (AVO,[126]) includes bounds on accelerations for mobile robots in the ORCA paradigm. This allows to cope with instantaneous changes of velocity that are non realistic in real world applications. The year 2013 marks the arrival of two other important contributions: *Optimal Reciprocal Collision Avoidance for Multiple Non-holonomic Robots* (NH-ORCA,[2]) and *Goal Velocity Obstacles for Spatial Navigation of Multiple Autonomous Robots or Virtual Agents* (Goal VO,[111]). The first algorithm directly extends ORCA to the case of differential-drive robots, the most common type of autonomous robots operating in automatic warehouses and other industrial facilities; the second one exploits the collision cones not only to model velocities that lead to collision, but also those that lead to a desired and possibly moving goal regions. This is justified by the fact that in several applications many

robots may have to share the same goal position, such as batteries recharge spots.

In 2015, [11] generalized the problem of collision avoidance in order to propose a unified framework where vehicles with different non-linear dynamics can be considered at the same time, since until then only homogeneous multi-agent systems where considered.

Finally, [71] proposed the last direct refinement to the original Velocity Obstacle algorithm that allows the robots to prioritize stopping in place of avoidance maneuvers. The motivation behind this contribution is that when obstacles are very fast, the most natural action is to stop the robot and wait until the obstacles have gone away, instead of performing hazardous collision-free maneuvers at all costs.

2.3.1 Reciprocal Velocity Obstacle

Van Den Berg et al. [125] provided one of the first adaptation of VO to multi-agent scenarios. In this formulation all the agents are meant to share the responsibility to avoid collision. Agents are decision-making entities with the same collision avoidance policy.

Consider an agent A that has to avoid collisions with another agent B travelling in the same workspace. At the current time t, A moves with velocity v_A and computes the velocity obstacle VO_{AB} with respect to B. Also B, moving with velocity v_B computes a velocity obstacle cone VO_{BA}. If the agents are on a collision course, it means that at time t we have that $v_A \in VO_{AB}$ and $v_B \in VO_{BA}$. In order to perform a correct maneuver, at time $t + \Delta t$, A must select a new velocity vector $v_A^{new} = v_A + w$ where $w \in \mathbb{R}^2$ is a vector such that $v_A + w \notin VO_{AB}$. Agent B does the same for symmetry (see Lemmas 2,3 and 7 of [125]) by selecting $v_B^{new} = v_B + w \notin VO_{BA}$.

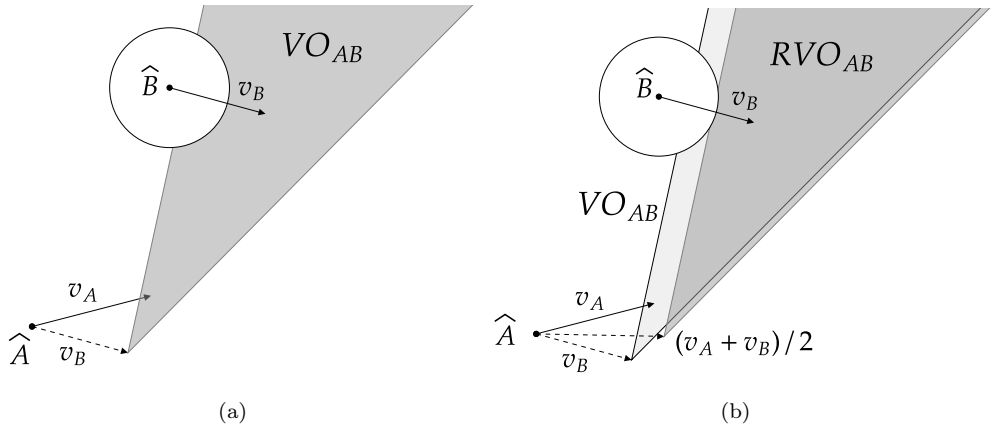


Figure 2.5: Velocity Obstacle VO_{AB} (light grey cone) compared to the Relative Velocity Obstacle RVO_{AB} (dark grey cone). Note that the apex of RVO_{AB} on the tip of $(v_A+v_B)/2$ vector.

Such velocity selection produces sub-optimal collision avoidance maneuvers, since the two agents would move further than necessary. To restore optimality, we must impose that A selects $v_A^{new} = v_A + w/2$ provided that B does

the same by choosing $v_B^{new} = v_B + w/2$. This means sharing the responsibility among pair of agents in computing collision avoidance maneuvers.

Formally, this means that A must select a new velocity vector whose tip is outside of the *Reciprocal Velocity Obstacle* of A with respect to agent B, given by

$$RVO_{AB} = \{v_A^{new} \,|\, 2v_A^{new} - v_A \in VO_{AB}\}\,.$$

From a geometric point of view RVO_{AB} is a cone equal to VO_{AB}, but translated by $(v_A + v_B)/2$ instead of v_B as shown Figure 2.5.

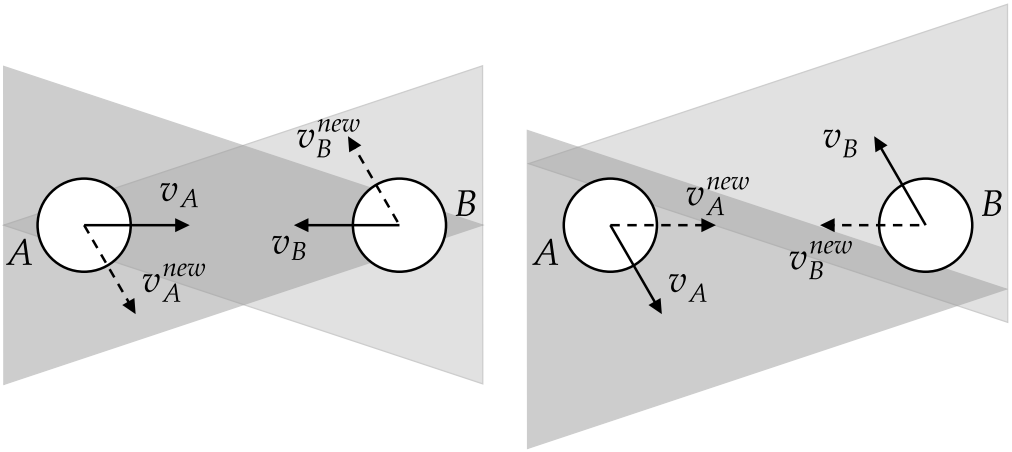
Let us consider a multi-agent scenario where a certain number $n \in \mathbb{N}$ of agents $A_1, ..., A_n$ move in the same workspace. Suppose that every agent A_i has a preferred velocity $v_{A_i}^{pref}$, i.e. the velocity vector leading to the goal position $x_{g_{Ai}}$ in absence of obstacles. At every cycle of the algorithm, A_i selects a new velocity vector such that

$$v_{A_i}^{new} = \underset{v \notin RVO_{A_i}}{\arg\min} ||v - v_{A_i}^{pref}||_2,$$

where RVO_{A_i} is given by

$$RVO_{A_i} = \bigcup_{j \neq i}^{n} RVO_{A_i A_j}.$$

The Relative Velocity Obstacle paradigm represents the most straightforward extension of the original Velocity Obstacle to multi-agent dynamic environments, providing on-line safe navigation trajectories for every decision making entity.



(a) (b)

Figure 2.6: Reciprocal dance between A and B. (a) Each one chooses the closest velocity to their preferred velocity outside the VO induced by the other, then (b) at the next sense-and-acting step each robot has attained its new velocity leaving the other one outside of the current velocity obstacle, that will be selected again in the following step.

However, it has the issue of *reciprocal dances* see Figure 2.6, i.e. an oscillating behaviour occurring when two agents A and B navigate one toward the

other. Reciprocal dances are due to the fact that once A and B have updated their current velocities with the new ones and moved to the new positions, x_A^{new} and x_B^{new}, the old velocities might return to be collision free in the successive cycle of the algorithm, making agents A and B adopting them once again and so on. This fact could bring to a deadlock. In Section 2.3.6, we will see how this problem has been solved by [112].

2.3.2 Optimal Reciprocal Collision Avoidance

Van Den Berg et al. [127] presented the Optimal Reciprocal Collision Avoidance (ORCA) improving the original Velocity Obstacle (Section 2.2) and the Reciprocal Velocity Obstacle (Section 2.3.1). The selection of collision avoidance velocity for every agent v_A^{new} is carried out by solving a linear programming problem in which the constraints are the straight lines defining the half-planes of feasible velocities.

The algorithm can be described as follows. Suppose agent A, moving with velocity v_A, has to avoid collisions with B moving with velocity v_B. The original VO to constructs the cone VO_{AB}^{τ} of all velocities for A that can cause a collision with B within a certain time horizon $\tau > 0$,

$$VO_{AB}^{\tau} \triangleq \{v \,|\, \exists\, t \in [0,\tau],\, tv \in \mathcal{D}(x_B - x_A, r_A + r_B)\}$$

where x_i and r_i, $i \in \{A, B\}$, denote the positions and radii of the agents, respectively. $\mathcal{D}$ is a disk centered in $x_B - x_A$ of radius $r_A + r_B$ as depicted in Figure 2.2b.

If A is on a collision course with B, it means that $v_{AB} \in VO_{AB}^{\tau}$. Let w be the vector from v_{AB} to the closest point on the boundary, ∂VO_{AB}^{τ}, of the Velocity Obstacle

$$w = (\underset{v \in \partial VO_{AB}^{\tau}}{\arg\min} ||v - v_{AB}||_2) - v_{AB},$$

and n be the outward normal vector at point $v_{AB} + w \in \partial VO_{AB}^{\tau}$. Then, the set of all collision free velocities for A is defined by the half-plane pointing in the direction n originating at the point $v_A + w/2$,

$$ORCA_{AB}^{\tau} \triangleq \{v | (v - (v_A + w/2) \cdot n \geq 0\}.$$

The set $ORCA_{BA}^{\tau}$ for B is defined in the same manner, see Figure 2.7. If A has to avoid collisions with a certain number of other agents $B_1, ..., B_m$, then the set of all the permitted velocities for agent A is given by

$$ORCA_A^{\tau} \triangleq \mathcal{D}(0, v_A^{max}) \cap \bigcap_{i=1}^{m} ORCA_{AB_i}^{\tau}, \tag{2.4}$$

where $\mathcal{D}(0, v_A^{max})$ is a disc centered in the origin with radius given by of the maximum speed that an agent can reach. The new collision-free velocity for agent A, is

$$v_A^{new} = \underset{v \in ORCA_A^{\tau}}{\arg\min} ||v - v_A^{pref}||_2.$$

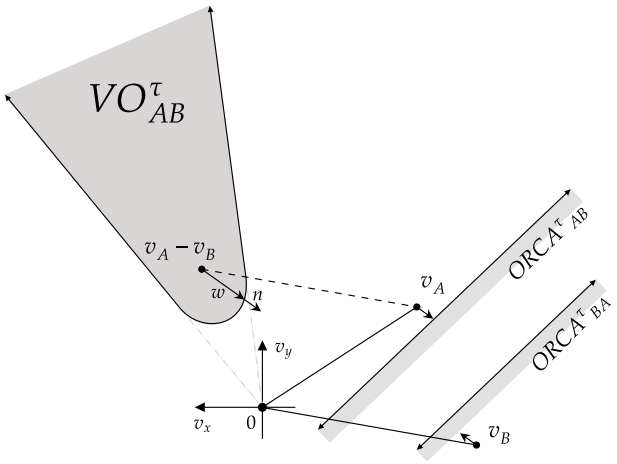
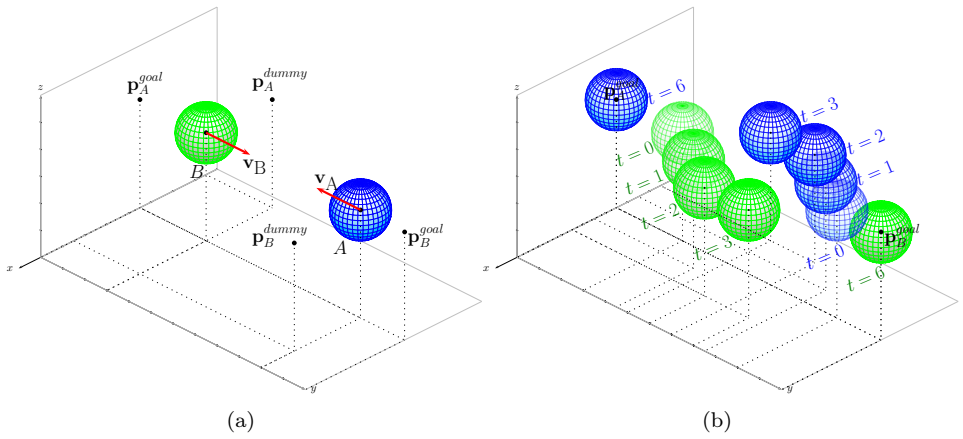


Figure 2.7: The $ORCA^{\tau}_{AB}$ half-plane for A induced by B and its counterpart, $ORCA^{\tau}_{BA}$, for B induced by A.

$ORCA^{\tau}_{A}$ is a convex region and the straight lines that defines the half-planes are the constraints in a linear programming problem, whose solution is v^{new}_{A} at every cycle of the algorithm.

This solution is able to provide the optimal and non-oscillating safe trajectories. Unfortunately two agents A and B may get stuck in a deadlock situation while navigating along a straight line, one toward the other, having each one the purpose to reach a goal position located beyond the other on a straight line. In [14] the problem is solved by adopting dummy goals, see Figure 2.8, leading the robots temporarily away from the aforementioned straight line and this resolving the deadlock situation.



(a) (b)

Figure 2.8: Dummy goals p^{dummy}_{A} and p^{dummy}_{B}, for either the two agents A and B moving one towards the other (a) and the time-lapse of the collision avoidance maneuver resolving the deadlock (b).

2.3.3 Generalized Velocity Obstacle

Wilkie et al. [130] presented a generalization of the original VO paradigm to non-holonomic car-like robots driven by some controls u belonging to a control set $\mathcal{U}$. Consider a car-like robot A that has to avoid collisions with a moving obstacle B. The approach identifies all the controls u that cause a collision between the robot and B in the future. Let u^{pref} be the preferred controls that would lead the robot to its goal in absence of obstacles. The set of controls $\mathcal{U}_c \subset \mathcal{U}$ that will cause a collision at time $t > 0$ with a moving obstacle B is defined as

$$\mathcal{U}_c \triangleq \{u \,|\exists\, t > 0 \, s.t. \, ||x_A(t,u) - x_B(t)||_2 < r_A + r_B\},$$

where $x_A(t,u)$ is the position of agent A at time t given the control u and $x_B(t)$ is the position of obstacle B at time t. The first step is to find the minimum t_{min}, solving $\frac{d}{dt}||x_A(t,u) - x_B(t)||_2 = 0$ in order to check if the actual control for robot A is collision free or not. If a collision is forseen, then a new control has to be selected according to

$$u = \underset{u' \notin VO_{AB}}{\arg\min} ||u^{pref} - u'||_2,$$

as the closest command to u^{pref} that does not belong to VO_{AB}.

2.3.4 Non-Linear Velocity Obstacle

Shiller et al. [107] introduced the Non-Linear Velocity Obstacle (NLVO) as a modification of the original VO paradigm for robot A that travels in a dynamic environment populated by moving obstacles B_i that follow a general non-linear *known* trajectory $c(t)$, for $t \in [0, +\infty)$.

The paradigm is stated considering A as a single point, but it can be extended also to vehicles characterized by more complex kinematics or dynamics. Given the current time $t_0 \geq 0$, the infinite-time horizon non-linear velocity obstacle $NLVO^{\infty}_{t_0}$ is given by

$$NLVO^{\infty}_{t_0}(t) \triangleq \bigcup_t NLVO(t), \tag{2.5}$$

where $NLVO(t)$ is the set of all velocities v_A of A that would cause a collision with $B(t)$, where $B(t)$ denotes the region of the plane that at time t is occupied by B. If $p \in B(t)$, a velocity that would cause a collision with p at $t > t_0$ can be expressed in closed form by

$$v = \frac{c(t) + p_r}{t - t_0} = H_{A,k}(c(t) + p_r), \quad k = \frac{1}{t - t_0}, \tag{2.6}$$

where p_r is the vector connecting $c(t)$ to p and $H_{A,k}$ is the homothetic transformation, centered in A and scaled by k, of the point $c(t) + p_r$.

$NLVO(t)$ is then given by applying (2.6) to every point of $B(t)$

$$NLVO(t) = H_{A,k}(B(t)), \quad k = \frac{1}{t - t_0}.$$

Contrary to what happens in the original VO, where $VO_A(t)$ is a proper cone, the $NLVO_A(t)$ appears to be a warped cone originated from A.

The NLVO paradigm can be improved by considering only collisions that may occur within a certain finite time-horizon τ: it will not construct (2.5), but an analogous truncated warped cone $NLVO^{\tau}_{t_0}(t)$. The optimal value for τ is found by solving a minimization problem of the form

$$\min \int_{t_0}^{t_h} 1\, dt$$

subjected to

1. initial conditions $x(t_0)$ and $\dot{x}(t_0)$,
2. terminal condition $x(t_h) \notin NLVO^{\infty}_{\tau}(t)$,
3. the vehicle dynamics $\ddot{x} = f(x, \dot{x}, u)$, $u \in \mathcal{U}$.

A robot A characterized by some dynamics selects a new velocity $v_A(t_0 + \Delta t)$ belonging to $NLVO^{\tau}_{t_0}(t)^C \cap ACV(t)$, where

$$ACV(t) \triangleq \{v \,|\, v = v(t) + u\Delta t,\, u \in \mathcal{U}\}$$

is the set of all *attainable Cartesian velocities* and $NLVO^{\tau}_{t_0}(t)^C$ is the complementary set of $NLVO^{\tau}_{t_0}(t)$. The shape of ACV depends on the dynamics itself. The major disvantage of this approach is that the planner is able to guarantee reachability of the goal position in optimal time only in presence of just one moving obstacle.

2.3.5 Acceleration Velocity Obstacle

Van Den Berg et al. [126] improved ORCA by considering mobile robots subjected to acceleration constraints. The introduction of this type of constraints is motivated by the necessity to model robot and other autonomous vehicle moving at high speed (e.g., aerial vehicles like UAVs).

Consider a robot A that has to avoid collisions with another robot B within time τ. At every sensing-and-acting cycle, the idea is to substitute the collision velocity cone VO^{τ}_{AB} with a new one called *Acceleration-Velocity Obstacle for A induced by B within time* τ, $AVO^{\delta,\tau}_{AB}$, where δ is a control parameter whose dimension is time. The acceleration $a_A(t)$ at time t is proportional to the difference between the new velocity v^{new}_A and the velocity $v_A(t)$ at time t, i.e.

$$a_A(t) = \frac{v^{new}_A - v_A(t)}{\delta}.$$

The solution of the previous differential equation is given by

$$v_A = v^{new}_A - e^{-t/\delta}\left(v^{new}_A - v_{A,0}\right),$$

where $v_{A,0}$ is the velocity of A at $t = 0$. Integrating this solution gives the position of the agent at time t

$$x_A = x_{A,0} + v_A^{new} t + \delta \left(e^{-t/\delta} - 1\right) (v_A^{new} - v_{A,0}) \tag{2.7}$$

where $x_{A,0}$ is the position of A at time $t = 0$. Let $x_e^{\tau,\delta} \triangleq \left(e^{-t/\delta} - 1\right)$, for simplicity, and let x_{AB}, v_{AB} and a_{AB} be the relative position, velocity and acceleration of A with respect to B. A collision occurs at time t if $||x_{AB}||_2 < r_{AB} = r_A + r_B$. Exploiting (2.7), such inequality can be re-written as

$$||x_A + v_A^{new} t + \delta x_e^{\tau,\delta} (v_A^{new} - v_A)||_2 < r_{AB}.$$

It is possible to re-arrange this inequality into the following one

$$||v_{AB}^{new} - \frac{\delta x_e^{\tau,\delta} v_{AB} - x_{AB}}{t + \delta x_e^{\tau,\delta}}||_2 < \frac{r_{AB}}{t + \delta x_e^{\tau,\delta}},$$

which defines the disc of all relative velocities v_{AB}^{new} that cause a collision between A and B at time t. The $AVO_{AB}^{\delta,\tau}$ can be defined as

$$AVO_{AB}^{\delta,\tau} \triangleq \bigcup_{0<t\leq\tau} \mathcal{D}\left(\frac{\delta x_e^{\tau,\delta} v_{AB} - x_{AB}}{t + \delta x_e^{\tau,\delta}}, \frac{r_{AB}}{t + \delta x_e^{\tau,\delta}}\right),$$

where $\mathcal{D}(\cdot,\cdot)$ is a disc whose center and radius are the first and second argument, respectively. In order to avoid collisions with B, robot A must select a new velocity vector outside of $AVO_{AB}^{\delta,\tau} \oplus v_A$, where $\oplus$ denotes the Minkowski Sum. The approach is designed for holonomic disc-shaped autonomous robots, but it can be extended also to robots that have to satisfy kinematic constraints. However since robot A is in any case subjected to acceleration constraint, the set of collision-free velocities for A, CA_A, is constructed as follows

$$CA_A \triangleq \mathcal{D}(v_A, \delta a_A^{max}) \setminus \bigcup_B AVO_{AB}^{\delta,\tau} \oplus v_A.$$

Given a preferred velocity vector v_A^{pref} for robot A, the algorithm will select a new velocity such that

$$v_A^{new} = \arg\min_{v \in CA_A} ||v - v_A^{pref}||_2.$$

2.3.6 Hybrid Reciprocal Velocity Obstacle

Snape et al. [112] formalized the so called Hybrid Reciprocal Velocity Obstacle (HRVO) with the aim to solve the reciprocal dance problem of RVO seen in Section 2.3.1, by forcing a collision avoidance maneuver policy among all the agents enlarging on purpose a side of every absolute collision cone.

Consider a certain number $m \in \mathbb{N}$ of autonomous agents $A_1, ..., A_m$ moving in the same workspace. The first three steps to construct the $HRVO$ of A_i induced by A_j are:

- (i) compute $VO_{A_iA_j}$ as in Section 2.2,
- (ii) compute the corresponding $RVO_{A_iA_j}$ as in Section 2.3.1 and
- (iii) draw the center line CL splitting $RVO_{A_iA_j}$ into two identical halves.

Suppose now that the tip of the current velocity vector of agent A_i, v_{A_i}, is on the right of CL, Figure 2.9 (a). It means that A_i will perform a collision maneuver to the right side of the other agent A_j. $HRVO_{A_iA_j}$ is defined as the cone which apex is the intersection point between the left edge of $VO_{A_iA_j}$ and the prolungated right edge of $RVO_{A_iA_j}$.

The exact same construction can be done for symmetry, when the tip of v_A is assumed to be on the left of CL. Figure 2.9 summarizes the $HRVO$ cone construction from agent A_i perspective.

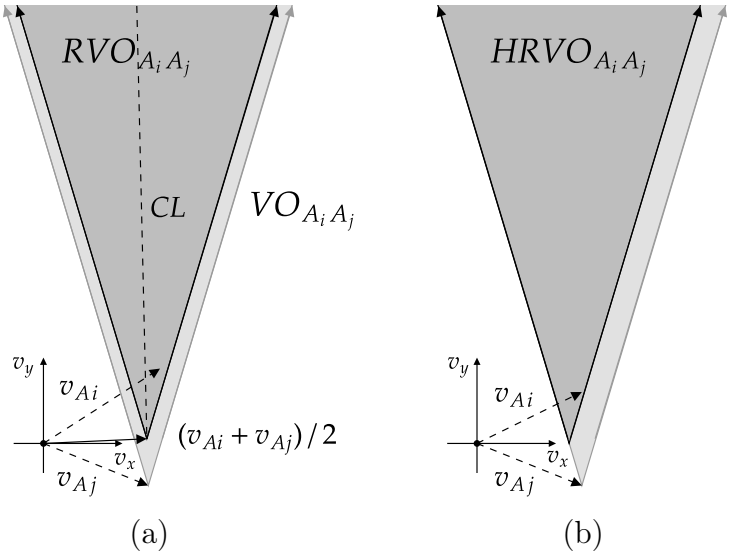


Figure 2.9: $HRVO_{A_iA_j}$ construction. The agent A_i (a) computes $RVO_{A_iB_j}$ induced by A_j and verifies that the tip of v_A is on the right of the center-line CL, then (b) calculates $HRVO_{AB}$ by intersecting the right edge of $RVO_{A_iB_j}$ with the left edge of $VO_{A_iB_j}$.

Finally, supposing that in the environment there are also a certain number $h \in \mathbb{N}$ of static obstacles $O_1, ..., O_h$, at every cycle of the algorithm, A_i selects a new velocity vector such that

$$v_{A_i}^{new} = \underset{v_{A_i} \notin HRVO_{A_i}}{\arg\min} ||v_{A_i} - v_{A_i}^{pref}||_2,$$

where $HRVO_{A_i}$ is given by

$$HRVO_{A_i} \triangleq \bigcup_{j \neq i}^{n} HRVO_{A_iA_j} \cup \bigcup_{k=1}^{h} HRVO_{A_iO_k}.$$

Computing $HRVO_{A_i}$ instead of RVO_{A_i} ensures that there are no reciprocal dances between agents.

The major drawback is the increased computational burden: if RVO paradigm requires that every agent A_i computes one collision cone for every A_j, $i \neq j$, then HRVO has to compute an additional cone for every other agent.

2.3.7 Non-holonomic Optimal Reciprocal Collision Avoidance

In [2], the authors extended ORCA to non-holonomic robots of the cart-like type. A cart-like vehicle has three degrees of freedom: two variables (x, y) for the position and one θ for the orientation. Its kinematics is

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta \\ \sin\theta \\ 0 \end{bmatrix} v + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \omega, \tag{2.8}$$

where v is the linear control velocity and ω is the angular control velocity. The idea behind the approach is that a robot A with kinematics described by (2.8) is able to track the motion of an "ideal" holonomic robot, namely a holonomic trajectory, within a certain bounded tracking error ϵ_H. Referring to Figure 2.10, if the current velocity of robot A is fixed to $\boldsymbol{v_H} = (v_H \cos\theta_H, v_H \sin\theta_H)$, where v_H is the speed of such ideal holonomic robot, then the maximum tracking error for the aforementioned holonomic trajectory is

$$\epsilon_H^2 = (v_H\, t - R\sin\theta_H)^2 + (R(1-\cos\theta_H))^2,$$

$$= v_H^2\, t^2 - \frac{2v_H\, t\sin\theta_H}{\omega} v + \frac{2(1-\cos\theta_H)}{\omega^2} v^2,$$

where R is the curvature radius of the trajectory followed by A and $t = k\Delta t$, with k the iteration index and Δt the time-step.

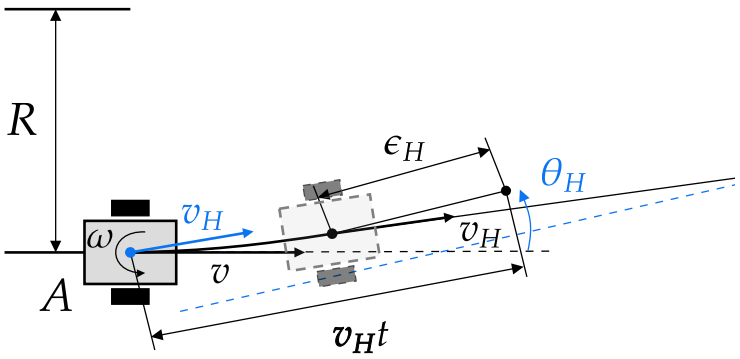


Figure 2.10: Geometric interpretation of the tracking error ϵ_H. $\boldsymbol{v_H}$ and θ_H (blue) defines the holonomic trajectory to be tracked by the robot A.

The set $\mathcal{D}(0, v_A^{max})$ for agent A defined in Section 2.3.2 is substituted here by another set denoted by S_{FHV}, that stands for set of feasible holonomic velocities under the kinematic constraint. Let S_{NHC} be the set of non-holonomic controls, the feasible holonomic velocities are defined as

$$S_{FHV} = \{\boldsymbol{v_H} : \exists (v(s), \omega(s)) \in S_{NHC}, ||x + s \cdot \boldsymbol{v_H} - \hat{x}^k(s)||_2 \leq \epsilon, \forall \tau \geq 0\},$$

where x is the robot current position, $\hat{x}^k(s)$ is the expected robot position at time $k\Delta t + s$ if the controls $(v(s), \omega(s))$ are applied at time $k\Delta t$. In other

words, this is the set of all allowed holonomic velocities $\boldsymbol{v_H}$ for which there exist control inputs that guarantee a tracking error below a certain fixed threshold ϵ_H as shown in Figure 2.10.

The optimal linear velocity input v^* that allows the cart to track $\boldsymbol{v_H}$ within a certain maximum tracking error ϵ_H is given by

$$v^* = \frac{v_H t \sin\theta_H \omega}{2(1-\cos\theta_H)} = \frac{\theta_H \sin(\theta_H)}{2(1-\cos\theta_H)}.$$

If the optimal linear velocity is not feasible, the optimal controls are

$$\begin{aligned}
&\omega = \theta_H/T \leq \omega_{max} \quad and \quad v = v^* \leq v_{max}, \\
&\omega = \theta_H/T \leq \omega_{max} \quad and \quad v = v_{max}, \\
&\omega = \omega_{max} \quad and \quad v = 0,
\end{aligned}$$

where T is a fixed amount of time given to the robot to achieve the correct orientation, v_{max} and ω_{max} are the maximum linear and angular velocities, respectively.

2.3.8 Goal Velocity Obstacle

The formulation in [111] extended the original VO to multi-agent systems in which the agents have to reach a goal region G rather than a target point. The reason behind this idea lies in the fact that in some applications all the agents or subgroups may have to share the same target region. The problem is solved by adopting the VO/RVO concepts to identify all the collision-free velocities that lead to a prescribed region by introducing the *Goal Velocity Obstacle* (GVO). Consider an agent A, moving with current velocity v_A, that has to enter in the goal region G; the Goal Velocity Obstacle for A induced by G is

$$GVO_{AG} \triangleq \{v \,|\, \exists\, s \in [0,t],\, s(v - v_A) \in G \oplus -A\},$$

as shown in Figure 2.11.

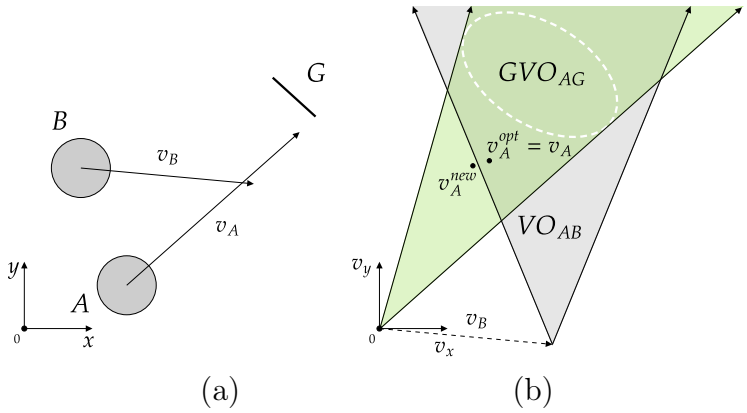
GVO_{AG} is the cone of all velocities that allow the agent A to enter in its goal region $G \subset \mathbb{R}^2$. At every cycle of sensing-and-acting, the agent A must select a new velocity such that

$$v_A^{new} = \underset{v \in GVO_{A|G} \setminus VO_A^\tau}{\arg\min} ||v - v_A^{opt}||_2,$$

where v_A^{opt} is the *optimization velocity*, that is defined as follows. If the current velocity v_A is pointing to G then the agent must choose $v_A^{opt} = v_A$, otherwise it means that $v_A \notin GVO_{AG}$ and thus the agent must choose

$$v_A^{opt} = \underset{v \in GVO_{A|G}}{\arg\min} ||v - v_A||_2.$$

When the number of autonomous agents in the dynamic environment is very large, it may occur that $GVO_{AG} \subset VO_A^\tau$, that means there are no collision free velocities leading to the goal region. This problem can be solved by



(a) (b)

Figure 2.11: Geometric representation of the Goal Velocity Obstacle. In (a) agents A, B and goal G (black segment). In (b) the Goal Velocity Obstacle GVO_{AG} of A induced by G (green cone). The white dashed region is G represented in the velocity space. v_A^{new} is selected in order to be simultaneously outside VO_{AB} and inside GVO_{AG}.

relaxing the constraints on the velocity space: by prioritizing the pure collision avoidance velocities forgetting the goal region for a while or not computing the collision cones with respect to the most distant agents. The approach takes into account moving regions, multiple regions, with/without time windows.

2.3.9 Optimal Velocity Selection for Velocity Obstacle

Kim and Oh [71] proposed an optimal velocity selection method for VO, named *Optimal Velocity selection for Velocity Obstacle* (OVVO). The motivation is that in very crowded scenarios, the original VO method is not able to select a safe velocity for each robot due to the lack of feasible candidates among which select the new safe velocity. The authors then proposed an optimization method based on minimizing a cost function for the desired velocity, allowing the robot to prioritize avoidance maneuvers over stopping policy and viceversa.

Consider an autonomous robot A and a dynamic obstacle O, moving with current velocities v_A and v_O, respectively. The cost function to be minimized is the sum of two components. The first component is

$$f_1 = k_{vd}||v_A^{pref} - v_A||_2,$$

where k_{vd} is a positive constant and v_A^{pref} is the robot preferred velocity. The second component is a function of the *clearance* (see Figure 2.12) and the *pass-time*. The clearance is the distance between a relative velocity and a relative distance

$$d_v = \frac{v_{OA} \times (x_A - x_O)}{||v_{OA}||_2},$$

where $v_{OA} = v_O - v_A$ is the relative velocity. A small d_v means that the obstacle is dangerous to the robot, thus v_{OA} is not safe. The function depending on the clearance is defined as follows

$$f_{dv} = k_1 d_v^{-c_1}$$

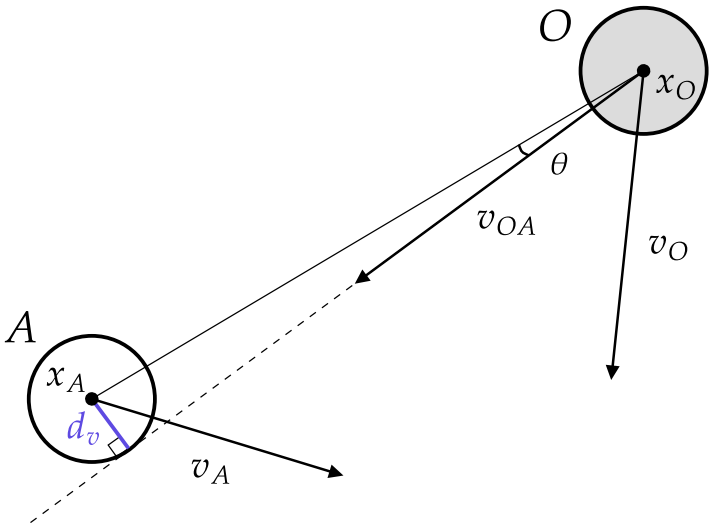


Figure 2.12: Geometrical interpretation of the clearance d_v (blue segment).

where k_1 and c_1 are positive constants.

The pass-time is the time needed by the robot to pass an obstacle,

$$t_p = \frac{(x_O - x_A)}{(v_O - v_A)}.$$

If t_p is negative, then the robot has already passed the obstacle, if t_p is large, then the obstacle is not dangerous for the robot. The function depending on the pass-time is given by

$$f_{tp} = k_2 t_p^{-c_2},$$

where k_2 and c_2 are positive constants.

The second component f_2 of the cost function is the product between f_{dv} and f_{tp}

$$f_2 = k_{tp} d_c^{-c_1} t_p^{-c_2}.$$

The overall cost is given by

$$J(v) \triangleq f_1 + f_2 = k_{vd} ||v_A^{pref} - v_A||_2 + k_{tp} d_c^{-c_1} t_p^{-c_2}.$$

The ratio between the constants k_{vd} and k_{tp} is of primary importance, since a higher k_{tp} means that the robot will give more importance to collision-free maneuvers while a higher k_{vd} makes the robot keeping its actual velocity v_A close to the desired one v_A^{pref}, even in presence of nearby obstacles. In their experimental results, authors compared the OVVO results with those coming from FVO, [59].

Table 2.1 briefly summarizes the Velocity Obstacle extensions analyzed so far. *Algorithm* denotes the acronym of the solution. *Environment* explains what type of scenarios the algorithm is able to handle. *Agents model* stands for the type of agents the algorithms are designed for. *Parameters* refer to agent's position, velocity, radius and orientation: deterministic means that all

these measurements do not contain uncertainty and are perfectly known by each agent, probabilistic means that some parameters are modeled as random variables. *Description* provides a quick overview of the method.

Table 2.1: Summary table

Algorithm	Environment	Agents model	Parameters	Description
VO	Agent-Obs	Holonomic	Deterministic	The agent computes an absolute collision cone for every moving obstacle and selects a new velocity out of it, as close as possible to the preferred one.
RVO	Multi-agent	Holonomic	Deterministic	Extension of VO to multi-agent systems: all the moving objects are decision making entities able to compute Velocity Obstacles.
ORCA	Multi-agent	Holonomic	Deterministic	Extension of RVO: to guarantee optimality, smooth and safe trajectories, the new velocity is selected by solving a constrained linear optimization problem.
GVO	Agent-Obs	Car-like	Deterministic	Generalization of the VO to car-like robots, in which the absolute collision cones are represented on the space of controls that move the agent.
NLVO	Agent-Obs	Non-Holonomic	Deterministic	Extension of the VO: the agent computes a warped cone to avoid collisions with obstacles following a known non-linear trajectory.
AVO	Multi-agent	Holonomic Car-like	Deterministic	Extension of ORCA: every agent selects its new velocity respecting a maximum acceleration constraint.
HRVO	Multi-agent	Holonomic Cart-like	Probabilistic, Kalman filtered	Extension of RVO, designed to solve the *reciprocal dances* problem. Position and velocity of agents are modeled as Gaussian random variables in order to model sensors noise.
NH-ORCA	Multi-agent	Cart-like	Deterministic	Extension of ORCA cart-like robots. Differential driven agents track holonomic velocities within a certain bounded error.
Goal VO	Multi-agent	Holonomic	Deterministic	Extension of VO: every agent computes and additional cone, containing all the velocities that allows it to reach a possibly moving goal region.
OVVO	Agent-Obs	Holonomic	Deterministic	VO integrated with optimal collision-free velocity selection, prioritizing clearance over pass-time and viceversa.

2.4 Velocity Obstacle for Constrained Agents

Path planning for multi-agent system is a critical aspect in automated warehouses where an efficient delivery and storage are mandatory. Moreover mobile robot cooperation, instead of developing bigger vehicles, can effectively increase the system robustness, coordination and fault-tolerant capabilities [79]. A common problem is the delivery of cumbersome loads. Mobile robot cooperation deals with two different sub-problems: formation control and cooperative path planning. Formation control aims to satisfy constraints on the relative position of the agents in the fleet [87] without any high-level decision-making architecture [79]. To account the formation control, many methods

are available in the literature and can be classified into: leader-follower approaches [92, 128] which require a centralised communication strategy; virtual shapes approaches [76] which provide better performance in terms of formation maintenance; and behaviour-based approaches [10] which are more flexible but lacks of system stability analysis. Cooperative path planning aims to provide optimised trajectories for the vehicles of the formation to accomplish some given task knowing the mission starting and end points, the environmental constraints and the required formation shape [15, 55].

In this section, a hybrid navigation system which uses the *Voronoi Diagrams* (VD) as a cooperative global planner and the Velocity Obstacle (VO) approach as local collision avoidance mechanism, such as in [94], will be used as a starting point. We will extend the Velocity Obstacle approach in order to handle rigid body constraints between agents during the local navigation and collision avoidance. Our solution is not a properly formation control: we assume the payload to be rigidly coupled to the agents.

2.4.1 Problem Statement and Background

The problem of cooperatively transporting a long and possibly heavy payload can be formulated as follows. Two holonomic agents are located at the end points of the load (e.g, a long steel bar) forming a system of two particles under a rigidity constraint.

The goal is to carry the bar from an initial pose to a goal pose without colliding with other agents and/or environmental obstacles.

The trajectory planner for the constrained agents, described in details in Section 2.4.2, is based on two levels: a global and a local level. The global level is for the collision avoidance with fixed expected obstacles, e.g, the shelves of a warehouse, and it is based on Generalized Voronoi Diagrams while the local level is based on the ORCA algorithm described in Section 2.3.2.

Voronoi Diagrams

As defined in [9], given a finite and discrete set of points $S = \{s_1, s_2, ..., s_k\}$ of $E \subseteq \mathbb{R}^2$, the Voronoi Diagram (VD) of S is a particular decomposition of E that associates a region $reg(s)$ to every point $s \in S$, such that all the points into $reg(s)$ are closer to s than any other point of the set S. Formally, for a generic point $s \in S$ the related plane region is computed as

$$reg(s) = \bigcap_{q \in S \setminus \{s\}} \{x \in E : ||x - s||_2 \leq ||x - q||_2\},$$

where $|| \cdot ||_2$ is the Euclidean distance. VD are suitable for autonomous navigation because moving along the edges of a Voronoi Diagram implies that the robot is as far away as possible from the neighbouring obstacles [120]. This implies that VD maximises the clearance. Generalised Voronoi Diagrams (GVDs) is an extension that can be used for path planning. GVDs allow to compute the

regions for primitives like points, lines, polygons, curves or surfaces [129]. Figure 2.13 shows an example of a space partition over the warehouse planimetry (black rectangles) used in our experimental setup.

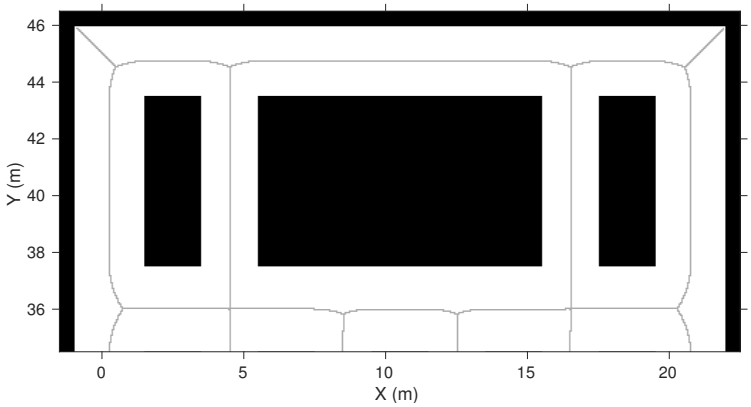


Figure 2.13: The grey lines shows an example of a $\mathbb{R}^2$ partition using Generalised Voronoi Diagram.

Rigid Body Constraint

Let (x_a, y_a) be the position of agent A and (x_b, y_b) be the position of B in the Euclidean space $E \subseteq \mathbb{R}^2$. They are bond by a rigid bar of length d which constraints their relative positions

$$(x_a - x_b)^2 + (y_a - y_b)^2 = d^2, \tag{2.9}$$

for every $t \geq 0$. The linked agents A and B form a system with three degrees of freedom: to know the exact configuration[1] of the system, it is sufficient to know the position of one of the agents and the relative orientation angle $\theta \in [0, 2\pi)$ with respect to a fixed reference frame $\Sigma_{x,y}$. The agents have to satisfy the following equations

$$\begin{cases} x_b &= x_a + d\cos\theta \\ y_b &= x_b + d\sin\theta \end{cases} \tag{2.10}$$

which is equivalent to (2.9). We can get the relation between the velocity of agents A and B as follows. Suppose that our constrained system AB lies in a certain initial configuration $r_a = (x_a, y_a)$ and $r_B = (x_b, y_b)$ at time t_0. The distance between the robots is d and θ is the orientation angle of the system accordingly to $\Sigma_{x,y}$. Suppose now that at time $t_0 + \Delta t$ the system reaches a new configuration in which r'_a and r'_b are the new positions of the robots and θ' is the new orientation angle. Since the distance between A and B is always ℓ because of the rigidity constraint, the following equations hold

$$\begin{aligned} \Delta r_a &= r'_a - r_a \\ \Delta r_b &= r'_b - r_b \\ \Delta\theta &= \theta' - \theta \end{aligned}$$

[1]The position in E of A and B.

and then

$$\Delta r_{ab} = \Delta r_a - \Delta r_b.$$

If $\Delta\theta$ is sufficiently close to zero, the following approximation $\ell \sin \Delta\theta \approx \ell\Delta\theta$ holds true. Thus, $\Delta r_{ab} \approx \ell\Delta\theta$ and the absolute value of the relative velocity can be derived by passing to the limit $\Delta t \to 0$

$$v_{ab} = \lim_{\Delta t \to 0} \frac{|\Delta r_{ab}|}{\Delta t} = \lim_{\Delta t \to 0} \ell \frac{\Delta\theta}{\Delta t} = \ell\omega$$

where $\omega = d\theta/dt$ is the angular velocity. The velocity vectors v_A and v_B then satisfy the following equation

$$v_a = v_b + v_{ab} = v_b + \ell \times \omega$$

or equivalently

$$v_b = v_a - v_{ab} = v_a - \ell \times \omega.$$

2.4.2 Trajectory Planner

As proposed in [94], we use Voronoi Diagrams as global planner as well. In our case we have to include some improvements to carry out the planning for linked agents. We start by planning the optimal trajectory for a single agent, obtaining the waypoints P^i for $i = 1, ..., m$. For every index i, the segment connecting the waypoints P^i and P^{i+1} is given by $L^i = P^{i+1} - P^i$. Then for every P^i like in [45], we define the waypoints for the constrained system as follows

$$P_A^i = P^i, \tag{2.11}$$

$$P_B^i = L^{k_{min}} \cap \mathcal{C}(P_A^i, \ell), \tag{2.12}$$

where $i = 1, ..., m$, $\mathcal{C}(P_A^i, \ell)$ is the circle of radius ℓ centred in P_A^i, and k_{min} is given by

$$k_{min} = \min\{k \in [i, ..., m) : L^k \cap \mathcal{C}(P_A^i, \ell) \neq \{0\}\}.$$

The index k_{min} identifies the index of the first segment L that has non-empty intersection with $\mathcal{C}(P_A^i, \ell)$. Equation (2.12) might result to be not satisfied when the payload is approaching its goal position, hence it might happen that, in a certain waypoint P_A^j, there is no k_{min} for which equation (2.12) is satisfied. This means that the global planner is not able to set the corresponding P_B^j. In that case we have to change our strategy in order to set a proper waypoint for agent B: if all the waypoints P_A^i set after P_A^j are such that

$$||P_A^j - P_A^i||_2 \leq \ell, \quad i = j + 1, ..., m,$$

i.e. entirely contained into the circle $\mathcal{C}(P_A^j, \ell)$, then the global planner sets the positions P_A^i and P_B^i by considering again equations (2.11) and (2.12) but switching the subscript indices A and B, so the circle will be centred in P_B^i, not in P_A^i. This allows us to stay on the optimal path planned in the beginning.

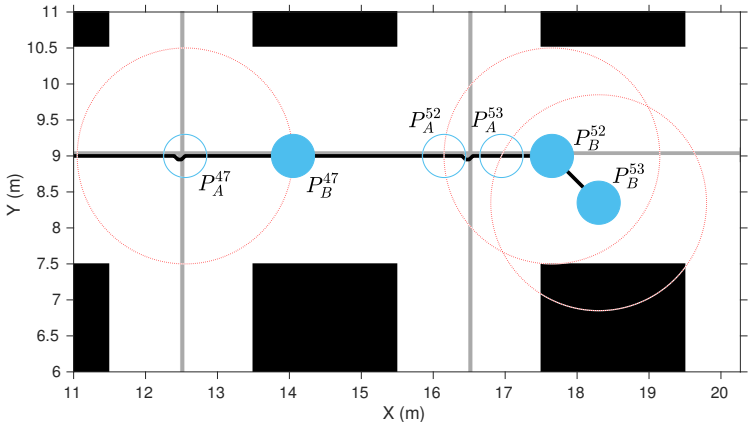


Figure 2.14: An example of planning for constrained agents A (empty blue circle) and B (filled blue circle) carrying a bar of 1.5m. The dotted red circles shows the position constraints between agents for each waypoint. The path (the thick black line) has been previously down sampled according to the path curvature.

Figure 2.14 shows an example of planning: waypoint P_B^{52} was P_A^{52} in origin, but it failed to satisfy equation (2.12) so it is necessary to rename it as P_B^{52} by swapping subscript indices and to use it as centre of $\mathcal{C}(P_B^{52}, \ell)$, in order to set P_A^{52}. Then move on to P_B^{53} that allows to position P_A^{53} and the goal is reached.

In order to make ORCA, seen in Section 2.3.2, able to manage collision avoidance for agents rigidly linked, we modified the constraints formulation in the linear programming section of the code for computing the half planes. First, we "group" the two robots A and B in order to constitute a single entity. Second, we run the "classic" ORCA algorithm to isolate the sets of all the collision avoidance velocities $\overline{ORCA}_A^\tau$ and $\overline{ORCA}_B^\tau$ with respect to any other agent or moving obstacle that does not belong to the constrained pair AB. We have

$$\overline{ORCA}_A^\tau = D(0, v_a^{max}) \cap \bigcap_{c \neq \{A,B\}} ORCA_{A|C}^\tau$$
$$\overline{ORCA}_B^\tau = D(0, v_b^{max}) \cap \bigcap_{c \neq \{A,B\}} ORCA_{B|C}^\tau$$

where A and B, do not collide each other by construction. Third, we update the new collision avoidance velocities v_a^{new} and v_b^{new} in such a way they satisfy

$$v_a^{new} \in \overline{ORCA}_A^\tau \cap RB$$
$$v_b^{new} \in \overline{ORCA}_B^\tau \cap RB$$

where RB in the set

$$RB = \{(v_a^{new}, v_b^{new}) : ||p_a^{new} - p_b^{new}||_2 - \ell = 0\}$$

with

$$p_a^{new} = p_a + v_a^{new} \Delta t$$
$$p_b^{new} = p_b + v_b^{new} \Delta t.$$

In other words, among the all possible collision avoidance velocities for A and B, i.e. those belonging to $\overline{ORCA}_A^\tau$ and $\overline{ORCA}_B^\tau$, robots A and B have to choose velocities v_A^{new} and v_B^{new} that not only guarantee no collisions within time τ, but that preserve the rigid link between them as well. Any other agent C sharing the same workspace avoids collisions accordingly to the original ORCA algorithm. Although under our assumption the physical link between robots A and B cannot be broken, the proposed approach is very important to avoid bad choices of v_A^{new} and v_B^{new}. If velocities that do not satisfy the rigid constraint are chosen, the robots will

- slip on the ground making the odometry measurements inaccurate,
- waste energy due to additional (and unnecessary) friction, and
- apply force/torque on the joints that could eventually damage the robots.

To obtain the velocities for A and B that satisfy the mechanical constraint and drive the robots toward their final targets, we adopt a non-linear programming solver based on the Augmented Lagrangian Method (ALM) [3]. ALM is a class of methods developed to solve (generally non-linear) constrained optimisation problems. Suppose we have to solve the following problem

$$\begin{aligned} &\arg\min_{x \in X} f(x) \\ &\text{subject to} \quad c_i(x) = 0 \quad \forall i \in I \end{aligned} \tag{2.13}$$

where I is a certain discrete set of indices and $f(\cdot)$, $c(\cdot)$ are possibly non-linear functions. Just like penalty methods, ALM solves (2.13) by iterating on a sequence of unconstrained problems

$$\min_{x \in X, \lambda_i \in \mathbb{R}} L_k(x, \lambda_i) = f(x) - \sum_{i \in I} \lambda_i c_i(x) + \frac{\mu_k}{2} \sum_{i \in I} c_i(x)^2$$

in which the Lagrangian multipliers λ_i, μ_k appear. The term $\frac{\mu_k}{2} \sum_{i \in I} c_i(x)^2$ is called "quadratic augmentation" and it makes the new objective function $L_k(x, \lambda)$ strongly convex as μ_k increases. After every iteration step k, the method updates not only μ_k (using the value at the previous step as initial guess), but also the Lagrangian multiplier in such a way to get $\mu_k c_i(x_k) \leftarrow \lambda_i$ where x_k is the solution of the unconstrained problem at the k-th step, i.e. $x_k = \arg\min L_k(x, \lambda_i)$. The accuracy of the Lagrangian multiplier λ_i increases at every step. The problem that we want to solve can be formalised as

$$\arg\min_{v_a^{new}, v_a^{new}} ||v_a^{new} - v_a^{pref}||_2 + ||v_b^{new} - v_b^{pref}||_2 \tag{2.14}$$

$$\text{subject to } ||p_a^{new} - p_b^{new}||_2 - d = 0 \tag{2.15}$$

$$f_i(v_a^{new}) \leq b_i, \quad i = 1, ..., N \tag{2.16}$$

$$f_j(v_b^{new}) \leq b_j, \quad j = 1, ..., M \tag{2.17}$$

where p_a^{new} and p_b^{new} are the optimal new positions, v_a^{new} and v_b^{new} the optimal new velocities of the constrained agents, v_a^{pref} and v_b^{pref} are the desired (unconstrained) velocities that allow the agents to reach their next waypoints, $d > 0$

is the length of the rigid link between A and B, $f_i(\cdot) \leq b_i$ and $f_j(\cdot) \leq b_j$ are the affine functions defining the $ORCA^{\tau}_{A|C}$ half-planes, $N = |\mathcal{C}_N \setminus \{B\}|$ and $M = |\mathcal{C}_M \setminus \{A\}|$ where $\mathcal{C}_N$ and $\mathcal{C}_M$ are the set of the agents detected through their sensing systems (e.g, laser scanners, cameras, LIDAR's etc) by A and B, respectively. The meaning of the objective function (2.14) is that we want to find collision avoidance velocities as close as possible to the preferred ones; equation (2.15) guarantees the rigid constraint between A and B; the meaning of equations (2.16) and (2.17) is that the new velocities have to belong to the half-planes generated by the current velocities of A and B, respectively.

2.4.3 Simulation Results

The proposed solutions has been tested in simulation using ROS Kinetic (Robot Operating System), RVO2 and Alglib 3.14.0 for solving the non-linear optimisation problem. The time-step used is $\Delta t = 50.0\,\text{ms}$, the ALM penalty coefficient and the number of outer iteration for the whole experiments are set to 10 and 500. During simulations the agents (abstracted as circles) have the same maximum speed $v_{max} = 0.5\,\text{m/s}$ and bounding circle radius of $0.3\,\text{cm}$. We tested out our solution in a simulated warehouse environment with corridor $2.5\,\text{m}$ wide and with a rigid body constraints of length $\ell = 1.5\,\text{m}$.

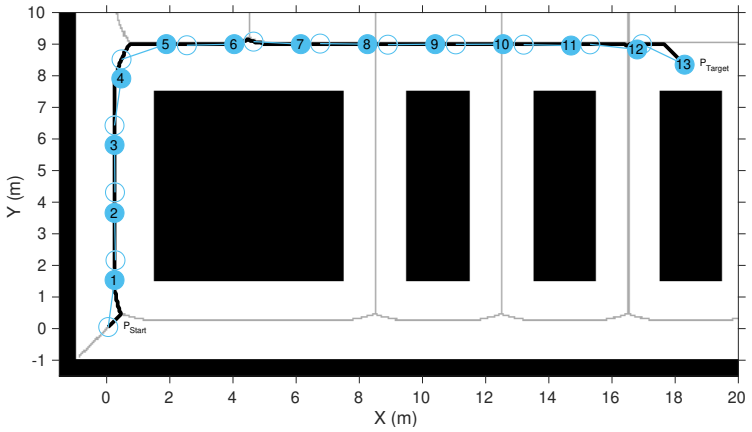


Figure 2.15: The Cartesian trajectory for the two coupled agents (agent A emptyblue circles, agent B filled blue circles). The thick black line shows the reference optimal path over the Voronoi diagram's edges (light gray lines). The numbers inside the filled blue circles refer to discrete time evolution.

Three different scenarios are implemented to validate the proposed approach:

1. **Constrained motion**: Two coupled agents A-B move along the precomputed path taking into account only the static obstacles defined by the planimetry of the warehouse;

2. **Constrained motion and obstacle**: Two coupled agents A-B move along the precomputed path where an obstacle is located;

3. **Constrained motion and agent**: Two coupled agents A-B move along the precomputed path and meet a third agent C moving along a trajectory which can cause a collision.

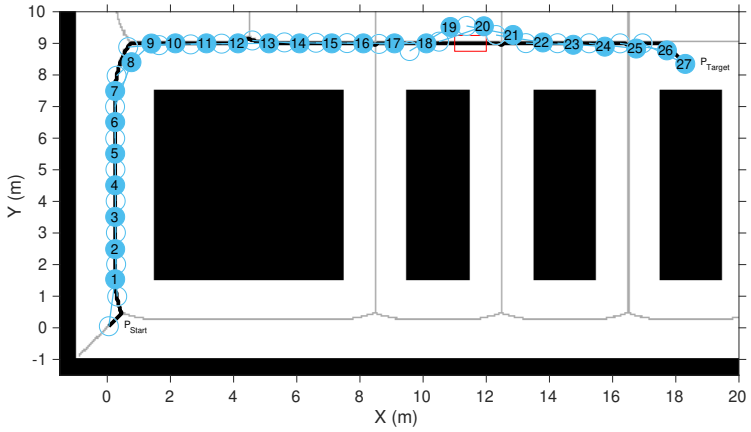


Figure 2.16: The Cartesian trajectory for the two coupled agents (agent A empty blue circles, agent B filled blue circles). The thick black line shows the reference optimal path over the Voronoi diagram's edges (light gray lines). The red rectangles is an obstacle over the precomputed trajectory.

Constrained motion

In the first scenario the agents move in a completely known and static environment. Two agents rigidly coupled with a bar are moving along a precomputed path as shown in Figure 2.15. The optimal path is the shortest part from P_{start} to P_{target} over the Voronoi edges which satisfies the rigid constraint. Since the optimisation algorithm explained in Section 2.4.2 is an iterative method, the solution may not be exact leading to a small error between the nominal positions of the robots and the actual ones. However, using a small time step Δt and increasing the iteration number it is possible to get feasible solutions in a reasonable time. In our case we computed collision free trajectories for A and B with a maximum error smaller than $\sim 4.4\,\text{mm}$.

Constrained motion and obstacle

In the second scenario we tested our solution with an obstacle placed over the precomputed paths for both agents A and B. We set a rectangle box of dimension $1.0\,\text{m} \times 0.5\,\text{m}$ and set the time horizon τ to $1.0\,\text{s}$. Figure 2.16 shows between discrete time 18 and 23 the avoiding manoeuvre. The coupled robots A-B move away from their nominal trajectories in order to avoid the obstacle. In this case they are in charge to completely carry out the avoidance. Once the manoeuvre is over, they get back to the nominal trajectory and continue the navigation towards the target position P_{target} according to the optimal paths. The maximum position error is $5.7\,\text{mm}$. It is larger than in the previous case because now the number of available solutions during the collision avoidance is smaller.

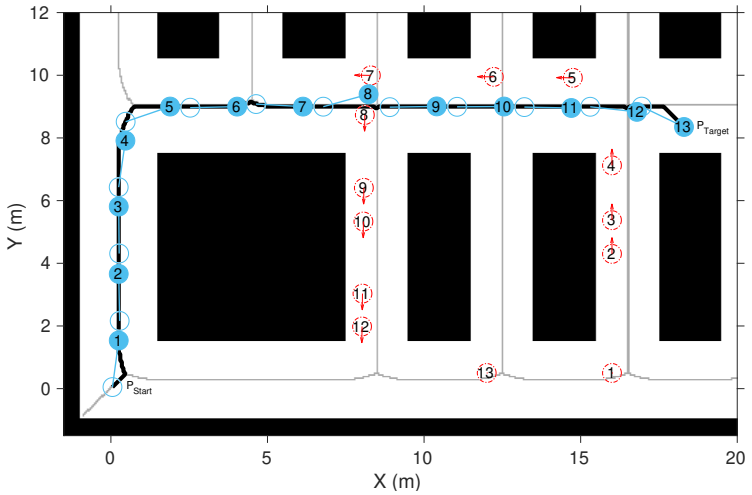


Figure 2.17: The Cartesian trajectory for two coupled agents (agent A empty blue circles, agent B filled blue circles), and an extra agent C which is free to move (red circle). The thick black line shows the reference optimal path over the Voronoi diagram's edges (light gray lines).

Constrained motion and agent

In the last scenario we have the coupled agents A and B moving over the pre-computed trajectory from P_{start} towards P_{target} following a set of waypoints and a third robot C is moving in a collision path. As shown in Figure 2.17 around discrete times 7-9 the trajectories of the agents cross, leading to a possible collision. As expected the agents A-B and C made successfully collision avoidance manoeuvres. The agent C treats the coupled agents as a single entity and senses the bar as an obstacle. This means that it cannot pass through the payload (constraint) during the collision avoidance manoeuvre. As in the previous scenarios the maximum position error is smaller than a centimetre (with respect to the bar length $\ell = 1.5\,\mathrm{m}$).

2.5 Velocity Obstacle for Planar Manipulators

As pointed out in Section 2.1, Velocity Obstacle is one of the most popular and studied decentralized trajectory planning methods for multi-agent systems moving in dynamic environments. It has been successfully used in a multitude of real and simulated scenarios for collision-free maneuvers of ground and aerial autonomous robots. Velocity Obstacles are exclusively used for the safe navigation of mobile robots, never for planning safe trajectories for robotic manipulators that have to avoid obstacles in their workspace. Artificial Potential Fields (APFs) [12, 70, 72] and Dynamic Movement Primitives (DMPs) [51, 52, 53, 113] are well known approaches for this scenario. In this section, we present a trajectory planner for planar two-link manipulators exploiting the VO algorithm and on its alternative formulation called Finite-Time Velocity Obstacle (FVO), both described in Section 2.2.

2.5.1 VO-based Planner

We consider a planar serial robot with 2 revolute joints (RR), consisting of a base link *Link* 1 of length ℓ_1 connecting the base J_1 to the elbow-joint J_2 and a distal link of length ℓ_2, *Link* 2, connecting J_2 to the end-effector EE.

In our formulation J_1, J_2 and EE are considered as *constrained* holonomic agents [95]. The planner we want to develop should be able to compute collision-free trajectories for *Link* 2 with respect to a moving obstacle, which does not block the motion of *Link* 1 in any instant of time.

Problem Statement

Let $P_i = (x_i, y_i)$, $i \in \{O, 1, 2, 3\}$, be the position of obstacle O, base J_1, elbow J_2 and end-effector EE, respectively. Similarly, let v_i and r_i denote their velocity vectors and radii. The manipulator has to move the end-effector from an initial configuration $P_3^0 = (x_3^0, y_3^0)$ (that corresponds to (θ_1^0, θ_2^0) in the joints space) to its goal position P_3^g, within a tolerance $d_\epsilon > 0$, i.e. $||P_g - P_3||_2 < d_\epsilon$, without colliding with an obstacle of radius r_O placed at $P_O = (x_O, y_O) \in \mathbb{R}^2$, see Fig. 2.18.

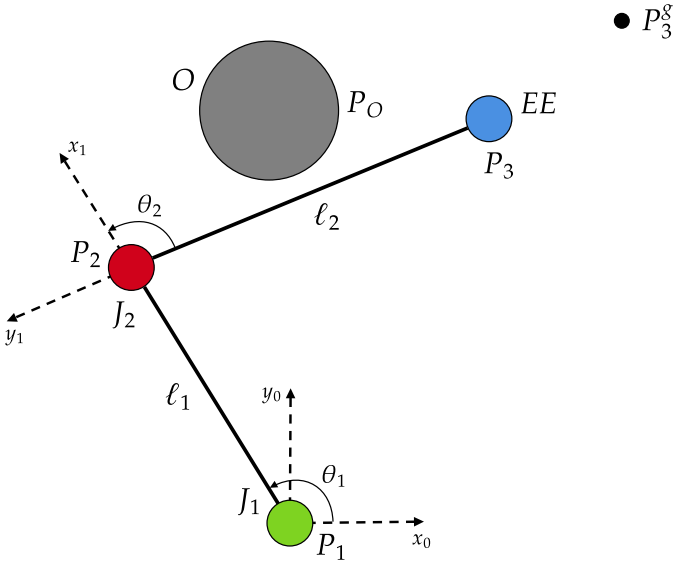


Figure 2.18: RR planar robot initial configuration, according to Denavit-Hartenberg convention. The base agent J_1 is coloured

Planner description

The base J_1 is fixed at the origin of the base frame $\Sigma_0 = \{x_0, y_0\}$, J_2 moves by remaining on the circumference of radius ℓ_1 centered in J_1 and at distance ℓ_2 from the end-effector. EE "drives" the motion of the manipulator calculating collision cones with respect to J_2 and O. EE acquires the goal position P_3^g in order to compute its preferred velocity vector v_3^{pref} given by

$$v_3^{pref} = v\frac{(P_3^g - P_3)}{||P_3^g - P_3||_2} = \left(\dot{x}_3^{pref}, \dot{y}_3^{pref}\right) \in \mathbb{R}^2, \tag{2.18}$$

where $v \in [0, v_3^{max}]$, and constructs the set of admissible velocities V as a disc centered in EE of radius $||v_3^{pref}||_2$. The EE acquires P_1 to reduce the set V to RV, the *Reduced Admissible Velocities set* for EE. P_1 placed on the right side of v_3^{pref} means that EE must not compute an avoiding maneuver on the left side of the obstacle, because this would surely cause a collision with the obstacle. RV is constructed by eliminating all the admissible velocities of V whose tip is on the left of v_3^{pref}, see Fig. 2.19. When J_1 is on the left side of v_3^{pref}, RV must be constructed by eliminating all the velocities on the right of v_3^{pref}.

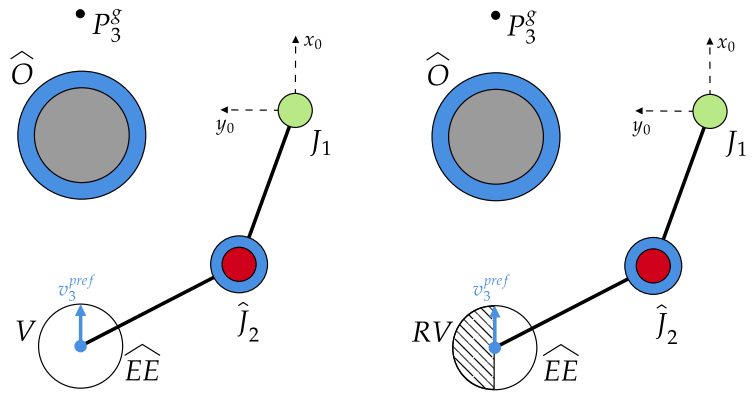


Figure 2.19: Configuration viewed from end-effector perspective. (a) EE defines v_3^{pref} via equation (2.18) and constructs the set V of admissible velocities, (b) RV is obtained from V by eliminating all those velocities on the left of preferred velocity, since J_1 is on the right of EE.

At each sensing-and-acting cycle EE acquires P_2 and P_O to construct the multiple velocity obstacle, given by

$$MVO_{EE} = CC_{EE\,J_2} \cup CC_{EE\,O}.$$

The end-effector EE updates its current position P_3^{curr} with P_3^{new} by selecting $v_3^{new} = (\dot{x}_3^{pref}, \dot{y}_3^{pref}) \in \mathbb{R}^2$ that satisfies the following constrained minimization problem

$$v_3^{new} = \underset{v \in RMVO_{EE}}{\arg\min} \; ||v_3^{pref} - v||_2 \tag{2.19}$$

$$\text{subject to } P_{Link2}^{new} \cap O = \emptyset, \tag{2.20}$$

where $RMVO_{EE} \triangleq RV \setminus MVO_{EE}$ is the *Reduced Multiple Velocity Obstacle*. P_{Link2}^{new} is the *Link* 2 configuration once v_3^{new} is selected, i.e.

$$P_{Link2}^{new} = P_3^{new} - P_2^{new}. \tag{2.21}$$

The new safe position for EE is given by $P_3^{new} = P_3^{curr} + v_3^{new}\Delta t$, where Δt is the time-step, and P_2^{new} belongs to the circumference of radius ℓ_1 centered in J_1. Let $P_2^{new} = (x_2, y_2)$ be the Cartesian coordinates of the new position for the elbow-joint. P_2^{new} coordinates can be computed directly from $P_3^{new} = (x_3, y_3)$

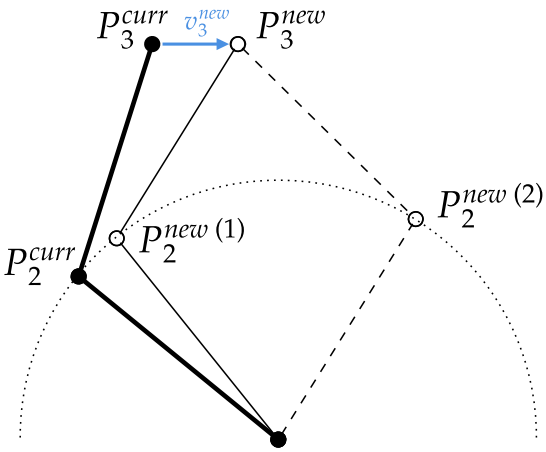


Figure 2.20: The two solutions for P_3^{new} once P_3^{new} is computed, for $\Delta > 0$. The thick continuous black line is the current configuration, the thin black line is the future feasible configuration and the dashed black line is the discarded configuration.

by solving the system

$$\begin{cases} x^2 + y^2 + ax + by + c &= 0 \\ x^2 + y^2 - \ell_1^2 &= 0 \end{cases} \tag{2.22}$$

where $a = -2x_3$, $b = -2y_3$ and $c = x_3^2 + y_3^2 - \ell_2^2$. The closed form solutions to system (2.22) are given by

$$x_2^{(1,2)} = \frac{-a(\ell_1^2 + c) \pm \sqrt{\Delta}}{(a^2 + b^2)},$$
$$y_2^{(1,2)} = -\frac{1}{b}\left(c + \ell_1^2 + \frac{a(a(\ell_1^2 + c) \pm \sqrt{\Delta})}{(a^2 + b^2)}\right),$$

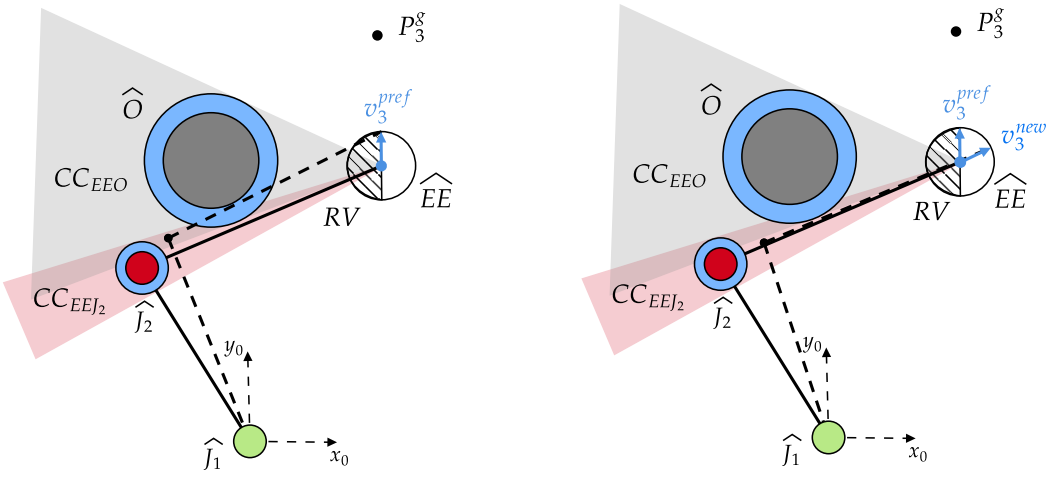
where $\Delta = a^2(\ell_1^2+c)^2-(a^2+b^2)(2\ell_1^2c+\ell_1^4+c^2-b^2\ell_1^2)$. $\Delta > 0$ implies that (2.22) has two distinct solutions (x_1^1, y_1^1) and (x_2^2, y_2^2), representing the two possible manipulator configurations corresponding to a new end-effector position for which the Jacobian is non-singular. Referring to Figure 2.20, the manipulator moves from the current configuration (thick continuous black line) to its new safe configuration (thin black line): the configuration given by $P_2^{new(2)}$ (gray dashed line) is not feasible for the continuity of motion, thus is discarded. $\Delta = 0$ implies that (2.22) has one solution (x, y) given by

$$x = -\frac{-a(\ell_1^2 + c)}{a^2 + b^2}, \quad y = -\frac{1}{b}\left(c + \ell_1^2 - \frac{a^2(\ell_1^2 + c)}{a^2 + b^2}\right)$$

since the manipulator is in a singularity configuration, i.e. the arm is stretched with $\theta_2 = 0$. $\Delta < 0$ corresponds to unfeasible configurations.

The planner satisfies constraint (2.20) as follows:

- (a) for every $v \in RMVO_{EE}$ it computes $P_3^{new} = P_3^{curr} + v\Delta t$,



(a) (b)

Figure 2.21: Geometric representation in the velocity space of how the planner works. Picture (a) shows the collision between the obstacle and *Link* 2 (black dashed line intersecting $\hat{O}$) that occurs if EE solves only equation (2.19) and therefore selects v_3^{pref} as new safe velocity vector. Picture (b) shows the collision-free maneuver occurring when EE selects v_3^{new} as safe velocity vector, by satisfying both equations (2.19) and (2.20).

- (b) it finds all the possible safe positions for the elbow-joint J_2 by solving system (2.22),
- (c) it calculates all the possible relative positions for *Link* 2 given by equation (2.21),
- (d) it discards all those intersecting the obstacle i.e. for which equation (2.20) is not satisfied and
- (e) finally, among the remaining, it selects the closest velocity v_3^{new} to v_3^{pref}.

Figure 2.21a shows what would happen if EE performs a collision avoidance maneuver by solving only equation (2.19). After constructing the RV set excluding all the velocities on the left of v_3^{pref} (striped half-circle), EE would keep the preferred velocity as safe velocity vector, since $v_3^{pref} \in RMVO_{EE}$. This leads to a collision between the obstacle and *Link* 2 (dashed black line intersecting $\hat{O}$). In Figure 2.21b EE satisfied both equations (2.19) and (2.20), selecting a velocity vector $v_3^{new} \in RMVO_{EE}$ that avoids collision between *Link* 2 and O. Once v_3^{new} is found, EE updates its position via $P_3^{new} = P_3^{curr} + v_3^{new}\Delta t$. The new velocity vector v_3^{new} for J_2 is calculated via

$$v_2^{new} = \frac{P_2^{new} - P_2^{curr}}{\Delta t}, \tag{2.23}$$

where P_2^{new} is obtained by solving system (2.22). Algorithm 1 shows the pseudo-code that implements the planner based on VO method for a planar 2R robot arm.

Algorithm 1 : **VO 2R Planner**

Set manipulator-obstacle-goal initial configuration;
Set $d \leftarrow ||P_3 - P_3^g||_2$;
while $d > d_\epsilon$ **do**
  acquire P_O, v_O and P_3^g;
  set v_3^{pref} for EE;
  create admissible velocities set V;
  adjust V creating RV according to P_1;
  compute $RMVO_{EE}$
  for all $v \in RMVO_{EE}$ **do**
    compute $P_3^{new} \leftarrow P_3 + v\,\Delta t$;
    compute P_2^{new} by solving system (2.22);
    compute $P_{link2}^{new} \leftarrow P_3^{new} - P_2^{new}$;
    find v closest to v_3^{pref} such that $P_{Link\,2}^{new} \cap O = \emptyset$;
  end for
  set $v_3^{new} \leftarrow v$;
  update $P_3^{new} \leftarrow P_3^{curr} + v_3^{new}\,\Delta t$;
  calculate v_2^{new} via equation (2.23);
  update $P_2^{new} \leftarrow P_2^{curr} + v_2^{new}\,\Delta t$;
  update $d \leftarrow ||P_3^{new} - P_3^g||_2$;
end while
Set $v_3^{pref} = 0$. % *target reached*

2.5.2 Simulation Results

We compare the results obtained with a planner based on the original VO paradigm, namely VO-planner, with those obtained by adopting a planner based on FVO paradigm, for different values of the cut-off factor. Joint velocities $\dot{q} = [\dot{\theta}_1, \dot{\theta}_2]^T$ are computed from Cartesian velocities of the end-effector $v = [\dot{x}, \dot{y}]^T$ via the following equation

$$\dot{q} = J^{-1}(q)v,$$

where $J^{-1}(q)$ is the inverse of the Jacobian matrix

$$J^{-1}(q) = \begin{bmatrix} \dfrac{\cos\theta_{12}}{\ell \sin\theta_2} & \dfrac{\sin\theta_{12}}{\ell \sin\theta_2} \\ \\ -\dfrac{\cos\theta_{12} + \cos\theta_1}{\ell \sin\theta_2} & -\dfrac{\sin\theta_{12} + \sin\theta_1}{\ell \sin\theta_2} \end{bmatrix},$$

with $\theta_{12} \triangleq \theta_1 + \theta_2$. Vector v is the safe velocity provided by the trajectory planner at every time-step $k\Delta t$, $k \in \mathbb{N}$.

Comparison VO-FVO

The manipulator is assumed to have links of the same length $\ell_1 = \ell_2 = 0.2\,\text{m}$. The initial configuration is $(\theta_1^0, \theta_2^0) = (135°, 35°)$, the goal is to move the end-effector in the position $P_3^g = (0.25, 0.27)\,\text{m}$, avoiding collisions with a fixed

obstacle placed in $P_O = (0.025, 0.260)\,\text{m}$ with radius $r_O = 0.05\,\text{m}$. (θ_1^0, θ_2^0), P_3^g and P_O have been chosen in such a way to induce a possible collision between *Link* 2 and the obstacle.

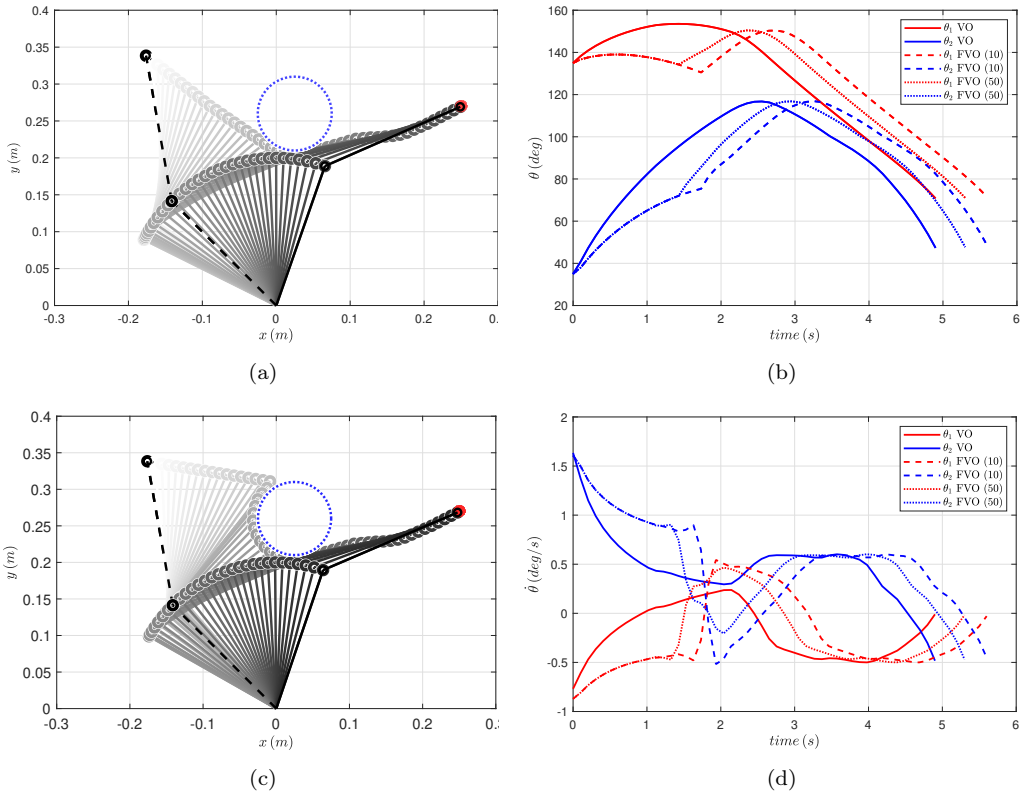


Figure 2.22: Manipulator motion time-lapses of the proposed scenario, obtained via VO planning in figure (a) and via FVO planning with $\tau = 10$ in figure (c). The corresponding joints angle and joints velocity time series are shown in figures (b) and (d). The dotted lines in both pictures (b) and (d) are obtained via FVO planning with a cut-off factor $\tau = 50$. Shaded colours are adopted to put in evidence the motion of the objects.

Figure 2.22a shows the time-lapse of the manipulator motion obtained using the VO-planner. The end-effector starts avoiding the obstacle from the very beginning, because VO planner is based on full collision cones and v_3^{pref} belongs to CC_{EEO} at the initial configuration. EE starts moving to the right and continues until is able to point at the goal position, i.e. until v_3^{pref} finally exits CC_{EEO}, and then moves directly towards it. Figure 2.22c shows the time-lapse of the same setup, obtained by adopting FVO-planner with $\tau = 10$. EE keeps the preferred velocity as long as possible, since v_3^{pref} does not belong to CC_{EEO} until the end-effector is very close to the obstacle. Then the manipulator performs an abrupt collision avoidance maneuver causing non-smooth position profiles, as it can be observed in Figure 2.22b at $t = 1.70\,\text{s}$ for $\tau = 10$ and at $t = 1.50\,\text{s}$ for $\tau = 50$. Figure 2.22d shows joint velocities, exhibiting similar non-smooth profiles. The VO-planner provides the smoothest trajectories and it is the most efficient in terms of time, leading the end-effector to the goal position in $T_{VO} = 4.90\,\text{s}$ against $T_{FVO}(10) = 5.50\,\text{s}$ and $T_{FVO}(50) = 5.20\,\text{s}$. It is worth highlighting that the adoption of FVO is suggested only when the

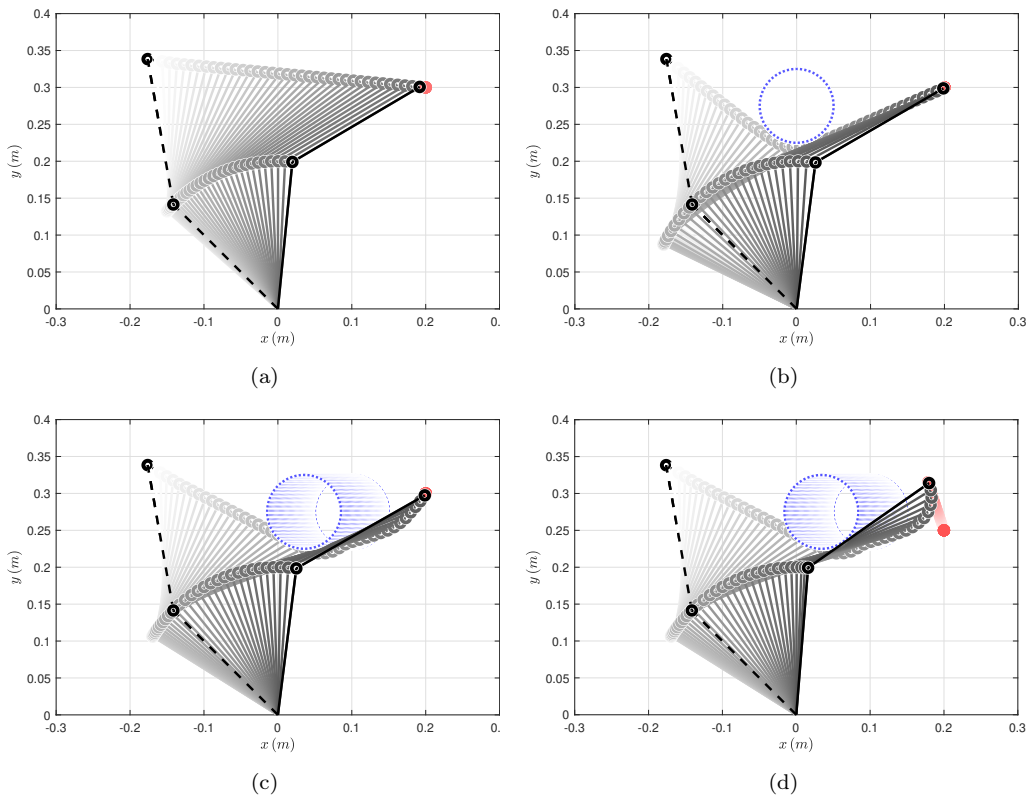


Figure 2.23: Four additional simulation setups. (a) A simulation without obstacle and with fixed target, (b) fixed obstacle and fixed target, (c) a moving obstacle and fixed target, and (d) moving obstacle and a moving target. Shaded colours are adopted to put in evidence the motion of the objects.

manipulator is moving in a crowded environment with multiple obstacles at different distances.

Setups with Moving Obstacle and Target

We examine four additional simulation setups in which collision avoidance maneuvers are performed by adopting the VO-planner and, where obstacle and target move with constant speed. The initial configuration for the manipulator is $(\theta_1^0, \theta_2^0) = (135°, 35°)$. Figure 2.23a shows a setup without obstacles and with a fixed target for the end-effector placed in $P_3^g = (0.20, 0.30)\,\text{m}$, while Figure 2.23b refers to the same situation but including a fixed obstacle placed in $P_O = (0.000, 0.275)\,\text{m}$. In Figure 2.23c the obstacle starts moving from the initial position $P_O = (0.100, 0.275)\,\text{m}$ and moves forward with constant velocity vector given by $v_O = (-0.015, 0.000)$ m/s. Figure 2.23d illustrates the same situation in which the target also moves, from the initial position $P_3^g = (0.200, 0.250)\,\text{m}$ along a straight line with a constant velocity vector given by $v_3^g = (-0.005, 0.015)\,\text{m/s}$. We highlight that, unlike case (b) in which the end-effector can proceed directly to the goal as soon as it passes the fixed obstacle, in cases (c) and (d) the planner moves EE a bit on the right before reaching the goal, in order to allow a safe navigation of *Link* 2.

Chapter 3

Dynamics Modeling for Delta Robots

3.1 Related Works

Parallel robots are spatial mechanical structures that consist of kinematic closed chains. They usually have two platforms: one is fixed (the *base*), the other one can have arbitrary motions in its workspace (the *moving platform*). Three mobile legs structures, assembled as serial robots with two rods linked by passive spherical joints, connect the effector attached to the moving platform to the base. Actuated revolute joints located at the base and prismatic joints at the links allow the moving platform to translate and rotate in the three dimensional space.

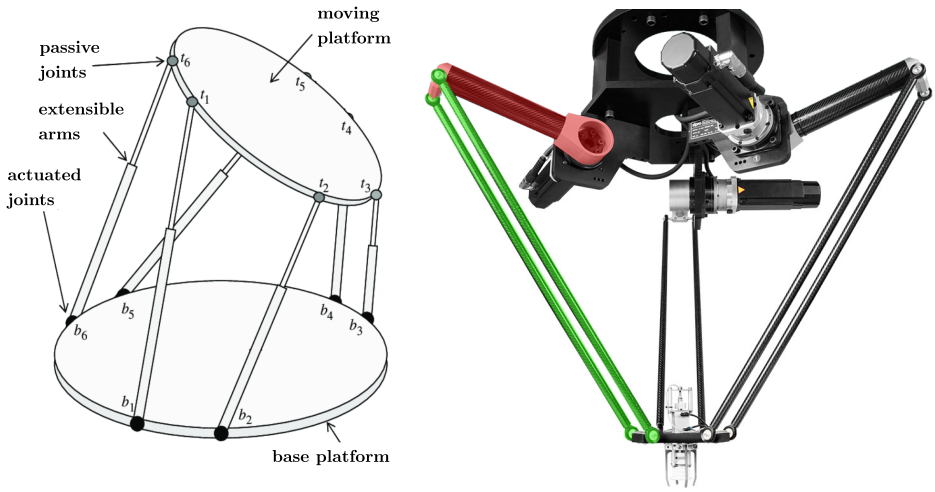
Parallel robots are particularly suitable for a wide range of industrial applications, such as pick-and-place, packaging, assembly etc., and over the decades have aroused particular interest in the robotic community. Parallel robots possess several advantages when compared to serial manipulators, offering a generally higher rigidity, smaller mobile mass and low manufacturing costs. These characteristic translate into a particularly favorable ratio between the maximum payload and the weight of the robot itself, fast movements with high repeatability and precise manipulations.

Research in the field of parallel robots dates back to the late '30s, when Pollard [99] invented and patented a mechanical structure for car painting. A few years later, Stewart used a parallel structure, see Figure 3.1a, with a 6-DoFs moving platform invented by Mc Cough in his flight simulator [117]. Recent interesting publications about modeling and model-based control of complex parallel manipulators with many degrees of freedom are [132, 133, 134].

The Delta Robot is a kind of parallel manipulator invented by Reymond

This chapter is based on the following publications:
▷ F.Falezza, F.Vesentini, A. Di Flumeri, L. Leopardi, G. Fiori, G. Mistrorigo, R. Muradore. *Gray-Box Model Identification and Payload Estimation for Delta Robots*, IFAC-PapersOnLine (Elsevier), 53(2), pp. 8771-8776.
▷ F.Falezza, F.Vesentini, A. Di Flumeri, L. Leopardi, G. Fiori, G. Mistrorigo, R. Muradore. *A Novel Inverse Dynamic Model for 3-Dof Delta Robots*, Mechatronics (Elsevier) 2022 (accepted for publication).

Clavel and his research team [30] in 1987. The robot was originally designed with a mechanical parallel structure with three links, forming a closed kinematic chain connecting a fixed platform to the end-effector. The tool mounted at the end-effector is capable of translating along the Cartesian X, Y and Z axes. Several modifications to the original Delta Robot model have been presented with the intent to increase the degrees of freedom of the robot. For example, a model with four actuated links allowed to rotate the end-effector [97]; or the Hexa Delta robot, whose six kinematic chains arranged by pairs allowed to reach 6-DoF giving the possibility to control the orientation of the end-effector besides its Cartesian position [96]. It is common to refer to the original model with the name 3-DoF Delta Robot, see Figure 3.1b.



(a) (b)

Figure 3.1: A schematic representation of the 6-DoFs moving platform by Mc Cough (a), and a 3-DoF Delta Robot (b). The first link is highlighted in red, and the second link in green. Courtesy: SIPRO Srl `https://www.sipro.vr.it/`.

The origin of such name is the presence of three serial kinematic chains with 3 degrees of freedom, which allow the end-effector to move in the three-dimensional workspace. Each kinematic chain consists of two rigid links connected by 2-DoF revolute passive joint. The first link of each chain is actuated by an electric motor, attached to the fixed upper base.

A reliable dynamic model able to provide real-time estimation of the input torques is important for the design of torque control laws, payload estimation tasks and fault detection algorithm. The dynamic modelling of parallel Delta Robots is, unfortunately, a quite complex task due to their complicated kinematic structure subjected to holonomic constraints, that closes the kinematic chains to the moving platform and the presence of passive joints whose positions do not belong to the set of generalised coordinates. Contrary to other types of robots with passive angles, e.g., the *parallelogram arm* [110], in the case of Delta robots it is not possible to express the passive joint angles as a functions of the actuated joint angles only. For these reasons, a significant simplification of each kinematic chain is common in order to derive a model suitable for the implementation in modern controllers and, in general,

for model-based control laws. In the literature there are several simplified dynamic models for Delta robots, obtained by exploiting different methods.

Codourey [31, 32] simplified the kinematic structure of 3-DoF Delta Robots by neglecting the inertia of the passive links (also referred as *forearms*) and concentrating one third of their mass on the moving plate and the other two thirds on the elbow. The dynamic model is then calculated by exploiting the Principle of Virtual Work. Furthermore, Codourey and Burdet [33] proposed a method for deriving the dynamic model for 3-DoF Delta robot in linear form. Based on the same assumptions, Miller [84] derived and experimentally validated the inverse dynamics of a direct drive DELTA-580 robot[1] by direct application of the Hamilton's Principle. Fumagalli and Masarati [48] used a general-purpose multibody software in order to algoritmically compute the inverse dynamics of a 3-DoF Delta Robot for real-time control applications. Recently, also Asadi and Heydari [7] proposed an inverse dynamic model in closed form based on the same type of assumptions.

Alternatively, Tsai [122, 123] simplified the kinematic chains of the Delta Robot by assuming that the mass of the passive link of each kinematic chain is evenly divided between the end-effector and the end of the actuated links. On this assumption, Staicu [115] proposed recursive algorithm for solving the inverse dynamics problem by combining the Principle of Virtual Work and the *the fundamental equations of the parallel robots dynamics* [114]. Other noteworthy contributions about dynamics modelling, control and payload estimation applications are Park et al. [91], Angel et al. [5], Castañeda et al. [24], Falezza et al. [38] and Farsoni et al. [41]. Recently, Kuo [74] exploited the model in the analysis of a Delta Robot with flexible links and by Kuo and Huang [75] in model-based control applications.

A simplified inverse dynamic model has the advantage of having a mathematical formulation in closed form relatively easy to handle, that can be readily used to design closed-loop controls laws and it is suitable for the hardware of modern microcontrollers. The major disadvantage is that simplifications on the kinematic chains lead to a significant loss in the total inertia and potential energy of each kinematic chain. The calculated control torques then result to be underestimated with respect to the real ones.

In order to remedy this issue, one has essentially two ways: adopting a parameter identification technique to estimate a few parameters that reduce the inaccuracy or searching for a complete model without simplifications of the kinematic structure.

Angel and Viola [4] improved the dynamic model by applying a parameter identification technique based on a recursive least square (RLS) algorithm. Falezza et al. [38] multiplied the inertia matrix and the compensation of gravity term by two unknown scalars and then performed a Gray-Box parameters identification in order to find their optimal values, i.e., those that minimize the error between the real and estimated torques.

Bortoff [20] recently formulated a complete dynamic model for a 3-DoF Delta Robot exploiting both the Lagrange's and Hamilton's methodologies,

[1] courtesy of Laboratory of Microengineering, Swiss Federal Institute of Technology.

resulting in a set of 24 first-order Differential Algebraic Equations (DAEs) of Index 3. The excessive index makes the model almost impossible to be solved and implemented in high-frequency control architectures. Suitability needs to be restored after the adoption of the index reduction method. To the best of our knowledge, there are no complete dynamic models for delta robots in the literature.

3.2 Delta Robot D3-1200 Kinematic Structure

A 3-DoF Delta Robot is composed of a moving platform $\boldsymbol{e}_1 - \boldsymbol{e}_2 - \boldsymbol{e}_3$ connected to a fixed base $\boldsymbol{f}_1 - \boldsymbol{f}_2 - \boldsymbol{f}_3$ through three sets of kinematic chains. Each chain consists of two rigid links $\overline{\boldsymbol{f}_i\boldsymbol{j}_i}$ and $\overline{\boldsymbol{j}_i\boldsymbol{e}_i}$, with $i \in \{1, 2, 3\}$, connected by a two-degree passive revolute joint in $\boldsymbol{j}_i$. Links in the kinematic chains are actuated by three electric motors located at points $\boldsymbol{f}_i$ of the fixed frame. The base reference frame $\Sigma_b = \{O;\, X,\, Y,\, Z\}$ is centered in the circumcenter O of rigid platform. Figure 3.2 shows the CAD model of the robot under study.

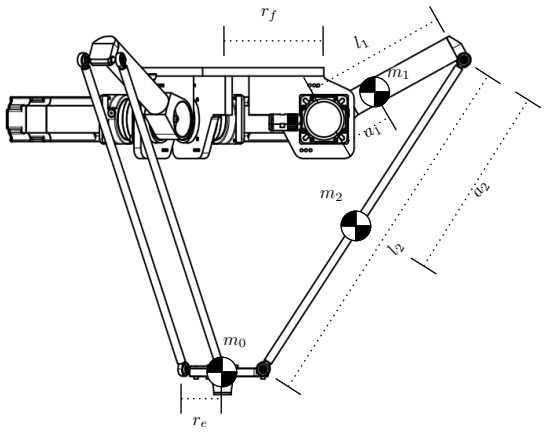


Figure 3.2: CAD model of the robot under study, reporting the position of the CoMs of links and end-effector.

The center of the moving platform is the end-effector, and its position, expressed with respect to the base frame Σ_b, is

$$\boldsymbol{x} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}. \tag{3.1}$$

The links $\overline{\boldsymbol{f}_i\boldsymbol{j}_i}$ and $\overline{\boldsymbol{j}_i\boldsymbol{e}_i}$ have two 2-DoF spherical joints on $\boldsymbol{j}_i$ and $\boldsymbol{e}_i$. The proximal and distal links of each kinematic chain have identical length, *i.e.* $l_1 = ||\overline{\boldsymbol{f}_i\boldsymbol{j}_i}||$ is the length of the actuated link of mass m_1, $l_2 = ||\overline{\boldsymbol{j}_i\boldsymbol{e}_i}||$ is the length of the passive links with $i \in \{1, 2, 3\}$. The passive links are composed by two parallel rods to increase mechanical rigidity. However, for deriving the mathematical model is sufficient to consider just a single rod with total mass m_2 (equal to the sum of the masses of the two parallel rods). Let $r_f = ||\overline{O\boldsymbol{f}_i}||$ be the radius of the circle passing from $\boldsymbol{f}_1 - \boldsymbol{f}_2 - \boldsymbol{f}_3$, and $r_e = ||\overline{\boldsymbol{x}\boldsymbol{e}_i}||$ be

the radius of the circle passing from $\boldsymbol{e}_1 - \boldsymbol{e}_2 - \boldsymbol{e}_3$, where $||\cdot||$ denotes the Euclidean norm. Angles θ_i with $i \in \{1, 2, 3\}$ represent the joint orientation of the actuated links and their values are provided by the motor encoders. The angles γ_i and ψ_i represent the orientation of the passive links and are not measured.

3.2.1 Direct Kinematics

The kinematic structure has to satisfy the following holonomic constraint of rigidity

$$||\boldsymbol{x} - \boldsymbol{j}_i'||^2 = l_2^2, \tag{3.2}$$

where

$$\boldsymbol{j}_i' = \begin{bmatrix} (r + l_1 \cos\theta_i)\cos\alpha_i \\ (r + l_1 \cos\theta_i)\sin\alpha_i \\ -l_1 \sin\theta_i \end{bmatrix}$$

with $r = r_e - r_f$ and α_i is the rotation of $\boldsymbol{f}_i$ with respect to O in the $\{X, Y\}$ plane. Equation (3.2) is the intersection of three spheres with centres $\boldsymbol{j}_i$ and radii l_2, a quadratic equation of the Cartesian coordinates that has two real solutions, one belonging to the real half-space

$$\mathcal{H}_+ \triangleq \{(x, y, z) \in \mathbb{R}^3 : z > 0\}$$

and the other belonging to

$$\mathcal{H}_- \triangleq \{(x, y, z) \in \mathbb{R}^3 : z < 0\}.$$

According to the reference frame we adopted, the end-effector is in $\mathcal{H}_-$, therefore the solution belonging to $\mathcal{H}_+$ is discarded [74, 75, 122].

3.2.2 Inverse Kinematics

The inverse kinematics $\kappa^{-1} : \mathbb{R}^3 \to \mathbb{R}^3$ computes the values of the generalized coordinates θ_1, θ_2 and θ_3 given the Cartesian coordinates (3.1) of the end-effector, expressed with respect to the robot fixed frame Σ_b. According to [74], the active joint angles can be obtained by solving equations (3.2) for $i \in \{1, 2, 3\}$. They can be written as $\ell_i \cos\theta_i + p_i \sin\theta_i = n_i$, where

$$\begin{aligned} \ell_i &= 2rl_1 - 2l_1 x \cos\alpha_i - 2l_1 y \sin\alpha_i, \\ p_i &= 2l_1 z, \\ n_i &= 2rx - 2ry \sin\alpha_i + x^2 + y^2 - z^2 + l_1^2 - l_2^2 + r^2. \end{aligned}$$

There are four possible solutions, refer to [122] for further details on how to select the right one.

Table 3.1: Kinematic and dynamic parameters of D3-1200 delta robot.

Parameter	Value	Unit
r_f	0.250	m
r_e	0.100	m
l_1	0.375	m
l_2	0.900	m
a_1	0.122	m
a_2	0.450	m
m_0	0.940	kg
m_1	1.400	kg
m_2	0.390	kg

Table 3.2: Electromechanical parameters of D3-1200 delta robot.

Parameter	Value
Weight	64 Kg
Power supply	380 V three phase
Motors	Kollmorgen AKM44G (3 units)
Installed power	4.5 KW
Class Protection	IP55
Temperature	$0 \div 45$°C
Humidity	95%
Repeatability	0.1 mm
Max. n° of cycles	150 per minute
DoF	3
Max. working range	1200 mm
Max. payload	4 Kg

3.2.3 Dynamic Parameters

The value of the dynamic parameters for the Delta robot under study are listed in Table 3.1. Table 3.2 reports the electromechanical parameters of the robot.

In particular, r_f is the radius of the fixed upper-base, r_e is the radius of the moving platform, l_1 and l_2 are the lengths of active and passive link of each kinematic chain, a_1 represents the distance between the active joint rotation axis in $\boldsymbol{f}_i$ and the center of mass of the active links, a_2 is the distance between each passive revolute joint in $\boldsymbol{j}_i$ and the center of mass of passive links; m_i for $i \in \{0, 1, 2\}$ are the mass of the moving platform, the active link and the passive link, respectively.

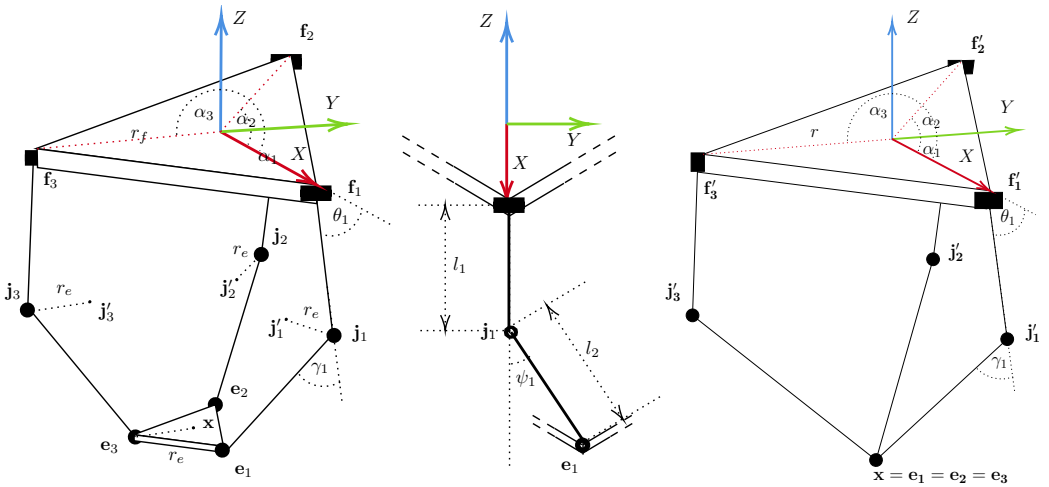
3.3 Gray-Box Model Identification and Payload Estimation

The inverse dynamic model is derived from a simplification of the kinematic structure; for this reason we refer to it as *simplified dynamic model*, denoted

with $\mathcal{M}_s$. The parameter identification procedure leads to derive the identified model, $\mathcal{M}_{Id}$, that improves the estimation of the torques and estimates the payload.

3.3.1 The Simplified Dynamic Model

In order to treat the end-effector as a point mass concentrated in $\boldsymbol{x}$, we assume to "horizontally shrink" the entire structure of the robot by r_e, i.e., we consider the geometric representation in Figure 3.3b which is equivalent to Figure 3.3a from the dynamics point of view. The radius of the fixed base becomes $r = r_f - r_e$, the motor positions are $\boldsymbol{f}'_i$, the passive joint positions are $\boldsymbol{j}'_i$ and the vertices $\boldsymbol{e}_i$ of the moving platform collapse to $\boldsymbol{x}$; angles θ_i, γ_i, ψ_i and α_i are preserved as well as the length of the links.



(a) (b)

Figure 3.3: Geometric representation of the kinematic structure (a) and the same structure horizontally shrinked by r_e. Angles $\alpha_i \in \{1, 2, 3\}$ are the rotation angles around the Z-axis of the reference frame.

According to [74, 75, 91, 116, 122], to derive $\mathcal{M}_s$ of the robot, we need the following approximations on the kinematic chains shown in Figure 3.3b:

A1 : the active links are considered as homogeneous rods of mass m_1, length l_1 and center of mass at $l_1/2$,

A2 : the passive links are modeled as two points of mass $m_2/2$ concentrated in $\boldsymbol{j}'_i$ and in $\boldsymbol{x}$.

These simplifications allow to avoid the inclusion of the passive joint angles γ_i and ψ_i into the generalized coordinate vector $\boldsymbol{q}$, leading to a system of differential equations easier to handle. The dynamics is obtained by solving the *constrained Euler-Lagrange equations*

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{\boldsymbol{q}}}\right) - \frac{\partial \mathcal{L}}{\partial \boldsymbol{q}} = \tau_g + \lambda \frac{\partial \mathcal{F}(\boldsymbol{q})}{\partial \boldsymbol{q}} \tag{3.3}$$

where

$$\boldsymbol{\lambda} = [\lambda_1\, \lambda_2\, \lambda_3]^T$$

are the Lagrange multipliers, $\boldsymbol{\tau}_g$ is the generalized force vector acting on the system, $\mathcal{F}(\boldsymbol{q})$ is the constraint (3.2) and

$$\boldsymbol{q} = (x\, y\, z\, \theta_1\, \theta_2\, \theta_3)^T \quad \dot{\boldsymbol{q}} = \left(\dot{x}\, \dot{y}\, \dot{z}\, \dot{\theta}_1\, \dot{\theta}_2\, \dot{\theta}_3\right)^T, \tag{3.4}$$

are the generalized coordinates and velocities [22]. The kinetic energy $\mathcal{T}$ of the simplified model $\mathcal{M}_s$, is given by the sum of the kinetic energies of the end-effector $\mathcal{T}_0$, of the actuated links $\mathcal{T}_1$ and of the passive links $\mathcal{T}_2$. Due to the assumptions A1 and A2, the inertia terms of the actuated and passive links can be written as

$$\mathcal{I}_1 = \frac{1}{3} m_1 l_1^2, \quad \mathcal{I}_2 = \frac{1}{2} m_2 l_1^2,$$

and since the end-effector $\boldsymbol{e}_1 - \boldsymbol{e}_2 - \boldsymbol{e}_3$ is modeled as a point of mass m_0 in $\boldsymbol{x}$, the kinetic energy contributions [74, 75, 122] are

$$\begin{aligned}
\mathcal{T}_0 &= \frac{1}{2} m_0 ||\dot{\boldsymbol{x}}||^2, \\
\mathcal{T}_1 &= \frac{1}{2} \mathcal{I}_1 \sum_{i=1}^{3} \dot{\theta}_i^2 = \frac{1}{6} m_1 l_1^2 \sum_{i=1}^{3} \dot{\theta}_i^2, \\
\mathcal{T}_2 &= \frac{1}{2} \sum_{i=1}^{3} \left(\frac{1}{2} m_2 ||\dot{\boldsymbol{x}}||^2 + \mathcal{I}_2 \dot{\theta}_i^2 \right) = \frac{1}{4} m_2 \sum_{i=1}^{3} \left(||\dot{\boldsymbol{x}}||^2 + l_1^2 \dot{\theta}_i^2 \right).
\end{aligned}$$

While the potential energy contributions [74, 75, 122] are

$$\begin{aligned}
\mathcal{V}_0 &= m_0 g z, \\
\mathcal{V}_1 &= \frac{1}{2} m_1 g\, l_1 \sum_{i=1}^{3} \sin\theta_i, \\
\mathcal{V}_2 &= \frac{1}{2} m_2 g \sum_{i=1}^{3} (z + l_1 \sin\theta_i),
\end{aligned}$$

where g is the gravity acceleration. The computation of the derivatives in

equation (3.3) results to

$$\begin{aligned}
\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{x}}\right) &= \left(m_0 + \frac{3}{2}m_2\right)\ddot{x}, \\
\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{y}}\right) &= \left(m_0 + \frac{3}{2}m_2\right)\ddot{y}, \\
\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{z}}\right) &= \left(m_0 + \frac{3}{2}m_2\right)\ddot{z}, \\
\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{\theta}_i}\right) &= \left(\frac{1}{3}m_1 + m_2\right) l_1^2 \ddot{\theta}_i, \qquad i \in \{1,2,3\} \\
\frac{\partial \mathcal{L}}{\partial x} &= 0, \\
\frac{\partial \mathcal{L}}{\partial y} &= 0, \\
\frac{\partial \mathcal{L}}{\partial z} &= -\left(m_0 + \frac{3}{2}m_2\right) g, \\
\frac{\partial \mathcal{L}}{\partial \theta_i} &= -\frac{1}{2}\left(m_1 + m_2\right) g l_1 \cos\theta_i, \qquad i \in \{1,2,3\}.
\end{aligned}$$

The derivatives with respect to the Cartesian variables, together with the right-hand side of Equation (3.3), lead to the following system of second-order equations

$$\boldsymbol{M}\left(\ddot{\boldsymbol{x}} + \boldsymbol{g}\right) - 2\boldsymbol{A}(\boldsymbol{q})\boldsymbol{\lambda} = \boldsymbol{h}_{\text{ext}}, \tag{3.5}$$

with

$$\boldsymbol{M} = \begin{bmatrix} m_0 + \frac{3}{2}m_2 & 0 & 0 \\ 0 & m_0 + \frac{3}{2}m_2 & 0 \\ 0 & 0 & m_0 + \frac{3}{2}m_2 \end{bmatrix},$$

$$\ddot{\boldsymbol{x}} = \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix}, \boldsymbol{g} = \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix}, \boldsymbol{A}(\boldsymbol{q}) = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix},$$

where

$$\begin{aligned}
a_{1i} &= x + r\cos\alpha_i - l_1\cos\theta_i\cos\alpha_i, \\
a_{2i} &= y + r\sin\alpha_i - l_1\cos\theta_i\sin\alpha_i, \qquad i \in \{1,2,3\} \\
a_{3i} &= z - l_1\sin\theta_i.
\end{aligned}$$

The vector $\boldsymbol{h}_{\text{ext}} \in \mathbb{R}^3$ represents the external forces applied to the end-effector. The derivatives with respect to the joint variables yield to

$$\boldsymbol{I}\ddot{\boldsymbol{\theta}} + \boldsymbol{G}(\boldsymbol{q}) - 2\boldsymbol{K}(\boldsymbol{q})\boldsymbol{\lambda} = \boldsymbol{\tau}, \tag{3.6}$$

where $\boldsymbol{\tau}$ is the command torque vector and

$$\boldsymbol{I} = l_1^2 \begin{bmatrix} \frac{1}{3}m_1 + \frac{1}{2}m_2 & 0 & 0 \\ 0 & \frac{1}{3}m_1 + \frac{1}{2}m_2 & 0 \\ 0 & 0 & \frac{1}{3}m_1 + \frac{1}{2}m_2 \end{bmatrix}, \ddot{\boldsymbol{\theta}} = \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \\ \ddot{\theta}_3 \end{bmatrix}, \boldsymbol{G}(\boldsymbol{q}) = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix},$$

$$\boldsymbol{K}(\boldsymbol{q}) = \begin{bmatrix} k_{11} & 0 & 0 \\ 0 & k_{22} & 0 \\ 0 & 0 & k_{33} \end{bmatrix}, \boldsymbol{\tau} = \begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \end{bmatrix}.$$

with

$$v_i = \frac{1}{2}(m_1 + m_2) g\, l_1 \cos\theta_i,$$
$$k_{ii} = (x\cos\alpha_i + y\sin\alpha_i + r)\sin\theta_i - z\cos\theta_i$$

for $i \in \{1, 2, 3\}$. The control torques $\boldsymbol{\tau}$ are calculated by solving the system (3.5) with respect to $\boldsymbol{\lambda}$, given the external forces contribution $\boldsymbol{h}_{\text{ext}}$, and then substituting $\boldsymbol{\lambda}$ into equation (3.6)[2]. In the following we assume $\boldsymbol{h}_{\text{ext}} = [0,\, 0,\, 0]^T$.

3.3.2 Gray-Box Model Identification and Friction Estimation

$\mathcal{M}_s$ is an approximation of the real robot, as pointed out by assumptions A1 and A2 in Section 3.3.1. In particular, the fact that the mass of the passive link is divided evenly between the end-effector and the end of the active link yields to a noticeable underestimation of the inertia effect. Moreover, the contribution of joint friction is not taken into account at all. These two issues lead to an under-estimation of the real torque computed by the industrial controller of the real robot.

We propose a method to compensate for the missing inertia and potential energy, and the viscous and Coulomb frictions. In other words, by starting from $\mathcal{M}_s$, we construct the identified model $\mathcal{M}_{Id}$ and the identified model with friction estimation $\mathcal{M}_{Id+F}$. Figure 3.4 shows the control loop and the identification block.

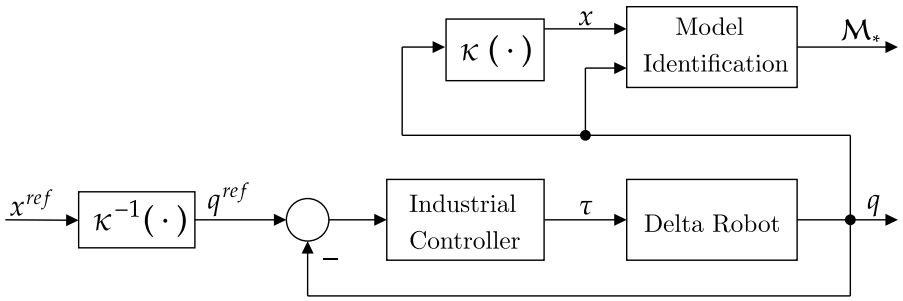


Figure 3.4: Control loop with the identification block. $\mathcal{M}_*$ represents the inverse dynamic model that one wants to identify, it can be either $\mathcal{M}_{Id}$ or $\mathcal{M}_{Id+F}$.

We construct $\mathcal{M}_{Id}$ as follows. Two scalar parameters ρ_1 and ρ_2 are added to $\mathcal{M}_s$ in order to take into account the missing inertia and potential energy.

[2]We point out that in [122], contrary to us, they assume m_2 to be the mass of just one of the two parallel rods constituting the passive links. However, the structure of the equations of motion remains unchanged: the only difference is that m_2 is divided by a factor 2.

Equation (3.6) is accordingly adapted to

$$\rho_1 \boldsymbol{I}\ddot{\boldsymbol{\theta}} + \rho_2 \boldsymbol{G}(\boldsymbol{q}) - 2\boldsymbol{K}(\boldsymbol{q})\boldsymbol{\lambda} = \boldsymbol{\tau}. \tag{3.7}$$

In order to fit the best values for ρ_1 and ρ_2, an offline approach has been adopted. The method consists in minimizing the error ϵ_τ between the real robot torques $\boldsymbol{\tau}$ and the estimated torques $\hat{\boldsymbol{\tau}}$, over a set of trajectories Υ.

The error minimization is performed with a greedy approach over a range of values between $(\rho_1^{min}, \rho_1^{max})$ and $(\rho_2^{min}, \rho_2^{max})$. In particular, the algorithm cycle through every trajectory $\upsilon \in \Upsilon$, computing the torques $\boldsymbol{\tau_\upsilon}$ for every $(\rho_1, \rho_2)_\upsilon \in (\rho_1^{min}, \rho_1^{max}) \times (\rho_2^{min}, \rho_2^{max})$, and finds the pair $(\rho_1^\star, \rho_2^\star)_\upsilon$ which minimizes

$$(\rho_1^\star, \rho_2^\star)_\upsilon = \underset{(\rho_1,\rho_2)_\upsilon}{\arg\min}\, \epsilon_{\tau_\upsilon} = RMS(\boldsymbol{\tau_\upsilon} - \hat{\boldsymbol{\tau}}_{\boldsymbol{\upsilon}}). \tag{3.8}$$

The overall optimal values $\rho_1^\star$ and $\rho_2^\star$ to be included in (3.7) are given by the average

$$(\rho_1^\star, \rho_2^\star) = \frac{\sum_\upsilon (\rho_1^\star, \rho_2^\star)_\upsilon}{|\Upsilon|}, \tag{3.9}$$

where $|\Upsilon|$ is the cardinality of Υ. Substituting $\rho_1^\star$ and $\rho_2^\star$ of equation (3.9) into (3.7) in place of ρ_1 and ρ_2 yields to the identified model $\mathcal{M}_{Id}$.

Friction plays an important role in determining the torques when the robot operates at high speed, as in the case of Delta Robots. Coulomb and viscous friction models, as reported in [69], account for the most of the contributions of friction at joints. We consider $\mathcal{M}_{Id}$ and we include the friction contribution into the model by writing

$$\rho_1^\star \boldsymbol{I}\ddot{\boldsymbol{\theta}} + \rho_2^\star \boldsymbol{G}(\boldsymbol{q}) - 2\boldsymbol{K}\boldsymbol{\lambda} + \boldsymbol{B}\dot{\boldsymbol{\theta}} + \boldsymbol{S}\,\mathrm{sign}(\dot{\boldsymbol{\theta}}) = \boldsymbol{\tau}, \tag{3.10}$$

where $\boldsymbol{B}$ and $\boldsymbol{S}$ are two 3×3 diagonal matrices given by

$$\boldsymbol{B} = \begin{bmatrix} f_{\upsilon_1} & 0 & 0 \\ 0 & f_{\upsilon_2} & 0 \\ 0 & 0 & f_{\upsilon_3} \end{bmatrix}, \quad \boldsymbol{S} = \begin{bmatrix} f_{c_1} & 0 & 0 \\ 0 & f_{c_2} & 0 \\ 0 & 0 & f_{c_3} \end{bmatrix}.$$

Matrices $\boldsymbol{B}$ and $\boldsymbol{S}$ contain the viscous and Coulomb friction coefficients, $f_{\upsilon_i} > 0$ and $f_{c_i} > 0$, respectively. The coefficients are estimated via Least Squares method [8]. For every $\upsilon \in \Upsilon$, let $|\upsilon|$ be the number of samples in υ. Let $\boldsymbol{\tau_{\upsilon,i}} \in \mathbb{R}^{|\upsilon|\times 1}$ and $\hat{\boldsymbol{\tau}}_{\boldsymbol{\upsilon,i}} \in \mathbb{R}^{|\upsilon|\times 1}$ be the times series of measured torques and computed torques, for joint i, at every sample time, respectively. Let $\dot{\boldsymbol{\theta}}_{\boldsymbol{\upsilon,i}} \in \mathbb{R}^{|\upsilon|\times 1}$ be the vector of i-th joint angular velocity, from now on, the joint index i will be omitted. By defining

$$\Delta\boldsymbol{\tau_\upsilon} = \boldsymbol{\tau_\upsilon} - \hat{\boldsymbol{\tau}}_{\boldsymbol{\upsilon}} \quad \Phi_\upsilon = \left[\dot{\boldsymbol{\theta}}_{\boldsymbol{\upsilon}} \;\; \mathrm{sign}(\dot{\boldsymbol{\theta}}_{\boldsymbol{\upsilon}})\right],$$

we have the following linear system

$$\Delta\boldsymbol{\tau_\upsilon} = \Phi_\upsilon \begin{bmatrix} f_c \\ f_v \end{bmatrix}.$$

The viscous and Coulomb friction coefficients are estimated by

$$\begin{bmatrix} f_c^\star \\ f_v^\star \end{bmatrix} = (\Phi_v^T \Phi_v)^{-1} \Phi_v^T \Delta\boldsymbol{\tau_v},$$

and, finally,

$$\begin{bmatrix} f_c^\star \\ f_v^\star \end{bmatrix} = \frac{\sum_v [f_c \, f_v]_v^T}{|\Upsilon|}$$

are the averages over all the trajectories. The model $\mathcal{M}_{Id+F}$ is obtained by plugging $f_v^\star$ and $f_c^\star$ into matrices $\boldsymbol{B}$ and $\boldsymbol{S}$, respectively, in order to be included in equation (3.10).

3.3.3 Payload Estimation

Delta Robots are largely used in the manufacturing industry for high-speed pick-and-place tasks. When an object is picked, the dynamic model needs to be modified accordingly to return the correct values of torques to be provided as feed-forward signal to the control architecture.

In industrial applications, robots may need to pick objects of unknown sizes and weights resulting in a loss of precision in the torque computation. The purpose of this section is to present a real-time method to identify the payload to increase the accuracy during these specific tasks. In what follows we assume that parameters ρ_1 and ρ_2 have already been identified and included in the inertia matrix $\boldsymbol{I}$ and gravity compensation term $\boldsymbol{G(q)}$.

Equation (3.5) (with $\boldsymbol{h}_{\text{ext}} = [0,\, 0,\, 0]^T$) is modified to explicitly consider the payload mass m_p, as

$$\boldsymbol{M}\,(\ddot{\boldsymbol{x}} + \boldsymbol{g}) + \boldsymbol{M_p}\,\ddot{\boldsymbol{x}} - 2\boldsymbol{A(q)}\boldsymbol{\lambda} = \boldsymbol{0}, \tag{3.11}$$

where $\boldsymbol{M_p}$ is a 3×3 symmetric matrix given by

$$\boldsymbol{M_p} = \begin{bmatrix} m_p & 0 & 0 \\ 0 & m_p & 0 \\ 0 & 0 & m_p \end{bmatrix}.$$

Substituting $\boldsymbol{\lambda} = \frac{1}{2}\boldsymbol{A^{-1}(q)}\,[\boldsymbol{M}\,(\ddot{\boldsymbol{x}} + \boldsymbol{g}) + \boldsymbol{M_p}\,\ddot{\boldsymbol{x}}]$ into equation (3.6) we get

$$\boldsymbol{I}\ddot{\boldsymbol{\theta}} + \boldsymbol{G(q)} - \boldsymbol{K(q)A^{-1}(q)}\,[\boldsymbol{M}\,(\ddot{\boldsymbol{x}} + \boldsymbol{g}) + \boldsymbol{M_p}\,\ddot{\boldsymbol{x}}] = \boldsymbol{\tau}.$$

Since $\Delta\boldsymbol{\tau} = \boldsymbol{\tau} - \hat{\boldsymbol{\tau}}$ and $\hat{\boldsymbol{\tau}} = \boldsymbol{I}\ddot{\boldsymbol{\theta}} + \boldsymbol{G(q)} - \boldsymbol{K(q)A^{-1}(q)M}\,(\ddot{\boldsymbol{x}} + \boldsymbol{g})$ are estimated torque with no payloads, we have the following equality

$$-\boldsymbol{A(q)K^{-1}(q)}\Delta\boldsymbol{\tau} = \boldsymbol{M_p}\ddot{\boldsymbol{x}}. \tag{3.12}$$

We define $\boldsymbol{y(k)} = -\boldsymbol{A(k)K^{-1}(k)}\Delta\boldsymbol{\tau(k)}$, where we set $\boldsymbol{A(k)} = \boldsymbol{A(q(k))}$ and $\boldsymbol{K(k)} = \boldsymbol{K(q(k))}$ to simplify the notation for the sake of clarity.

The Least Square problem to be solved to estimate $\boldsymbol{M_p}$ is given by

$$\min_{\widehat{\boldsymbol{M}}_{\boldsymbol{p}}} \sum_k ||\boldsymbol{y^T(k)} - \ddot{\boldsymbol{x}}\boldsymbol{(k)}\widehat{\boldsymbol{M}}_{\boldsymbol{p}}||^2,$$

where $t = kT_s$ is the current time with T_s the sample time of the controller. As shown in [8], the recursive method to solve a least-square problem is needed since observations are obtained sequentially at run time during the picking and releasing of objects. Therefore, the current estimation $\hat{\boldsymbol{M}}_{\boldsymbol{p}}(\boldsymbol{k})$ is given by the recursive equation

$$\hat{\boldsymbol{M}}_{\boldsymbol{p}}(\boldsymbol{k+1}) = \hat{\boldsymbol{M}}_{\boldsymbol{p}}(\boldsymbol{k}) - \boldsymbol{\Psi}(\boldsymbol{k}) \left(\boldsymbol{y}^{\boldsymbol{T}}(\boldsymbol{k}) - \ddot{\boldsymbol{x}}^{\boldsymbol{T}}(\boldsymbol{k}) \widehat{\boldsymbol{M}}_{\boldsymbol{p}}(\boldsymbol{k}) \right), \tag{3.13}$$

where

$$\boldsymbol{\Psi}(\boldsymbol{k}) = \boldsymbol{\mathcal{P}}(\boldsymbol{k}) \ddot{\boldsymbol{x}}(\boldsymbol{k+1}) \left[\sigma + \ddot{\boldsymbol{x}}^{\boldsymbol{T}}(\boldsymbol{k+1}) \boldsymbol{\mathcal{P}}(\boldsymbol{k}) \ddot{\boldsymbol{x}}(\boldsymbol{k+1}) \right]^{-1},$$
$$\boldsymbol{\mathcal{P}}(\boldsymbol{k}) = \frac{1}{\nu} \left[\boldsymbol{I} - \boldsymbol{\Psi}(\boldsymbol{k-1}) \ddot{\boldsymbol{x}}^{\boldsymbol{T}}(\boldsymbol{k}) \right] \boldsymbol{\mathcal{P}}(\boldsymbol{k-1}).$$

The value $\nu \in [0, 1]$ is the forgetting factor. With a fine-tuned ν, the model can adapt the torques accordingly with the payload very fast.

3.3.4 Experimental Results

We now show the experimental results proving the higher accuracy of the torques estimated by $\mathcal{M}_{Id}$ and by $\mathcal{M}_{Id+F}$ than to the ones estimated by the simplified model $\mathcal{M}_s$. The torque profile of each model has been compared to the real torque profile of a D3-1200 Delta robot manufactured by SIPRO Srl using an industrial controller, over a set of trajectories. A Net Analyzer, over Ether-CAT, allowed the recording of kinematic information, commanded and executed torques. Recorded torque values are already multiplied by the motor gear ratio and are the ground truth to verify the goodness of each model presented in this paper.

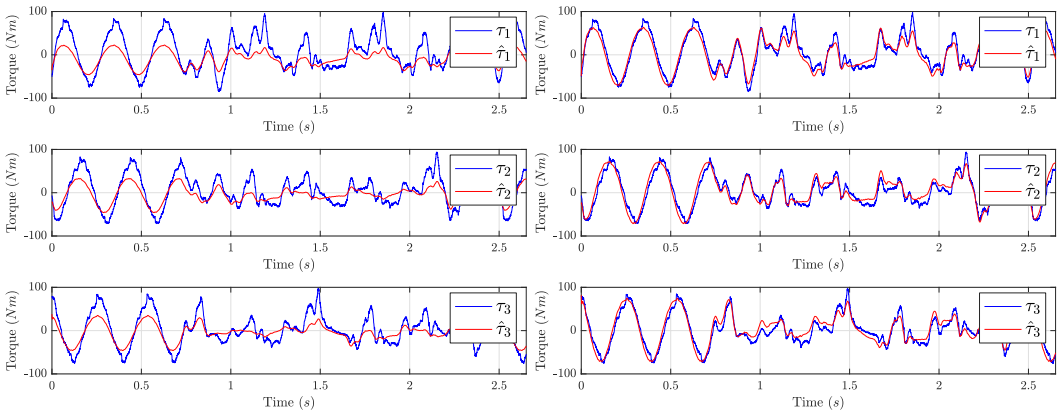
Table 3.3: Mean error and standard deviation of each model over υ_F

	$\mathcal{M}_{KH}$		$\mathcal{M}_{Id}$		$\mathcal{M}_{Id+F}$	
	μ	σ	μ	σ	μ	σ
τ_1 (Nm)	14.32	23.97	3.62	11.89	3.94	5.67
τ_2 (Nm)	7.98	20.30	-1.09	11.55	-1.11	5.79
τ_3 (Nm)	6.62	20.23	-2.10	11.91	-1.49	6.06

We proposed a free-motion trajectory without payload υ_F, meant to stress the robot, e.g., large accelerations and motion close to the workspace boundary. Figure 3.5a shows the torque profiles $\hat{\boldsymbol{\tau}}_{\boldsymbol{i}}$ and the real robot torques $\boldsymbol{\tau}_{\boldsymbol{i}}$. The dynamical model underestimates the real torques, due to the model simplifications A1 and A2 as explained in Section 3.3.1.

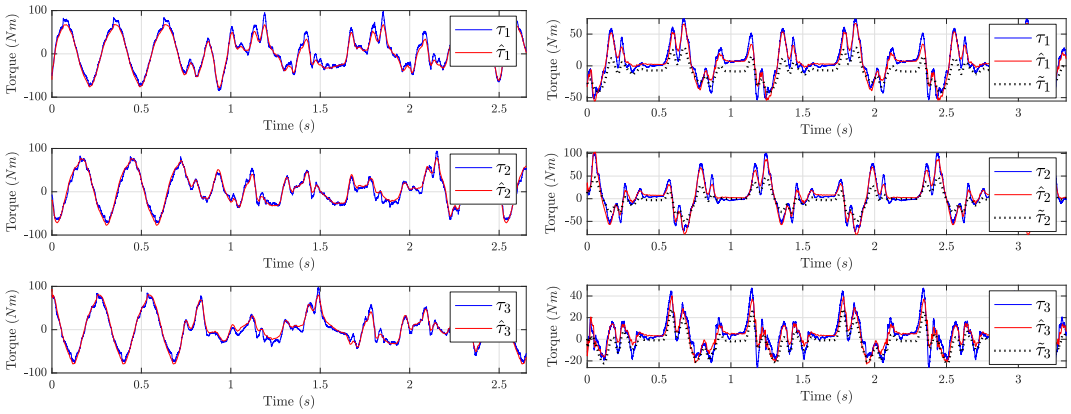
The gray-box dynamic parameter identification seen in Section 3.3.2, over a set of $|\Upsilon| = 12$ trajectories, allowes to identify the optimal parameters $\rho_1^\star$ and $\rho_2^\star$. The same trajectory υ_F is used to compare the model with the identified parameters, $\mathcal{M}_{Id}$ and $\mathcal{M}_s$.

Figure 3.5b shows the identified model torque time series $\hat{\boldsymbol{\tau}}_{\boldsymbol{i}}$ with respect to the torques $\boldsymbol{\tau}_{\boldsymbol{i}}$. The dynamic parameters $\rho_1^\star$ and $\rho_2^\star$ compensate for the



(a) (b)

Figure 3.5: Real torques $\boldsymbol{\tau}$ compared with estimated torques $\hat{\boldsymbol{\tau}}$ computed by the model $\mathcal{M}_s$ (a) and with estimated torques $\hat{\boldsymbol{\tau}}$ computed by the model $\mathcal{M}_{Id}$ (b), for each actuated joint using the trajectory υ_F.



(a) (b)

Figure 3.6: Real torques $\boldsymbol{\tau}$ compared with estimated torques $\hat{\boldsymbol{\tau}}$ computed by the model $\mathcal{M}_{Id+F}$, for each actuated joint over trajectory υ_F (a), and with and without payload identification, using the trajectory υ_P (b).

lack of inertia and potential energy in the previous model. The model follows more accurately the robot torque profile. The absence of friction correction is noticeable, especially when the robot is moving at low and high velocities. In those cases, the error between the real and computed torques increases. Figure 3.6b shows the torques of the dynamic model $\mathcal{M}_{Id+F}$ with the identified payload information $\hat{\boldsymbol{\tau}}_{\boldsymbol{i}}$, at run-time, compared to the dynamic model without this information, $\tilde{\boldsymbol{\tau}}_{\boldsymbol{i}}$, and the real robot torques $\boldsymbol{\tau}_{\boldsymbol{i}}$. The test has been executed over the trajectory υ_P. Figure 3.7b shows the reduction of the errors when the payload m_P is correctly computed. The implementation of the friction estimation method of Section 4.2 brings to the third model $\mathcal{M}_{Id+F}$. The matrices $\boldsymbol{B}$ and $\boldsymbol{S}$ increase the overall model accuracy. The dynamic model $\mathcal{M}_{Id+F}$ with the previously identified dynamic parameters and the friction coefficients is compared with $\mathcal{M}_s$ over trajectory υ_F. Figure 3.6a shows the dynamic model enriched with friction computation $\hat{\boldsymbol{\tau}}_{\boldsymbol{i}}$, with respect to the real

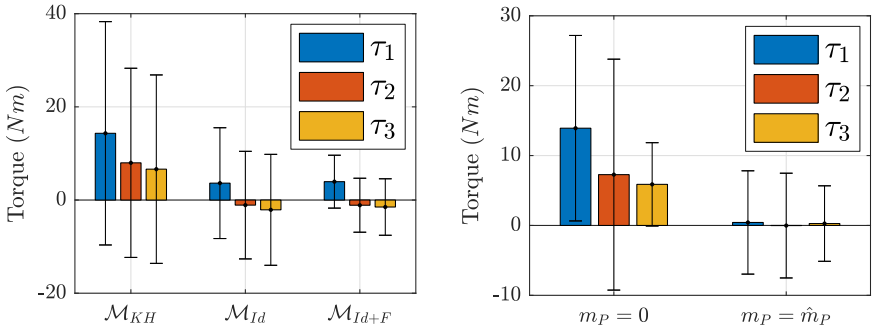


Figure 3.7: Mean errors with standard deviations. In (a) each label represents the errors between the dynamical model $\mathcal{M}_s$ and the real robot torque profile. In (b) each label represents the errors between the dynamical model $\mathcal{M}_{Id+F}$ and the real robot torque profile with $m_P = 0$ and $m_P = \hat{m}_P$.

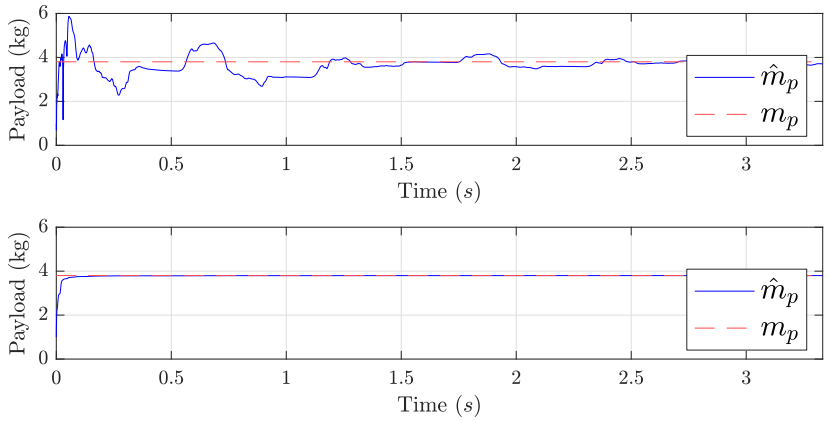


Figure 3.8: Payload estimation over time. The mass $\hat{m}_p$ is the estimated payload, while m_p is the correct payload attached to the end-effector. The top graph shows the estimation using the real torque measurements whereas the bottom graph shows the estimation using the simulated torque of the $\mathcal{M}_{Id+F}$ model.

robot torque $\boldsymbol{\tau_i}$. The estimated torques by the mathematical model $\mathcal{M}_{Id+F}$ follow the real torques much better, both at high and low velocities.

The mean μ and the variance σ of the error $e(k) = \tau(k) - \hat{\tau}(k)$ have been drastically reduced as shown in Table 3.3. Figure 3.7a displays the values of Table 3.3. Until now, each model has been tested in free-motion with no payload. υ_P is a free-motion trajectory, with a payload of 3.8 kg at the end-effector. The trajectory contains movements from a standard pick-and-place motion between two industrial conveyors. This trajectory is used to show the behavior of the model $\mathcal{M}_{Id+F}$ in case of trajectory with/without a payload. In case of pick-and-place tasks with heavy payloads, the performance would decrease, since the model does not take into consideration the different weights at the end-effector. The purpose of the payload identification is to increase the accuracy when the lifted payload m_P is greater than zero. Figure 3.8 shows the estimated payload $\hat{m}_p$ using the method (3.13). This model achieves good results in terms of transient time and accuracy. Table 3.4 shows the values of

	$m_P = 0$		$m_P = \hat{m}_P$	
	μ	σ	μ	σ
τ_1 (Nm)	13.91	13.27	0.42	7.39
τ_2 (Nm)	7.27	16.52	-0.02	7.49
τ_3 (Nm)	5.88	5.96	0.27	5.40

Table 3.4: Mean error and standard deviation of the $\mathcal{M}_{Id+f}$ torque and real τ, over the trajectory υ_P, with $m_P = 0$ and $m_P = \hat{m}_P$.

the mean error and standard deviation of the torque error with $m_P = \hat{m}_p$ and without payload, $m_P = 0$.

3.3.5 Feed-forward Control

Future research will focus on the implementation of a feedforward torque control loop, see Figure 3.9, based on the $\mathcal{M}_{Id+F}$ model.

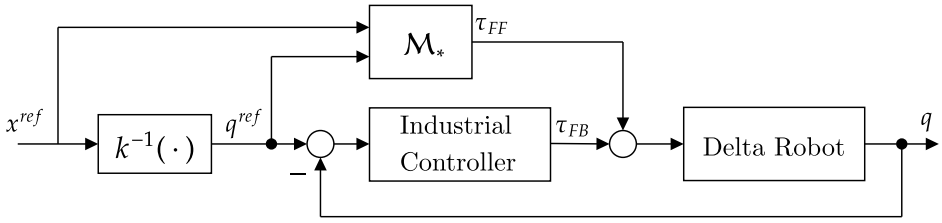


Figure 3.9: Control loop with feed-forward torque τ_{FF} based on the identified model $\mathcal{M}_*$ and feedback torque τ_{FB}.

3.4 Novel Inverse Dynamic Model

The offline parameter identification process described in Section 3.3.2 has some flaws: the offline minimization formalized in equation (3.8) is highly time-consuming and, furthermore, the identified parameters $\rho_i^\star$, $i \in \{1, 2\}$ depend on the set of trajectories Υ.

In this section we propose a novel inverse dynamic model, to which we will refer with the name of $\mathcal{M}_f$, that does not need to rely on any parameter identification. The effect of friction at the active and passive joints is not considered in $\mathcal{M}_f$, but it remains possible to estimate it separately. The model is derived by discarding the assumption A2 and then by modeling the passive links as homogeneous rods of mass m_2, length l_2 with center of mass positioned at $a_2 = l_2/2$. They oscillate with respect to passive revolute joints placed in $\boldsymbol{j}_i'$ and so their inertia is

$$\mathcal{I}_2 = \frac{1}{3} m_2 l_2^2.$$

Since the end-effector is only capable of translating in the operational space, it can still be modeled as a point mass m_0 placed in $\boldsymbol{x}$ as in (3.1).

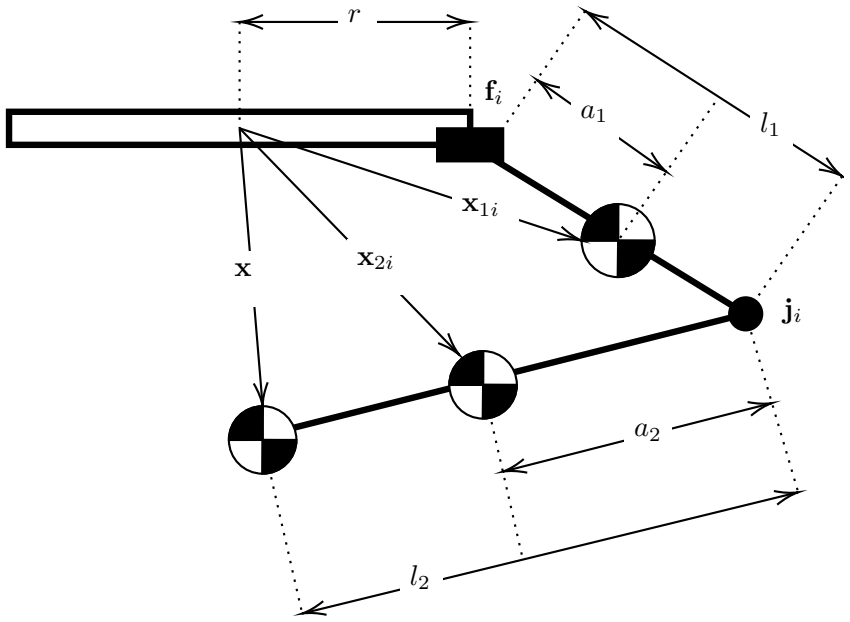


Figure 3.10: The position of the center of mass of the first link is placed at distance a_1 with respect to $\boldsymbol{f}_i$ along the active link. The position of the center of mass of the second link is placed at distance a_2 with respect to $\boldsymbol{j}_i$ along the passive link.

In what follows we will adopt the compact notation $\mathrm{c}\alpha$ for $\cos\alpha$ and $\mathrm{s}\alpha$ for $\sin\alpha$; furthermore, we introduce the new variables β_i as the sum of the angle of the joints connected to the active links, and the angle of the passive joints whose rotation lies on the same plane of the active link to which it is connected

$$\beta_i \triangleq \theta_i + \gamma_i.$$

Figure 3.10 shows the coordinates of the center of mass of each actuated link $\boldsymbol{x}_{1i}$ and the coordinates of the center of mass of each passive link $\boldsymbol{x}_{2i}$

$$\boldsymbol{x}_{1i} = \begin{bmatrix} (r + a_1\mathrm{c}\theta_i)\mathrm{c}\alpha_i \\ (r + a_1\mathrm{c}\theta_i)\mathrm{s}\alpha_i \\ a_1\mathrm{s}\theta_i \end{bmatrix} \quad \boldsymbol{x}_{2i} = \begin{bmatrix} (r + l_1\mathrm{c}\theta_i + a_2\mathrm{c}\beta_i\mathrm{c}\psi_i)\mathrm{c}\alpha_i - a_2\mathrm{s}\psi_i\mathrm{s}\alpha_i \\ (r + l_1\mathrm{c}\theta_i + a_2\mathrm{c}\beta_i\mathrm{c}\psi_i)\mathrm{s}\alpha_i + a_2\mathrm{c}\psi_i\mathrm{s}\alpha_i \\ l_1\mathrm{s}\theta_i + a_2\mathrm{s}\beta_i\mathrm{c}\psi_i \end{bmatrix}, \tag{3.14}$$

while the coordinates of the end-effector is $\boldsymbol{x}$.

The corresponding velocity vectors computed by differentiating equations (3.1) and (3.14) are

$$\dot{\boldsymbol{x}}_{1i} = \begin{bmatrix} -a_1\,\dot{\theta}_i\mathrm{s}\theta_i\mathrm{c}\alpha_i \\ -a_1\,\dot{\theta}_i\mathrm{s}\theta_i\mathrm{s}\alpha_i \\ a_1\,\dot{\theta}_i\mathrm{c}\theta_i \end{bmatrix}, \quad \dot{\boldsymbol{x}} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix}, \tag{3.15}$$

$$\dot{\boldsymbol{x}}_{2i} = \begin{bmatrix} -\left(l_1\dot{\theta}_i\mathrm{s}\theta_i + a_2\dot{\beta}_i\mathrm{s}\beta_i\mathrm{c}\psi_i + a_2\mathrm{c}\beta_i\dot{\psi}_i\mathrm{s}\psi_i\right)\mathrm{c}\alpha_i - a_2\,\dot{\psi}_i\mathrm{c}\psi_i\mathrm{s}\alpha_i \\ -\left(l_1\dot{\theta}_i\mathrm{s}\theta_i + a_2\dot{\beta}_i\mathrm{s}\beta_i\mathrm{c}\psi_i + a_2\mathrm{c}\beta_i\dot{\psi}_i\mathrm{s}\psi_i\right)\mathrm{s}\alpha_i - a_2\,\dot{\psi}_i\mathrm{s}\psi_i\mathrm{s}\alpha_i \\ l_1\dot{\theta}_i\mathrm{c}\theta_i + a_2(\dot{\beta}_i\mathrm{c}\beta_i\mathrm{c}\psi_i - \mathrm{s}\beta_i\dot{\psi}_i\mathrm{s}\psi_i) \end{bmatrix}. \tag{3.16}$$

3.4.1 Kinetic Energy Contribution

In $\mathcal{M}_f$ we also consider the kinetic energy of the rotors $\mathcal{T}_3$. The contribution can be re-written as

$$\begin{aligned}
\mathcal{T}_0 &= \frac{1}{2} m_0 ||\dot{\boldsymbol{x}}||^2, \\
\mathcal{T}_1 &= \sum_{i=1}^{3} \left(\frac{1}{2} m_1 ||\dot{\boldsymbol{x}}_{1i}||^2 + \frac{1}{2} \mathcal{I}_1 \dot{\theta}_i^2 \right), \\
\mathcal{T}_2 &= \sum_{i=1}^{3} \left(\frac{1}{2} m_1 ||\dot{\boldsymbol{x}}_{2i}||^2 + \frac{1}{2} \mathcal{I}_2 \dot{\beta}_i^2 + \frac{1}{2} \mathcal{I}_2 \dot{\psi}_i^2 \right), \\
\mathcal{T}_3 &= \frac{1}{2} \mathcal{I}_3 k_r^2 \sum_{i=1}^{3} \dot{\theta}_i^2,
\end{aligned} \tag{3.17}$$

where k_r is the motor gear ratio and $\mathcal{I}_3$ is the inertia of the rotor. For clarification, the following terms are explicitly computed

$$\begin{aligned}
||\dot{\boldsymbol{x}}_{1i}||^2 &= a_1^2 \dot{\theta}_i^2, \\
||\dot{\boldsymbol{x}}_{2i}||^2 &= a_1^2 k_2^2 + a_2^2 \mathrm{s}^2 \alpha_i \dot{\psi}^2 + k_1 l_2 \mathrm{s}\alpha_i \dot{\psi}_i \mathrm{c}(\alpha_i - \psi_i) + k_1^2 + k_2 l_1 l_2 \dot{\theta}_i \mathrm{c}\theta_i + l_1^2 \dot{\theta}_i^2 \mathrm{c}^2 \theta_i, \\
||\dot{\boldsymbol{x}}||^2 &= \dot{x}^2 + \dot{y}^2 + \dot{z}^2,
\end{aligned}$$

where k_1 and k_2 group some of the terms for simplicity,

$$k_1 \triangleq \dot{\theta}_i l_1 \mathrm{s}\theta_i + a_2 \left(\dot{\beta}_i \mathrm{s}\beta_i \mathrm{c}\psi_i + \dot{\psi}_i \mathrm{c}\beta_i \mathrm{s}\psi_i \right), \tag{3.18}$$

$$k_2 \triangleq \dot{\beta}_i \mathrm{c}\beta_i \mathrm{c}\psi_i - \dot{\psi}_i \mathrm{s}\beta_i \mathrm{s}\psi_i. \tag{3.19}$$

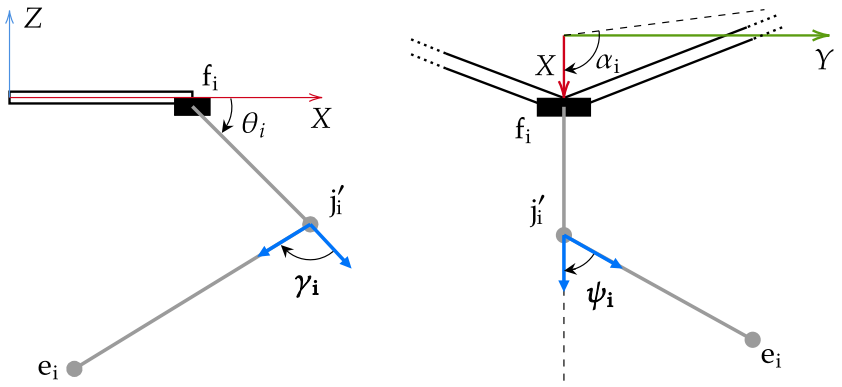


Figure 3.11: The passive angles γ_i and ψ_i represented on the planes $\{O; X, Z\}$ and $\{O; X, Y\}$, respectively. The unit vectors of the links projections are drawn in blue.

The angle ψ_i is between the projections of $\overline{\boldsymbol{f_i j_i'}}$ and $\overline{\boldsymbol{j_i e_i}}$ onto the $\{O; X, Y\}$, and γ_i is the angle betweem the projections of $\overline{\boldsymbol{f_i j_i'}}$ and $\overline{\boldsymbol{j_i e_i}}$ onto $\{O; X, Z\}$ plane, as shown in Figure 3.11. They cannot be expressed as functions of the

actuated joint angles θ_i, therefore we define them as function of the generalized coordinates $\boldsymbol{q}$ defined in (3.4). They are expressed as

$$\psi_i \triangleq \arccos(h_i) \qquad \gamma_i \triangleq \arccos(n_i), \tag{3.20}$$

with

$$h_i \triangleq \frac{\mathrm{c}\theta_i(x\mathrm{c}\alpha_i + y\mathrm{s}\alpha_i) - r\mathrm{c}\theta_i - l_1\mathrm{c}^2\theta_i}{l_2},$$
$$n_i \triangleq \frac{x\mathrm{c}\theta_i\mathrm{c}\alpha_i - (r\mathrm{c}\theta_i + l_1\mathrm{c}^2\theta_i)\mathrm{c}^2\alpha_i + z\mathrm{s}\theta_i - l_1\mathrm{s}^2\theta_i}{l_2}.$$

Definitions (3.20) are derived by considering the projections of the links onto the planes $\{O; X, Y\}$ and $\{O; X, Z\}$, respectively. The velocities of the passive joints in equations (3.17) are given by

$$\dot{\psi}_i = \frac{-1}{\sqrt{1-h_i^2}}\dot{h}_i, \qquad \dot{\gamma}_i = \frac{-1}{\sqrt{1-n_i^2}}\dot{n}_i. \tag{3.21}$$

The total kinetic energy contribution $\mathcal{T}$ is the sum of all the previous contributions

$$\mathcal{T} = \sum_{i=0}^{3} \mathcal{T}_i. \tag{3.22}$$

3.4.2 Potential Energy Contribution

The potential energies of the end-effector $\mathcal{V}_0$, of the active links $\mathcal{V}_1$ and the passive links $\mathcal{V}_2$, are

$$\mathcal{V}_0 = m_0 g z, \tag{3.23}$$

$$\mathcal{V}_1 = m_1 g a_1 \sum_{i=1}^{3} \mathrm{s}\theta_i, \tag{3.24}$$

$$\mathcal{V}_2 = m_2 g \left(l_1 \sum_{i=1}^{3} \mathrm{s}\theta_i + a_2 \sum_{i=1}^{3} \mathrm{s}\beta_i \mathrm{c}\psi_i \right), \tag{3.25}$$

The total potential energy $\mathcal{V}$ is

$$\mathcal{V} = \sum_{i=0}^{2} \mathcal{V}_i. \tag{3.26}$$

The rotors do not change position with respect to the Z-axis, therefore it is not necessary to compute their potential energy $\mathcal{V}_3$.

3.4.3 Delta Robot Novel Inverse Dynamics

The dynamics of a Delta Robot is obtained by solving the constrained Euler-Lagrange equations of the Lagrangian of the system in (3.3). By expliciting the terms we get

$$\frac{d}{dt}\left(\frac{\partial \mathcal{T}}{\partial \dot{\boldsymbol{q}}}\right) - \frac{d}{dt}\left(\frac{\partial \mathcal{V}}{\partial \dot{\boldsymbol{q}}}\right) - \frac{\partial \mathcal{T}}{\partial \boldsymbol{q}} + \frac{\partial \mathcal{V}}{\partial \boldsymbol{q}} = \boldsymbol{\tau}_g + \boldsymbol{\lambda}\frac{\partial \mathcal{F}(\boldsymbol{q})}{\partial \boldsymbol{q}}. \tag{3.27}$$

with

$$\frac{d}{dt}\left(\frac{\partial \mathcal{V}}{\partial \dot{\boldsymbol{q}}}\right) = 0, \tag{3.28}$$

since there is no dependency on $\dot{\boldsymbol{q}}$ in $\mathcal{V}$. The time derivative of the partial derivatives of the kinetic energy with respect to $\dot{\boldsymbol{q}}$ are shown in equations (3.38).

$$\begin{aligned}
\frac{d}{dt}\left(\frac{\partial \mathcal{T}}{\partial \dot{x}}\right) &= \frac{1}{2}m_2 \sum_{i=1}^{3} \frac{d}{dt}\frac{\partial ||\dot{\boldsymbol{x}}_{2i}||^2}{\partial \dot{x}} \\
&\quad + \mathcal{I}_2 \sum_{i=1}^{3} \left(\ddot{\beta}_i \frac{\partial \dot{\gamma}_i}{\partial \dot{x}} + \dot{\beta}_i \frac{d}{dt}\frac{\partial \dot{\gamma}_i}{\partial \dot{x}} + \ddot{\psi}_i \frac{\partial \dot{\psi}_i}{\partial \dot{x}} + \dot{\psi}_i \frac{d}{dt}\frac{\partial \dot{\psi}_i}{\partial \dot{x}} \right), \qquad (3.29a) \\
\frac{d}{dt}\left(\frac{\partial \mathcal{T}}{\partial \dot{y}}\right) &= \frac{1}{2}m_2 \sum_{i=1}^{3} \frac{d}{dt}\frac{\partial ||\dot{\boldsymbol{x}}_{2i}||^2}{\partial \dot{y}} \\
&\quad + \mathcal{I}_2 \sum_{i=1}^{3} \left(\ddot{\beta}_i \frac{\partial \dot{\gamma}_i}{\partial \dot{y}} + \dot{\beta}_i \frac{d}{dt}\frac{\partial \dot{\gamma}_i}{\partial \dot{y}} + \ddot{\psi}_i \frac{\partial \dot{\psi}_i}{\partial \dot{y}} + \dot{\psi}_i \frac{d}{dt}\frac{\partial \dot{\psi}_i}{\partial \dot{y}} \right), \qquad (3.29b) \\
\frac{d}{dt}\left(\frac{\partial \mathcal{T}}{\partial \dot{z}}\right) &= \frac{1}{2}m_2 \sum_{i=1}^{3} \frac{d}{dt}\frac{\partial ||\dot{\boldsymbol{x}}_{2i}||^2}{\partial \dot{z}} \\
&\quad + \mathcal{I}_2 \sum_{i=1}^{3} \left(\ddot{\beta}_i \frac{\partial \dot{\gamma}_i}{\partial \dot{z}} + \dot{\beta}_i \frac{d}{dt}\frac{\partial \dot{\gamma}_i}{\partial \dot{z}} + \ddot{\psi}_i \frac{\partial \dot{\psi}_i}{\partial \dot{z}} + \dot{\psi}_i \frac{d}{dt}\frac{\partial \dot{\psi}_i}{\partial \dot{z}} \right), \qquad (3.29c) \\
\frac{d}{dt}\left(\frac{\partial \mathcal{T}}{\partial \dot{\theta}_i}\right) &= \frac{1}{2}m_2 \frac{d}{dt}\frac{\partial ||\dot{\boldsymbol{x}}_{2i}||^2}{\partial \dot{\theta}_i} + \left(\frac{1}{4}m_1 l_1^2 + \mathcal{I}_1 + \mathcal{I}_3 k_r^2\right)\ddot{\theta}_i \\
&\quad + \mathcal{I}_2 \left(\ddot{\beta}_i \left(1 + \frac{\partial \dot{\gamma}_i}{\partial \dot{\theta}_i}\right) + \dot{\beta}_i \frac{d}{dt}\left(\frac{\partial \dot{\gamma}_i}{\partial \dot{\theta}_i}\right) \right). \qquad (3.29d)
\end{aligned}$$

The partial derivatives of the kinetic energy with respect to $\boldsymbol{q}$ are

$$\begin{aligned}
\frac{\partial \mathcal{T}}{\partial x} &= \frac{1}{2}m_2 \sum_{i=1}^{3} \frac{\partial ||\dot{\boldsymbol{x}}_{2i}||^2}{\partial x} + \mathcal{I}_2 \sum_{i=1}^{3} \left(\frac{\partial \dot{\gamma}_i}{\partial x}\dot{\beta}_i + \dot{\psi}_i \frac{\partial \dot{\psi}_i}{\partial x} \right), \\
\frac{\partial \mathcal{T}}{\partial y} &= \frac{1}{2}m_2 \sum_{i=1}^{3} \frac{\partial ||\dot{\boldsymbol{x}}_{2i}||^2}{\partial y} + \mathcal{I}_2 \sum_{i=1}^{3} \left(\frac{\partial \dot{\gamma}_i}{\partial y}\dot{\beta}_i + \dot{\psi}_i \frac{\partial \dot{\psi}_i}{\partial y} \right), \\
\frac{\partial \mathcal{T}}{\partial z} &= \frac{1}{2}m_2 \sum_{i=1}^{3} \frac{\partial ||\dot{\boldsymbol{x}}_{2i}||^2}{\partial z} + \mathcal{I}_2 \sum_{i=1}^{3} \left(\frac{\partial \dot{\gamma}_i}{\partial z}\dot{\beta}_i + \dot{\psi}_i \frac{\partial \dot{\psi}_i}{\partial z} \right), \\
\frac{\partial \mathcal{T}}{\partial \theta_i} &= \frac{1}{2}m_2 \frac{\partial ||\dot{\boldsymbol{x}}_{2i}||^2}{\partial \theta_i} + \mathcal{I}_2 \left(\dot{\gamma}_i \frac{\partial \dot{\gamma}_i}{\partial \theta_i} + \dot{\psi}_i \frac{\partial \psi_i}{\partial \theta_i} \right).
\end{aligned}$$

For what concern the potential energy, we have the following equations

$$\begin{aligned}
\frac{\partial \mathcal{V}}{\partial x} &= -\, m_2 g a_2 \sum_{i=1}^{3} \left(\frac{\partial \gamma_i}{\partial x} \mathrm{c}\beta_i \mathrm{c}\psi_i - \mathrm{s}\beta_i \frac{\partial \psi_i}{\partial x} \mathrm{s}\psi_i \right), \\
\frac{\partial \mathcal{V}}{\partial y} &= -\, m_2 g a_2 \sum_{i=1}^{3} \left(\frac{\partial \gamma_i}{\partial y} \mathrm{c}\beta_i \mathrm{c}\psi_i - \mathrm{s}\beta_i \frac{\partial \psi_i}{\partial y} \mathrm{s}\psi_i \right), \\
\frac{\partial \mathcal{V}}{\partial z} &= -\, m_2 g a_2 \sum_{i=1}^{3} \left(\frac{\partial \gamma_i}{\partial z} \mathrm{c}\beta_i \mathrm{c}\psi_i - \mathrm{s}\beta_i \frac{\partial \psi_i}{\partial z} \mathrm{s}\psi_i \right) - m_2 g, \\
\frac{\partial \mathcal{V}}{\partial \theta_i} &= -\left(\frac{m_1}{2} + m_2 \right) g\, l_1 \mathrm{c}\theta_i - m_2 g a_2 \left[\left(1 + \frac{\partial \gamma_i}{\partial \theta_i} \right) \mathrm{c}\beta_i \mathrm{c}\psi_i - \mathrm{s}\beta_i \frac{\partial \psi_i}{\partial \theta_i} \mathrm{s}\psi_i \right].
\end{aligned}$$

The accelerations $\ddot{\psi}_i$ and $\ddot{\gamma}_i$ of the passive joints are given by

$$\ddot{\psi}_i = \frac{-\left[\ddot{h}_i (1 - h_i^2) + h_i \dot{h}_i^2 \right]}{(1 - h_i^2)^{3/2}} \qquad \ddot{\gamma}_i = \frac{-\left[\ddot{n}_i (1 - n_i^2) + n_i \dot{n}_i^2 \right]}{(1 - n_i^2)^{3/2}},$$

for $i \in \{1, 2, 3\}$. The partial derivatives of velocities in equations (3.21) are

$$\frac{\partial \dot{\gamma}_i}{\partial \dot{x}} = \frac{-1}{\sqrt{1 - n_i^2}} \frac{\partial \dot{n}_i}{\partial \dot{x}}, \qquad \frac{\partial \dot{\psi}_i}{\partial \dot{x}} = \frac{-1}{\sqrt{1 - h_i^2}} \frac{\partial \dot{h}_i}{\partial \dot{x}}, \tag{3.30}$$

and analogous expressions hold true also for derivatives with respect to $\dot{y}$, $\dot{z}$ and $\dot{\theta}_i$. The total derivatives of equations (3.30) are

$$\frac{d}{dt} \frac{\partial \dot{\gamma}_i}{\partial \dot{x}} = -\left[\frac{\partial \dot{n}_i}{\partial \dot{x}} n_i \dot{n}_i + \frac{d}{dt} \frac{\partial \dot{n}_i}{\partial \dot{x}} (1 - n_i^2) \right] \frac{1}{(1 - n_i^2)^{3/2}}, \tag{3.31}$$

$$\frac{d}{dt} \frac{\partial \dot{\psi}_i}{\partial \dot{x}} = -\left[\frac{\partial \dot{h}_i}{\partial \dot{x}} h_i \dot{h}_i + \frac{d}{dt} \frac{\partial \dot{h}_i}{\partial \dot{x}} (1 - h_i^2) \right] \frac{1}{(1 - h_i^2)^{3/2}}. \tag{3.32}$$

Similar equations hold for the derivatives with respect to $\dot{y}$, $\dot{z}$ and $\dot{\theta}_i$. All these derivatives are involved in the kinetic energy $\mathcal{T}$, while in the potential energy $\mathcal{V}$ we only have derivatives with respect to Cartesian and joint positions of passive angles positions and velocities, defined by equations (3.20) and (3.21), respectively. Thus, we have

$$\begin{aligned}
\frac{\partial \gamma_i}{\partial x} &= \frac{-1}{\sqrt{1 - n_i^2}} \frac{\partial n_i}{\partial x}, \qquad & \frac{\partial \dot{\gamma}_i}{\partial x} &= -\left[\frac{\partial n_i}{\partial x} n_i^2 + \frac{\partial \dot{n}_i}{\partial x} (1 - n_i^2) \right] \frac{1}{(1 - n_i^2)^{3/2}}, \\
\frac{\partial \psi_i}{\partial x} &= \frac{-1}{\sqrt{1 - h_i^2}} \frac{\partial h_i}{\partial x}, \qquad & \frac{\partial \dot{\psi}_i}{\partial x} &= -\left[\frac{\partial h_i}{\partial x} h_i^2 + \frac{\partial \dot{h}_i}{\partial x} (1 - h_i^2) \right] \frac{1}{(1 - h_i^2)^{3/2}},
\end{aligned}$$

where, as before, the same pattern holds true for derivatives with respect to y, z and $\dot{\theta}_i$. The equations of motion also contain the derivatives of $||\dot{\boldsymbol{x}}_{2i}||^2$ with respect to generalized coordinates and velocities, shown in (3.35), where the time derivatives of k_1 and k_2 terms, defined by equations (3.33a) and (3.33b), are

$$\dot{k}_1 = a_2\left(\ddot{\beta}_i \mathrm{s}\beta_i \mathrm{c}\psi_i + \dot{\beta}_i^2 \mathrm{c}\beta_i \mathrm{c}\psi_i\right) + a_2\left[\mathrm{c}\beta_i(\ddot{\psi}_i \mathrm{s}\psi_i + \dot{\psi}_i^2 \mathrm{c}\psi_i) - 2\dot{\beta}_i \mathrm{s}\beta_i \dot{\psi}_i \mathrm{s}\psi_i\right] + l_1(\ddot{\theta}_i \mathrm{s}\theta_i + \dot{\theta}_i^2 \mathrm{c}\theta_i), \tag{3.33a}$$

$$\dot{k}_2 = \ddot{\beta}_i \mathrm{c}\beta_i \mathrm{c}\psi_i - \dot{\beta}_i^2 \mathrm{s}\beta_i \mathrm{c}\psi_i - 2\dot{\beta}_i \mathrm{c}\beta_i \dot{\psi}_i \mathrm{s}\psi_i - \mathrm{s}\beta_i(\ddot{\psi}_i \mathrm{s}\psi_i + \dot{\psi}_i^2 \mathrm{c}\psi_i). \tag{3.33b}$$

The partial derivatives are

$$\frac{\partial k_1}{\partial \dot{x}} = a_2\left(\frac{\partial \dot{\gamma}_i}{\partial \dot{x}} \mathrm{s}\beta_i \mathrm{c}\psi_i + \frac{\partial \dot{\psi}_i}{\partial \dot{x}} \mathrm{c}\beta_i \mathrm{s}\psi_i\right), \quad \frac{\partial k_2}{\partial \dot{x}} = \frac{\partial \dot{\gamma}_i}{\partial \dot{x}} \mathrm{c}\beta_i \mathrm{c}\psi_i - \frac{\partial \dot{\psi}_i}{\partial \dot{x}} \mathrm{s}\beta_i \mathrm{s}\psi_i,$$

and the total derivatives are showed in (3.34).

$$\frac{d}{dt}\frac{\partial k_1}{\partial \dot{x}} = \left[\frac{d}{dt}\frac{\partial \dot{\gamma}_i}{\partial \dot{x}} \mathrm{s}\beta_i + \left(\frac{\partial \dot{\gamma}_i}{\partial \dot{x}}\dot{\beta}_i + \frac{\partial \dot{\psi}_i}{\partial \dot{x}}\dot{\psi}_i\right)\mathrm{c}\beta_i\right] a_2 \mathrm{c}\psi_i + \left[\frac{d}{dt}\frac{\partial \dot{\psi}_i}{\partial \dot{x}} \mathrm{c}\beta_i - \left(\frac{\partial \dot{\gamma}_i}{\partial \dot{x}}\dot{\beta}_i + \frac{\partial \dot{\gamma}_i}{\partial \dot{x}}\dot{\psi}_i\right)\mathrm{s}\beta_i\right] a_2 \mathrm{s}\psi_i, \tag{3.34a}$$

$$\frac{d}{dt}\frac{\partial k_2}{\partial \dot{x}} = \left[\frac{d}{dt}\frac{\partial \dot{\gamma}_i}{\partial \dot{x}} \mathrm{c}\beta_i - \left(\frac{\partial \dot{\gamma}_i}{\partial \dot{x}}\dot{\beta}_i - \frac{\partial \dot{\psi}_i}{\partial \dot{x}}\dot{\psi}_i\right)\mathrm{s}\beta_i\right] \mathrm{c}\psi_i + \left[\frac{d}{dt}\frac{\partial \dot{\psi}_i}{\partial \dot{x}} \mathrm{s}\beta_i + \left(\frac{\partial \dot{\gamma}_i}{\partial \dot{x}}\dot{\psi}_i + \frac{\partial \dot{\psi}_i}{\partial \dot{x}}\dot{\beta}_i\right)\mathrm{c}\beta_i\right] \mathrm{s}\psi_i. \tag{3.34b}$$

$$\begin{aligned}\frac{d}{dt}\frac{\partial ||\dot{\boldsymbol{x}}_{2i}||^2}{\partial \dot{x}} &= 2\left(\dot{k}_1\frac{\partial k_1}{\partial \dot{x}} + k_1\frac{d}{dt}\frac{\partial k_1}{\partial \dot{x}}\right) + \frac{l_2^2}{2}\mathrm{s}^2\alpha_i\left(\ddot{\psi}_i\frac{\partial \dot{\psi}_i}{\partial \dot{x}} + \dot{\psi}_i\frac{d}{dt}\frac{\partial \dot{\psi}_i}{\partial \dot{x}}\right) \\ &+ l_2 \mathrm{s}\alpha_i\left[(\dot{\psi}_i \mathrm{c}\psi_i \mathrm{s}\alpha_i - \dot{\psi}_i \mathrm{s}\psi_i \mathrm{c}\alpha_i)\left(\frac{\partial k_1}{\partial \dot{x}}\dot{\psi}_i + k_1\frac{\partial \dot{\psi}_i}{\partial \dot{x}}\right)\right] \\ &+ l_2 \mathrm{s}\alpha_i\left[(\mathrm{c}\psi_i \mathrm{c}\alpha_i + \mathrm{s}\psi_i \mathrm{s}\alpha_i)\left(\frac{d}{dt}\frac{\partial k_1}{\partial \dot{x}}\dot{\psi}_i + \frac{\partial k_1}{\partial \dot{x}}\ddot{\psi}_i + \dot{k}_1\frac{\partial \dot{\psi}_i}{\partial \dot{x}} + k_1\frac{d}{dt}\frac{\partial \dot{\psi}_i}{\partial \dot{x}}\right)\right] \\ &+ a_2\left(\dot{k}_2\frac{\partial k_2}{\partial \dot{x}} + k_2\frac{d}{dt}\frac{\partial k_2}{\partial \dot{x}}\right) + l_2 l_1\left(\ddot{\theta}_i \mathrm{c}\theta_i\frac{\partial k_2}{\partial \dot{x}} - \dot{\theta}_i^2 \mathrm{s}\theta_i\frac{\partial k_2}{\partial \dot{x}} + \dot{\theta}_i \mathrm{c}\theta_i\frac{d}{dt}\frac{\partial k_2}{\partial \dot{x}}\right).\end{aligned} \tag{3.35}$$

$$\frac{d}{dt}\frac{\partial||\dot{\boldsymbol{x}}_{2i}||^2}{\partial\dot{\theta}_i} = 2\left(\dot{k}_1\frac{\partial k_1}{\partial\dot{\theta}_i} + k_1\frac{d}{dt}\frac{\partial k_1}{\partial\dot{\theta}_i}\right) + \frac{l_2^2}{2}\mathrm{s}^2\alpha_i\left(\ddot{\psi}_i\frac{\partial\dot{\psi}_i}{\partial\dot{\theta}_i} + \dot{\psi}_i\frac{d}{dt}\frac{\partial\dot{\psi}_i}{\partial\dot{\theta}_i}\right)$$
$$+ l_2\mathrm{s}\alpha_i\left[(\dot{\psi}_i\mathrm{c}\psi_i\mathrm{s}\alpha_i - \dot{\psi}_i\mathrm{s}\psi_i\mathrm{c}\alpha_i)\left(\frac{\partial k_1}{\partial\dot{\theta}_i}\dot{\psi}_i + k_1\frac{\partial\dot{\psi}_i}{\partial\dot{\theta}_i}\right)\right]$$
$$+ l_2\mathrm{s}\alpha_i\left[(\mathrm{c}\psi_i\mathrm{c}\alpha_i + \mathrm{s}\psi_i\mathrm{s}\alpha_i)\left(\frac{d}{dt}\frac{\partial k_1}{\partial\dot{\theta}_i}\dot{\psi}_i + \frac{\partial k_1}{\partial\dot{\theta}_i}\ddot{\psi}_i + \dot{k}_1\frac{\partial\dot{\psi}_i}{\partial\dot{\theta}_i} + k_1\frac{d}{dt}\frac{\partial\dot{\psi}_i}{\partial\dot{\theta}_i}\right)\right]$$
$$+ a_2\left(\dot{k}_2\frac{\partial k_2}{\partial\dot{\theta}_i} + k_2\frac{d}{dt}\frac{\partial k_2}{\partial\dot{\theta}_i}\right) + 2l_1(\dot{\theta}_i\mathrm{c}^2\theta_i - 2\dot{\theta}_i^2\mathrm{c}\theta_i\mathrm{s}\theta_i)$$
$$+ l_2l_1\left[\mathrm{c}\theta_i\left(\dot{k}_2 + \ddot{\theta}_i\frac{\partial k_2}{\partial\dot{\theta}_i} + \dot{\theta}_i\frac{d}{dt}\frac{\partial k_2}{\partial\dot{\theta}_i}\right) - \dot{\theta}_i\mathrm{s}\theta_i\left(k_2 + \ddot{\theta}_i\frac{\partial k_2}{\partial\dot{\theta}_i}\right)\right]. \tag{3.36}$$

$$\frac{\partial||\dot{\boldsymbol{x}}_{2i}||^2}{\partial\theta_i} = \frac{\partial k_1}{\partial\theta_i}k_1 + l_1l_2\dot{\theta}_i\left(\frac{\partial k_2}{\partial\theta_i}\mathrm{c}\theta_i - k_2\mathrm{s}\theta_i\right) + \frac{l_2^2}{2}\left(\frac{\partial k_2}{\partial\theta_i}k_2 + \frac{\partial\dot{\psi}_i}{\partial\theta_i}\dot{\psi}_i\mathrm{s}^2\alpha_i\right) - l_1^2\dot{\theta}_i^2\mathrm{s}2\theta_i$$
$$+ l_2\mathrm{s}\alpha_i(\mathrm{c}\psi_i\mathrm{c}\alpha_i + \mathrm{s}\psi_i\mathrm{s}\alpha_i)\left[\frac{\partial k_1}{\partial\theta_i}\dot{\psi}_i + k_1\left(\frac{\partial\psi_i}{\partial\theta_i} + \dot{\psi}_i\frac{\partial\psi_i}{\partial\theta_i}\right)\right]. \tag{3.37}$$

where

$$\frac{\partial k_1}{\partial\theta_i} = a_2\left[\left(\frac{\partial\dot{\gamma}_i}{\partial\theta_i}\mathrm{c}\psi_i - \dot{\beta}_i\frac{\partial\psi_i}{\partial\theta_i}\mathrm{s}\psi_i - \left(1 + \frac{\partial\gamma_i}{\partial\theta_i}\right)\dot{\psi}_i\mathrm{s}\psi_i\right)\mathrm{s}\beta_i\right]$$
$$+ a_2\left[\left(\frac{\partial\dot{\psi}_i}{\partial\theta_i}\mathrm{s}\psi_i + \dot{\beta}_i\frac{\partial\gamma_i}{\partial\theta_i}\mathrm{c}\psi_i + \frac{\partial\psi_i}{\partial\theta_i}\dot{\psi}_i\mathrm{c}\psi_i\right)\mathrm{c}\beta_i\right] + l_1\dot{\theta}_i\mathrm{c}\theta_i,$$
$$\frac{\partial k_2}{\partial\theta_i} = \left[\frac{\partial\dot{\gamma}_i}{\partial\theta_i}\mathrm{c}\psi_i - \dot{\beta}_i\frac{\partial\psi_i}{\partial\theta_i}\mathrm{s}\psi_i - \left(1 + \frac{\partial\gamma_i}{\partial\theta_i}\right)\dot{\psi}_i\mathrm{s}\psi_i\right]\mathrm{c}\beta_i$$
$$+ \left[\frac{\partial\dot{\psi}_i}{\partial\theta_i}\mathrm{s}\psi_i - \dot{\beta}_i\left(1 + \frac{\partial\gamma_i}{\partial\theta_i}\right)\mathrm{c}\psi_i - \frac{\partial\psi_i}{\partial\theta_i}\dot{\psi}_i\mathrm{c}\psi_i\right]\mathrm{s}\beta_i.$$

The derivatives with respect to $\dot{y}$ and $\dot{z}$ follow the same pattern. Derivatives about joint variables θ_i and $\dot{\theta}_i$ are given in (3.36). The partial and total derivatives of k_1 and k_2 terms, defined by equations (3.33a) and (3.33b), are

$$\frac{\partial k_1}{\partial\dot{\theta}_i} = l_1\mathrm{s}\theta_i + a_2\left[\left(1 + \frac{\partial\dot{\gamma}_i}{\partial\dot{\theta}_i}\right)\mathrm{s}\beta_i\mathrm{c}\psi_i + \mathrm{s}\beta_i\frac{\partial\dot{\psi}_i}{\partial\dot{\theta}_i}\mathrm{s}\psi_i\right],$$
$$\frac{\partial k_2}{\partial\dot{\theta}_i} = \left(1 + \frac{\partial\dot{\gamma}_i}{\partial\dot{\theta}_i}\right)\mathrm{c}\beta_i\mathrm{c}\psi_i - \mathrm{s}\beta_i\frac{\partial\dot{\psi}_i}{\partial\dot{\theta}_i}\mathrm{s}\psi_i,$$

$$
\begin{aligned}
\frac{d}{dt}\frac{\partial k_1}{\partial \dot{\theta}_i} &= a_2\left[\left(\frac{d}{dt}\frac{\dot{\gamma}_i}{\partial \dot{\theta}_i}\mathrm{s}\beta_i + 2\dot{\beta}_i\left(1+\frac{\partial \dot{\gamma}_i}{\partial \dot{\theta}_i}\right)\mathrm{c}\beta_i\right)\mathrm{c}\psi_i\right] \\
&\quad + a_2\left[\mathrm{c}\beta_i\left(\frac{d}{dt}\frac{\partial \dot{\psi}_i}{\partial \dot{\theta}_i}\mathrm{s}\psi_i + 2\dot{\psi}_i\frac{\partial \dot{\psi}_i}{\partial \dot{\theta}_i}\mathrm{c}\psi_i\right)\right] \\
&\quad + 2l_1\dot{\theta}_i\mathrm{c}\theta_i - \left[l_2\mathrm{s}\beta_i\mathrm{s}\psi_i\left(\left(1+\frac{\partial \dot{\psi}_i}{\partial \dot{\theta}_i}\dot{\psi}_i\right)\dot{\psi}_i + \dot{\beta}_i\frac{\partial \dot{\psi}_i}{\partial \dot{\theta}_i}\right)\right], \\
\frac{d}{dt}\frac{\partial k_2}{\partial \dot{\theta}_i} &= \left[\frac{d}{dt}\frac{\partial \dot{\gamma}_i}{\partial \dot{\theta}_i}\mathrm{c}\beta_i - \left(\frac{\partial \dot{\gamma}_i}{\partial \dot{\theta}_i}\dot{\beta}_i + \frac{\partial \dot{\psi}_i}{\partial \dot{\theta}_i}\dot{\psi}_i\right)\mathrm{s}\beta_i\right]\mathrm{c}\psi_i \\
&\quad - \left[\frac{d}{dt}\frac{\partial \dot{\psi}_i}{\partial \dot{\theta}_i}\mathrm{s}\beta_i + \left(\frac{\partial \dot{\gamma}_i}{\partial \dot{\theta}_i}\dot{\psi}_i + \frac{\partial \dot{\psi}_i}{\partial \dot{\theta}_i}\dot{\beta}_i\right)\mathrm{c}\beta_i\right]\mathrm{s}\psi_i,
\end{aligned}
$$

Finally, we have the derivatives of $||\dot{\boldsymbol{x}}_{2i}||^2$ with respect to Cartesian and joint positions. As usual, we only show the partial derivatives with respect to x since those about y and z follow the same pattern. For Cartesian positions, we have

$$
\begin{aligned}
\frac{\partial ||\dot{\boldsymbol{x}}_{2i}||^2}{\partial x} &= l_2\mathrm{s}\alpha_i\left[\left(k_1\dot{\psi}_i\frac{\partial \psi_i}{\partial x} + \frac{\partial k_1}{\partial x}\dot{\psi}_i + k_1\frac{\partial \dot{\psi}_i}{\partial x}\right)\left(\mathrm{c}\psi_i\mathrm{c}\alpha_i + \mathrm{s}\psi_i\mathrm{s}\alpha_i\right)\right] \\
&\quad + \frac{l_2^2}{2}\left[\frac{\partial \dot{\psi}_i}{\partial x}\dot{\psi}_i\mathrm{s}^2\alpha_i + \frac{\partial k_2}{\partial x}\left(k_2 + l_1\dot{\theta}_i\mathrm{c}\theta_i\right)\right] + 2k_1\frac{\partial k_1}{\partial x},
\end{aligned}
$$

where

$$
\begin{aligned}
\frac{\partial k_1}{\partial x} &= c_2\left[\left(\frac{\partial \dot{\gamma}_i}{\partial x}\mathrm{c}\psi_i - \dot{\beta}_i\frac{\partial \psi_i}{\partial x}\mathrm{s}\psi_i - \frac{\partial \gamma_i}{\partial x}\dot{\psi}_i\mathrm{s}\psi_i\right)\mathrm{s}\beta_i\right] \\
&\quad + c_2\left[\left(\dot{\beta}_i\frac{\partial \gamma_i}{\partial x}\mathrm{c}\psi_i + \frac{\partial \dot{\psi}_i}{\partial x}\mathrm{s}\psi_i + \dot{\psi}_i\frac{\partial \dot{\psi}_i}{\partial x}\mathrm{c}\psi_i\right)\mathrm{c}\beta_i\right], \\
\frac{\partial k_2}{\partial x} &= \mathrm{c}\beta_i\left(\frac{\partial \dot{\gamma}_i}{\partial x}\mathrm{c}\psi_i - \dot{\beta}_i\frac{\partial \psi_i}{\partial x}\mathrm{s}\psi_i - \frac{\partial \gamma_i}{\partial x}\dot{\psi}_i\mathrm{s}\psi_i\right) \\
&\quad - \mathrm{s}\beta_i\left(\dot{\beta}_i\frac{\partial \gamma_i}{\partial x}\mathrm{c}\psi_i + \frac{\partial \dot{\psi}_i}{\partial x}\mathrm{s}\psi_i + \dot{\psi}_i\frac{\partial \psi_i}{\partial x}\mathrm{c}\psi_i\right),
\end{aligned}
$$

whereas, for joint positions, we end up with (3.37).

Remark 3.4.3.1. *The complete inverse dynamic model $\mathcal{M}_f$ for the Delta robot cannot be expressed in the usual form*

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

where $\boldsymbol{q} = [\theta_1\,\theta_2\,\theta_3]^T$ are the joint variables, $\boldsymbol{B}(\boldsymbol{q})$ is the inertia matrix, $\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})$ represents the Coriolis and centrifugal effects and $\boldsymbol{g}(\boldsymbol{q})$ the gravity term, because of the impossibility of finding a closed form for the passive joint angles ψ_i, γ_i that depends only on the actuated joint angles θ_i for $i \in \{1,2,3\}$. A

more complex mathematical formulation, including Cartesian and joint variables, given by equations (3.39) *and* (3.40) *is thus needed.*

The constrained Lagrangian dynamics can thus be written in compact form as

$$2\boldsymbol{A}\boldsymbol{\lambda} = \boldsymbol{b}, \tag{3.39}$$

where the matrix $\boldsymbol{A}$ and the vectors $\boldsymbol{b}$ and $\boldsymbol{\lambda}$ are defined as

$$\boldsymbol{A} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}, \quad \boldsymbol{\lambda} = \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \end{bmatrix}, \quad \boldsymbol{b} = \begin{bmatrix} \dfrac{d}{dt}\left(\dfrac{\partial \mathcal{T}}{\partial \dot{x}}\right) - \dfrac{\partial \mathcal{T}}{\partial x} + \dfrac{\partial \mathcal{V}}{\partial x} \\ \dfrac{d}{dt}\left(\dfrac{\partial \mathcal{T}}{\partial \dot{y}}\right) - \dfrac{\partial \mathcal{T}}{\partial y} + \dfrac{\partial \mathcal{V}}{\partial y} \\ \dfrac{d}{dt}\left(\dfrac{\partial \mathcal{T}}{\partial \dot{z}}\right) - \dfrac{\partial \mathcal{T}}{\partial z} + \dfrac{\partial \mathcal{V}}{\partial z} \end{bmatrix},$$

with

$$\begin{aligned} a_{1i} &= x + r\cos\alpha_i - l_1\cos\theta_i\cos\alpha_i, \\ a_{2i} &= y + r\sin\alpha_i - l_1\cos\theta_i\sin\alpha_i, \\ a_{3i} &= z - l_1\sin\theta_i, \end{aligned}$$

for $i \in \{1, 2, 3\}$. The solutions of the system (3.39) is $\boldsymbol{\lambda} = \frac{1}{2}\boldsymbol{A}^{-1}\boldsymbol{b}$, and the command torques are obtained by substituting $\boldsymbol{\lambda}$ into the following equation

$$\boldsymbol{\tau} = \boldsymbol{n}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}) + \boldsymbol{c}(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \boldsymbol{g}(\boldsymbol{q}) - 2\boldsymbol{K}(\boldsymbol{q})\Lambda, \tag{3.40}$$

where

$$\boldsymbol{n}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}) = \begin{bmatrix} \dfrac{d}{dt}\left(\dfrac{\partial \mathcal{T}}{\partial \dot{\theta}_1}\right) \\ \dfrac{d}{dt}\left(\dfrac{\partial \mathcal{T}}{\partial \dot{\theta}_2}\right) \\ \dfrac{d}{dt}\left(\dfrac{\partial \mathcal{T}}{\partial \dot{\theta}_3}\right) \end{bmatrix}, \boldsymbol{c}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \begin{bmatrix} -\dfrac{\partial \mathcal{T}}{\partial \theta_1} \\ -\dfrac{\partial \mathcal{T}}{\partial \theta_2} \\ -\dfrac{\partial \mathcal{T}}{\partial \theta_3} \end{bmatrix}, \boldsymbol{g}(\boldsymbol{q}) = \begin{bmatrix} \dfrac{\partial \mathcal{V}}{\partial \theta_1} \\ \dfrac{\partial \mathcal{V}}{\partial \theta_2} \\ \dfrac{\partial \mathcal{V}}{\partial \theta_3} \end{bmatrix} \boldsymbol{\tau} = \begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \end{bmatrix},$$

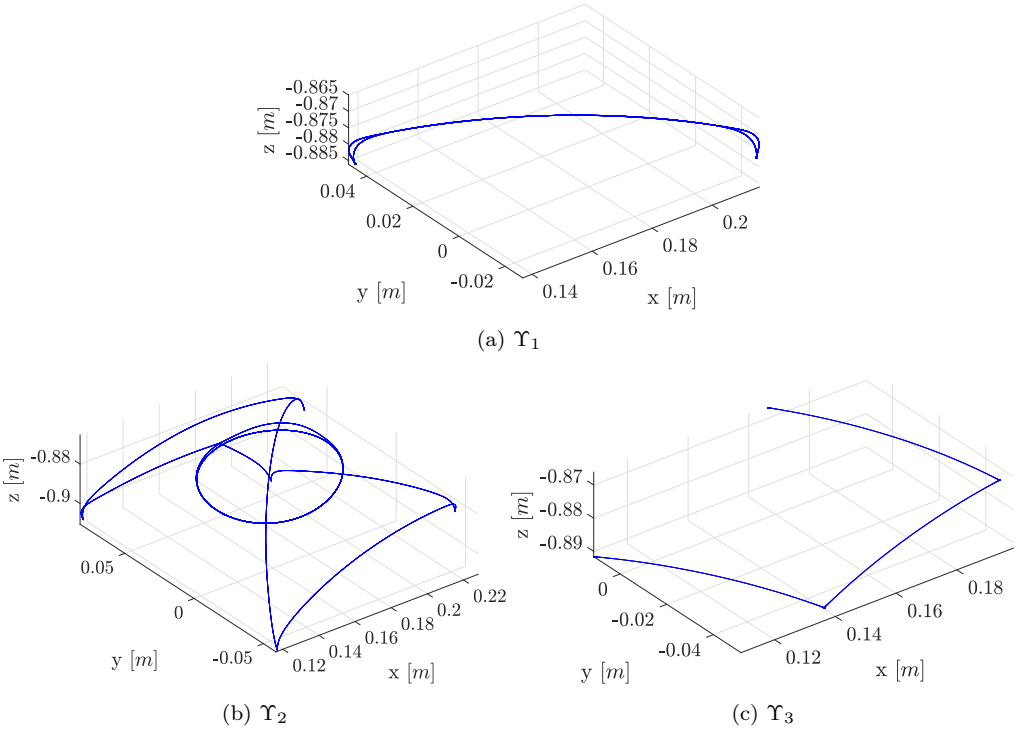
and the $\boldsymbol{K}(\boldsymbol{q})$ matrix, that takes into account the effect of holonomic constraint.

3.4.4 Experimental Validation

We compare the accuracy of the precise model $\mathcal{M}_f$ derived in Section 3.4.3 and the simplified model $\mathcal{M}_s$ recalled in Section 3.3.1, by using the data collected as explained in the previous section. Moreover, we provide a complexity analysis to evaluate the trade-off between accuracy and computational complexity.

The command torques and joint positions θ_i are sampled at a frequency of 2 kHz. The joint velocities $\dot{\theta}_i$ and accelerations $\ddot{\theta}_i$ are estimated with a Kalman smoother and the corresponding Cartesian velocities $\dot{x}, \dot{y}, \dot{z}$ and accelerations $\ddot{x}, \ddot{y}, \ddot{z}$ are computed via the robot Jacobian matrix. To ensure the highest possible accuracy of the estimated torques, the mass values m_i of each component of the robot have been measured with a precision balance, while the length l_i of each link and the position of their respective centers of mass a_i come from their CAD models.

The accuracy of the model is evaluated offline by comparing the real command torques obtained from the D3-1200 Delta Robot with the estimations computed with the mathematical models $\mathcal{M}_f$ and $\mathcal{M}_s$.



(a) Υ_1

(b) Υ_2

(c) Υ_3

Figure 3.12: Experimental validation of the models accuracy on three trajectories. (a) The trajectory Υ_1 is a pick-and-place profile; (b) the trajectory Υ_2 involves high and low accelerations in the center of the workspace; (c) the trajectory Υ_3 moves the robot to its workspace limits.

We consider three trajectories:

- Υ_1 is a pick-and-place trajectory over a belt conveyor, without payload (Figure 3.12a);
- Υ_2 consists of the end-effector motion in the middle of the workspace with high and low accelerations (Figure 3.12b): this trajectory is meant to evaluate the impact of the inertia and Coriolis/centrifugal terms;
- Υ_3 is a trajectory that moves the robot and then stops in three target points nearby its workspace limits (Figure 3.12c): this trajectory is meant to test the model in “extreme" poses.

Υ_1 aims to evaluate the models in the most frequent task of the robot: it consists in rapid movements between two belt conveyors combined with a vertical movement to pick the object.

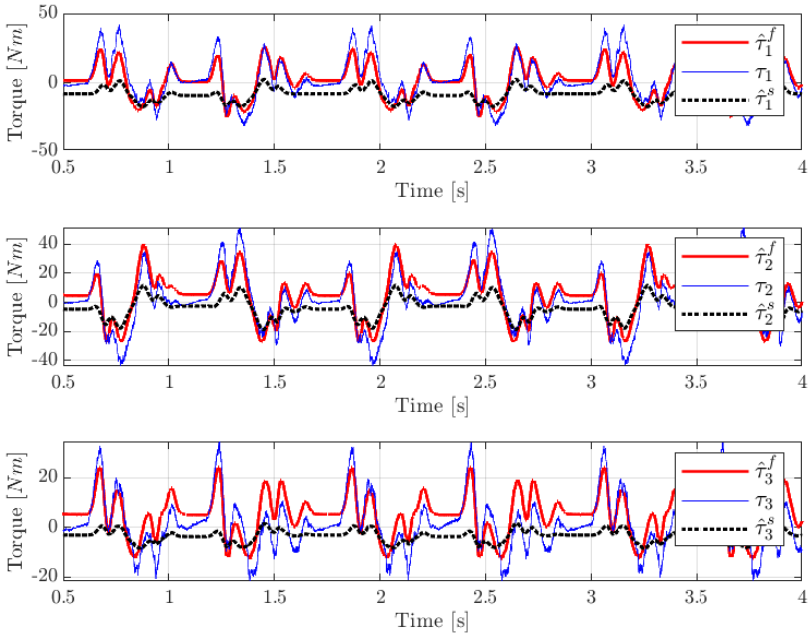


Figure 3.13: Real Torques (blue line) τ_i compared with estimated torques $\hat{\tau}_i^f$ computed by $\mathcal{M}_f$ (red line) and the estimated torques $\hat{\tau}_i^s$ computed by $\mathcal{M}_s$ (dotted line), for each actuated joint using the trajectory Υ_1 showed in Figure 3.12a.

Table 3.5: The means and standard deviations of models of the error for $\mathcal{M}_f$ and $\mathcal{M}_s$ over trajectory Υ_1.

Υ_1	Mean $[Nm]$			Std $[Nm]$		
	$\tilde{\tau}_1$	$\tilde{\tau}_2$	$\tilde{\tau}_3$	$\tilde{\tau}_1$	$\tilde{\tau}_2$	$\tilde{\tau}_3$
$\mathcal{M}_f$	0.63	-2.99	-3.77	6.42	7.79	6.61
$\mathcal{M}_s$	10.17	5.74	4.85	11.66	13.23	9.51

From the torque profiles in Figure 3.13, it is possible to notice how the simplified model $\mathcal{M}_s$ underestimates the control torque profiles. Moreover, during a pick action, i.e., when the accelerations are small, the simplified model predicts an almost constant negative offset. The precise model $\mathcal{M}_f$ has a much more accurate estimation of the command torques. When the robot moves the end-effector from one conveyor belt to the other, i.e, when the trajectory has the highest dynamics, the torque peaks of $\mathcal{M}_f$ are close to the real ones.

To evaluate the precision of each model, we consider the errors $\tilde{\tau}_i^f$, $\tilde{\tau}_i^s$ between the estimates $\hat{\tau}_i^f$, $\hat{\tau}_i^s$ for $\mathcal{M}_f$ and $\mathcal{M}_s$ and the real commands τ_i

$$\tilde{\tau}_i^f = \tau_i - \hat{\tau}_i^f, \quad \tilde{\tau}_i^s = \tau_i - \hat{\tau}_i^s.$$

Figure 3.16a and Table 3.5 report the mean and the standard deviation of the torque errors. The bar-plot allows to notice the reduction of the error of $\mathcal{M}_f$ with respect to the model $\mathcal{M}_s$.

The trajectory Υ_2 is meant to evaluate the behavior of the model when subjected to high and low accelerations in the middle of the robot workspace. In particular, this trajectory is similar to pick-and-place operations which took place between 5 to 10 centimeters apart from the center of the workspace as shown in Figure 3.12b. The torque profiles in Figure 3.14 show the improvements of $\mathcal{M}_f$ with respect to $\mathcal{M}_s$, especially during motion generated by high torque values.

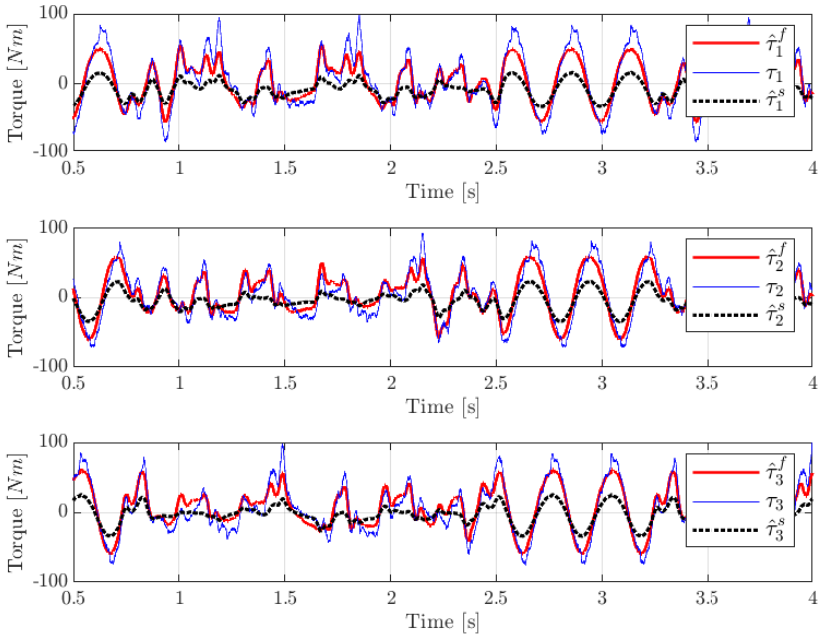


Figure 3.14: Real Torques (blue line) τ_i compared with estimated torques $\hat{\tau}_i^f$ computed by $\mathcal{M}_f$ (red line) and the estimated torques $\hat{\tau}_i^s$ computed by $\mathcal{M}_s$ (dotted line), for each actuated joint using the trajectory Υ_2 showed in Figure 3.12b.

Table 3.6: The means and standard deviations of the errors for models $\mathcal{M}_f$ and $\mathcal{M}_s$ over trajectory Υ_2.

Υ_2	Mean $[Nm]$			Std $[Nm]$		
	$\tilde{\tau}_1$	$\tilde{\tau}_2$	$\tilde{\tau}_3$	$\tilde{\tau}_1$	$\tilde{\tau}_2$	$\tilde{\tau}_3$
$\mathcal{M}_f$	2.41	-1.43	-2.18	14.27	12.62	11.60
$\mathcal{M}_s$	12.13	7.25	6.02	27.72	23.74	23.87

When the robot operates around the center of the workspace, the model $\mathcal{M}_s$ performs better than in the previous experiment: indeed, it predicts with higher accuracy when the accelerations are high with fast changes of directions (i.e., between $2.5s$ and $3.25s$), and the profiles are quite close to the real ones. On the other hand, it underestimates the torques during the pick-and-place

motion (i.e., between $0.75s$ and $2.25s$). The model $\mathcal{M}_f$ provides a better estimation of the torques over the whole duration of the trajectory. Figure 3.16b and Table 3.6 show the mean and the standard deviation of the estimation errors.

Υ_3 aims to evaluate accuracy when the robot motion alternates high and slow dynamics. The trajectory consists of high accelerations toward poses on the boundary of the workspace, followed by constant poses for a few seconds. The trajectory allows to test the accuracy of the model outside the usual working area.

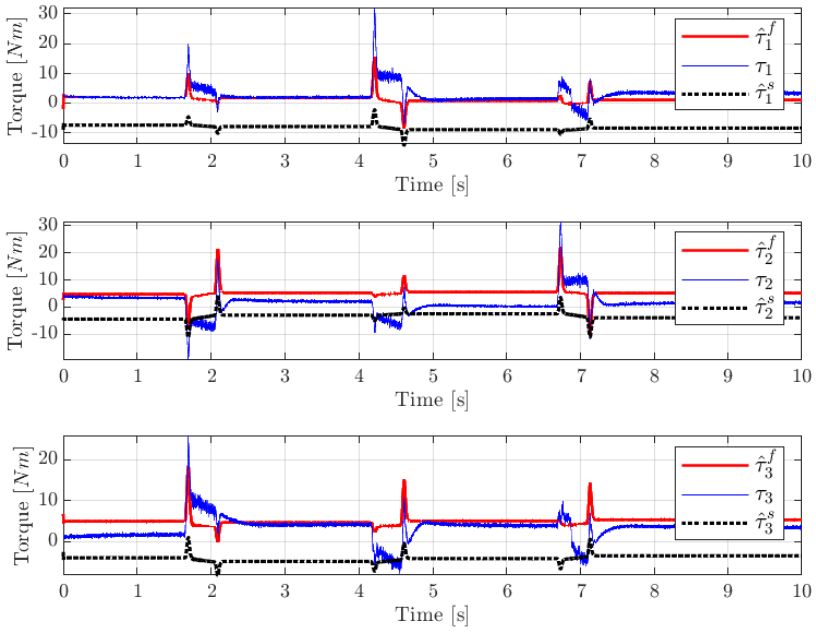
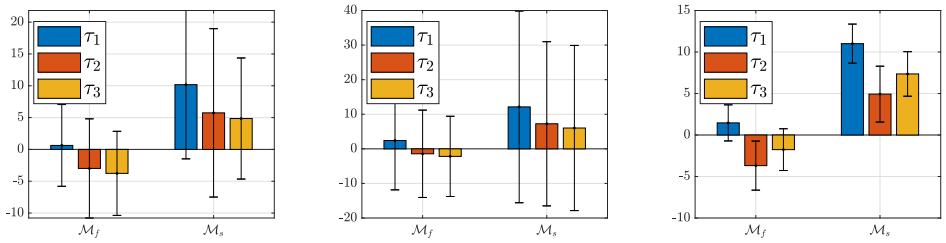


Figure 3.15: Real Torques (blue line) τ_i compared with estimated torques $\hat{\tau}_i^f$ computed by $\mathcal{M}_f$ (red line) and the estimated torques $\hat{\tau}_i^s$ computed by $\mathcal{M}_s$ (dotted line), for each actuated joint using the trajectory Υ_3 showed in Figure 3.12c.

Table 3.7: The means and standard deviations of the errors for models $\mathcal{M}_f$ and $\mathcal{M}_s$ over trajectory Υ_3.

Υ_3	Mean $[Nm]$			Std $[Nm]$		
	$\tilde{\tau}_1$	$\tilde{\tau}_2$	$\tilde{\tau}_3$	$\tilde{\tau}_1$	$\tilde{\tau}_2$	$\tilde{\tau}_3$
$\mathcal{M}_f$	1.46	-3.69	-1.76	2.17	2.95	2.52
$\mathcal{M}_s$	11.01	4.93	7.36	2.36	3.37	2.68

Figure 3.15 shows that the model $\mathcal{M}_s$ has a significantly large error when the robot is near the workspace limit. The torque peaks during the motion between the target points are not correctly computed and most of the time are underestimated. $\mathcal{M}_f$ is more accurate, both during static and dynamic parts. Means and standard deviations of the errors are shown in Figure 3.16c and listed in Table 3.7.



(a) (b) (c)

Figure 3.16: Mean errors with standard deviation bars of the errors for model $\mathcal{M}_f$ and $\mathcal{M}_s$ using the trajectory Υ_i.

The remaining errors between the command torques calculated by $\mathcal{M}_f$ and the real torques applied to the D3-1200 Delta Robot are due to different phenomena that are not taken into consideration in our model in order not to make it even complex. The main terms not modelled are: joints flexibility, links flexibility, friction at the joints, non-linearity of the gear-boxes, dynamic model of the electric motors. Moreover, it is worth mentioning that in our model we used the values for the parameters given by the CAD models. It is possible that such values are not precise.

3.4.5 Computational Complexity

The study of the computational complexity of the two models $\mathcal{M}_f$ and $\mathcal{M}_s$ addressed in this section is evaluated in terms of time complexity in the worst case scenario, which involves the usage of processors which do not have integrated ALU and the usage of not optimised algorithms for the computation of the binary operations. The time complexity is computed by multiplying the number of basic operations with the complexity of the operations. Given two n-digit numbers, the complexity for the addition and subtraction is $O(n)$ using the addition with carry and subtraction with borrow algorithms. The complexity for multiplication and division, using standard algorithms is $O(n^2)$. Trigonometric functions as sine and cosine are constructed by composing arithmetic functions; in particular exploiting the Taylor series, with repeated argument reduction and direct summation, we obtain a time complexity of $O(n^{5/2})$. The number of executed scalar operations, counted from the C++ implementation of both models, is reported in Table 3.8. According to the table the time complexity of $\mathcal{M}_s$ is

$$O(77n + 85n^2 + 16n^{5/2}),$$

while the time complexity of $\mathcal{M}_f$ is

$$O(1176n + 4045n^2 + 27n^{5/2}).$$

Thus, both models can be computed in polynomial time.

As expected the coefficients that multiply n and n^2 for $\mathcal{M}_f$ are larger than the coefficients for $\mathcal{M}_s$. We performed an analysis on the maximum and average execution frequency for both models in order to experimentally

Table 3.8: Number of operations executed in the full and simplified inverse dynamic model.

	Number of operations					
	$+$	$-$	$\times$	$\div$	sin	cos
$\mathcal{M}_s$	50	27	79	6	7	9
$\mathcal{M}_f$	669	477	2029	2016	12	15

Table 3.9: The average and maximum frequency obtained by computing the torque profile over a large set of trajectory for the models $\mathcal{M}_s$ ans $\mathcal{M}_f$.

	Avg frequency $[kHz]$	Max frequency $[kHz]$
$\mathcal{M}_s$	32.10	35.97
$\mathcal{M}_f$	4.65	5.02

evaluate the impact of the coefficients for $\mathcal{M}_s$ and $\mathcal{M}_f$ on the computational burden. The computer has a single-core CPU operating at 2.5 GHz. We calculated the torques with both models, over a set of different trajectories, as reported in Table 3.9. It turns out that $\mathcal{M}_f$ has a negative speed-up of 0.1449 on the average frequency. It still has to be considered that $\mathcal{M}_f$ has an average control loop that reaches frequencies higher than 4 kHz, which is still a high computation rate for modern controllers. Thus, despite the negative speed-up, $\mathcal{M}_f$ is totally suitable to be used within controllers of Delta Robots.

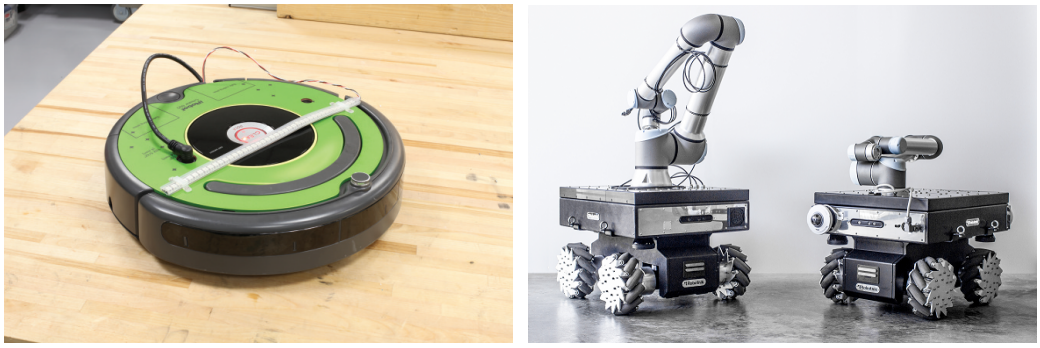
Concluding, a mathematically rigorous inverse dynamic model for industrial Delta Robots, obtained by adopting the Euler-Lagrange approach, has been presented. The model has been validated through a direct comparison with real torque profiles of an industrial Delta Robot. The proposed model has been tested and verified over a set of trajectories which covers the most common and stressful tasks, and proved to be more accurate than state-of-art counterparts. The increased computational complexity has been analyzed and confirmed to be larger than state-of-art counterparts, but it does not compromise the implementation in real-time industrial controllers. The model does not need any parameter identification, making it ready for direct use into a robot controller, and it is adaptable to any 3-DoF Delta Robot of which the kinematic and dynamic parameters are known.

Chapter 4

Stochastic Modeling for Mobile Robots

4.1 Related Works

Mobile robots are autonomous vehicles that have become increasingly popular in the last decades. Wheeled mobile robots (WMRs) of cart-like type, also known as differential-drive wheeled mobile robot, are widely used to accomplish a number of heterogeneous tasks such as transportation of objects in automated warehouses [95], moving platforms for manipulators, automated surveillance systems, autonomous house cleaning, garden care and so forth. Figure 4.1a shows a programmable iRobot Create 2, that is often used as base for autonomous vacuum cleaners and Figure 4.1b shows two RB-Kairos+ mobile manipulators, that have the peculiarity of being able to move laterally keeping unchanged their orientation thanks to a particular type of wheels called *Mecanum wheels*, also known as *Swedish wheels*.



(a) (b)

Figure 4.1: A programmable iRobot Create 2 by `https://www.irobot.com/` (a) and two RB-Kairos+ mobile omnidirectional manipulators by Robotnik `https://robotnik.eu/` (b).

This chapter is based on the following:
▷ F.Vesentini, L. Di Persio, R. Muradore. *A Brownian-Markov Stochastic Model for Cart-Like Wheeled Mobile Robots*, (under review).

A reliable dynamic model is important in real-world applications to properly address motion simulation, gray-box model identification, path planning, motion controller design and similar.

The dynamic models that are most commonly considered in literature [22, 23, 36, 50, 89, 93, 110, 137] are derived by exploiting Lagrangian-based methodologies or approaches based on the application of the Newton-Euler principle on the force diagram of the robots. They assume the non-holonomic constraint of no-lateral motion. This assumption leads to a linear and relatively simple to handle dynamic model, with two control inputs: forward and angular velocities. At the same time it has the disadvantage of not being able to explain unexpected behaviours such as losses of lateral grip while turning around a corner at high speed. In other words, it does not consider the action of inertial forces on the robot whenever the angular velocity is non-zero. Furthermore, their deterministic nature does not allow to take into account uncertainty characterizing unknown external forces acting on the WMR and model inaccuracy, such as inertial forces and friction.

In this chapter, we present a novel dynamic model for cart-like WMRs able to take into account lateral motion by relaxing the aforementioned non-holonomic constraint and the contribution of random dissipative or inertial external forces. In particular, we start from a the deterministic model which consists of a two-state hybrid system [54]: the *grip state*, determined by a set of five ordinary differential equations (ODEs) that does not allow lateral motion and the *slip state*, governed by a set of six differential equations (the additional one is due a new variable corresponding to the lateral velocity) obtained by relaxing the non-holonomic constraint of no-lateral motion [85]. Afterwards, following a procedure similar to the one adopted in [138], we transform the ODEs characterizing both system states into the corresponding set of Stochastic Differential Equations (SDEs) with possibly non-linear drift and diffusion terms.

In particular, such SDEs are characterized by

- independent Brownian motions modeling the uncertainty about dissipative external forces acting on the vehicle (e.g., the rolling friction),
- a two-state Markov chain, regulating the transition from the grip state to thr slip state of the stochastic model and viceversa.

We prove existence and uniqueness of solutions [6, 13, 28, 29, 37, 78, 88, 98] to our novel hybrid stochastic Brownian-Markov dynamic model (HSBM) by verifying that the sufficient conditions on drift and diffusion terms are satisfied. Moreover, via numerical simulations we show that it is able to model both nominal and non-conventional behaviors, e.g., lateral grip phenomenon.

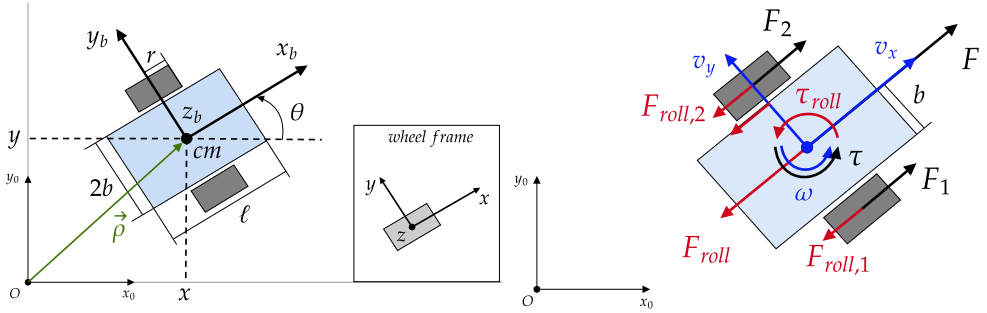
4.2 Deterministic Dynamics

The deterministic dynamic model from which we start before deriving the corresponding stochastic model is a hybrid system with two states of differential

equations, which represent the states of motion "*grip*" and "*slip*", respectively. In particular, every state is characterized by a precise set of ODEs obtained by considering two different motion constraints and by exploiting two different but equivalent methodologies: the *Newton-Euler Laws of Motion* and the *Lie Groups Theory*.

4.2.1 Robot Diagram

Let consider a fixed *spatial frame* $\Sigma_0 = (O, x_0, y_0, z_0)$, a moving *body frame* $\Sigma_b = (O_b, x_b, y_b, z_b)$ such that the z_b and z_0 axes are aligned, and a *wheel frame* $\Sigma_w = (x, y, z)$, as shown in Figure 4.2. The configuration of the robot is given by coordinates $q = (x, y, \theta)^T$, where (x, y) is the position and θ is the orientation of the body frame in the spatial frame. Vector $\vec{\rho}$ is the position of the robot center of mass in the spatial frame. The x-axis of the wheel frame is the forward direction of motion, the y-axis is the rolling axis of the wheel, while the z-axis is aligned with z_0 and z_b.



(a) (b)

Figure 4.2: (a) Robot spatial and body frames, and wheel frame; (b) force diagram of the robot describing all the forces and torques acting on the WMR. The control forces are depicted in black, the dissipative and uncontrolled forces are in red and the linear and angular velocities in blue.

The WMR is modeled as a rectangular rigid body moving along x_b and rotating on the plane around z_b. Coordinates for velocities with respect to the body frame, i.e. the body velocities, are denoted by $v = (v_x, v_y, \omega)^T$, where v_x is the *forward velocity* along x_b, v_y is the *lateral velocity* along y_b and ω is the *angular velocity* about z_b.

Table 4.1 summarizes the dynamic parameters of the WMR. The total mass of WMR is $m = m_r + 2m_w$ and since the robot is of rectangular shape and wheels are modeled as solid and thin disks, the total inertia I is given by

$$I = I_r + 2I_{wz} = \frac{m_r(4b^2 + \ell^2) + 6m_w r^2}{12}, \tag{4.1}$$

where

$$I_r = \frac{m_r(4b^2 + \ell^2)}{12} \quad \text{and} \quad I_{wz} = \frac{m_w r^2}{4}$$

Table 4.1: WMR dynamic parameters.

parameter	description
ℓ	length of the robot
r	radius of a wheel
b	length of the semi-axle
CoM	robot center of mass
m_r	mass of the robot (no wheels)
m_w	mass of a wheel
m	total mass of the robot,
I_r	inertia about z_b-axis (no wheels)
I_{wz}	inertia of a wheel about z-axis
I	inertia of the robot about z_b-axis

are the robot inertia without considering wheels and the inertia of a single wheel about z-axis, respectively. The velocity with respect to frame Σ_0, $\dot{q} = (\dot{x}, \dot{y}, \dot{\theta})^T$, is obtained from the body velocity v via the following equations

$$\dot{x} = v_x \cos\theta - v_y \sin\theta, \tag{4.2a}$$

$$\dot{y} = v_x \sin\theta + v_y \cos\theta, \tag{4.2b}$$

$$\dot{\theta} = \omega. \tag{4.2c}$$

The vector $\boldsymbol{x} = (q, v)^T = (x, y, \theta, v_x, v_y, \omega)^T$ represents the *state of the WMR*. As depicted in Figure 4.2(b), we assume that the controls of the robot are the linear force $F \triangleq F_1 + F_2$ and the control torque $\tau \triangleq (F_1 - F_2)b$, where F_1 and F_2 are the traction forces at the wheels.

4.2.2 External Forces

The external forces acting on the robot are shown in Figure 4.2(b) and Figure 4.3. The rolling resistance F_{roll} is the total force opposing the control force F, defined by

$$F_{roll} = F_{roll,1} + F_{roll,2} = \text{sign}(v_x) c_{rr} m g, \tag{4.3}$$

where g is the gravity acceleration and $c_{rr} > 0$ is the *rolling resistance coefficient*. The value of c_{rr} depends on the material of ground and wheels. τ_{roll} is the total friction momentum opposing the control torque τ, given by

$$\tau_{roll} = (F_{roll,1} - F_{roll,2})b. \tag{4.4}$$

The centrifugal force F_{cf} acting on the WMR when it moves along non-straight trajectories, is defined by

$$F_{cf} = m\omega^2 R = m v_x \omega,$$

where $R = v_x/\omega$ is the curvature of the trajectory. The centripetal force F_{cp} opposing the centrifugal force, is generated by the static lateral friction force and is given by

$$F_{cp} = \mu_s m g, \tag{4.5}$$

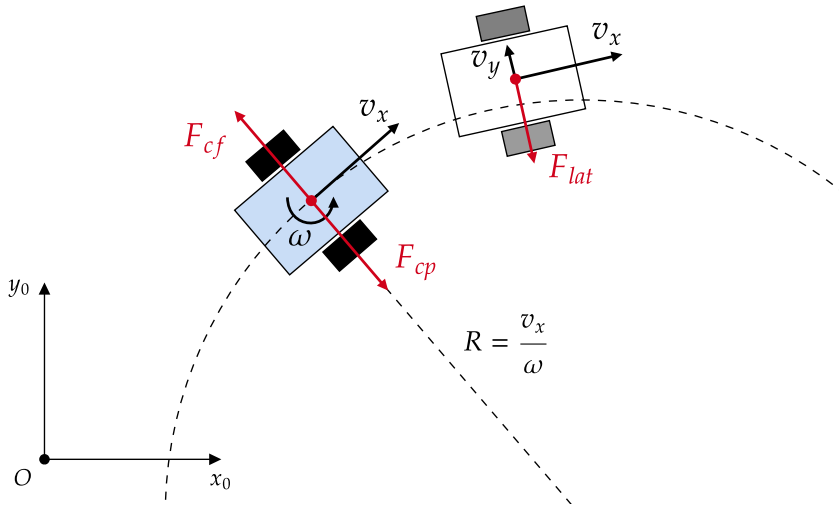


Figure 4.3: Representation of the centrifugal force F_{cf}, the centripetal force F_{cp} and lateral friction force F_{lat} acting on the robot.

where μ_s is the static friction coefficient.

The lateral friction force F_{lat} acting on the robot during lateral slip, is the same as equation (4.5) where μ_s is replaced with the kinetic friction coefficient μ_k, such that $\mu_k < \mu_s$. Referring to Figure 4.3, as long as the balance between the F_{cf} and F_{cp} holds, the robot has perfect grip with the ground and stays on track (blue robot). If the balance is broken the robot has non-zero orthogonal velocity v_y, subjected to the action of lateral friction force F_{lat} (white robot).

4.2.3 Equations of Motion

We describe the state variables, the dynamic parameters and the system of ODEs governing the dynamics of a WMR. We derive the equations of motion by adopting the Newton-Euler Approach, based on the well known *Euler's Laws of Motion*, see [60, 77, 102], extension of Newton's Laws of Motion for point particles to rigid bodies; and also by applying the *Lie Groups Theory*, see [17, 22, 28, 29, 61].

Newton-Euler approach. Consider the free-body diagram of the robot in Figure 4.2 (a) and representing the position vector $\vec{\rho} = (x,\, y)^T$ of the body frame Σ_b in polar coordinates

$$\begin{cases} x &= \rho\cos\theta \\ y &= \rho\sin\theta \end{cases} \qquad \rho \geq 0 \quad \theta \in [0, 2\pi). \tag{4.6}$$

A double differentiation of equations (4.6) with respect to time, allows to find the expressions for the spatial velocity and acceleration of the body frame: by letting $\dot{\theta} = \omega$, the components of velocity and acceleration of the robot frame

Σ_b in the spatial frame Σ_0 are given by

$$\begin{aligned}
\dot{x} &= \dot{\rho}\cos\theta - \rho\omega\sin\theta, \\
\ddot{x} &= \left(\ddot{\rho} - \rho\omega^2\right)\cos\theta - (2\dot{\rho}\omega + \rho\dot{\omega})\sin\theta, \\
\dot{y} &= \dot{\rho}\sin\theta + \rho\omega\cos\theta, \\
\ddot{y} &= \left(\ddot{\rho} - \rho\omega^2\right)\sin\theta + (2\dot{\rho}\omega + \rho\dot{\omega})\cos\theta.
\end{aligned}$$

Such equations can be re-arranged into a sum of unitary vectors that identify a radial direction $\vec{r}$ and a tangential direction $\vec{t}$ with respect to Σ_0 as follows

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \dot{\rho}\,\vec{r} + \rho\omega\,\vec{t}, \tag{4.7}$$

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \end{bmatrix} = \left(\ddot{\rho} - \rho\omega^2\right)\,\vec{r} + (2\dot{\rho}\omega + \rho\dot{\omega})\,\vec{t}, \tag{4.8}$$

where $\vec{r} = (\cos\theta, \sin\theta)^T$ and $\vec{t} = (-\sin\theta, \cos\theta)^T$. From equations (4.8) we deduce $\dot{\rho} = v_x$, $\rho\omega = v_y$, $\ddot{\rho} - \rho\omega^2 = a_x$ and $2\dot{\rho}\omega + \rho\dot{\omega} = a_y$, that in turn allow to write the components of the acceleration vector, a_x and a_y, as follows

$$a_x = \dot{v}_x - v_y\omega, \qquad a_y = \dot{v}_y + v_x\omega, \tag{4.9}$$

with $a_{cm} = (a_x, a_y)^T$ the acceleration of the centre of mass. We now apply the Euler's Laws of Motion [60, 77, 102] to obtain the following equations

$$\begin{aligned}
m\,a_x &= F - F_{roll}, \\
m\,a_y &= F_{lat}, \\
I\dot{\omega} &= \tau - \tau_{roll},
\end{aligned}$$

from which, thanks to equations (4.9) and the kinematics (4.2a), we can derive the equations of motion

$$\dot{x} = v_x\cos\theta - v_y\sin\theta, \tag{4.10a}$$

$$\dot{y} = v_x\sin\theta + v_y\cos\theta, \tag{4.10b}$$

$$\dot{\theta} = \omega, \tag{4.10c}$$

$$\dot{v}_x = v_y\omega + \frac{1}{m}(F - F_{roll}), \tag{4.10d}$$

$$\dot{v}_y = -v_x\omega + \frac{1}{m}F_{lat}, \tag{4.10e}$$

$$\dot{\omega} = \frac{1}{I}(\tau - \tau_{roll}). \tag{4.10f}$$

The directions of forward velocity v_x and lateral velocity v_y are represented in Figure 4.2(b) and external forces are described in Section 4.2.2.

Lie Groups approach. We now show that equations (4.10d) - (4.10f) can be derived using the concepts reported in Section A.1. We begin the discussion by stating that the configuration g and the velocity ξ of the robot

can be expressed as elements $g \in SE(2)$ and $\xi \in \mathfrak{se}(2)$, respectively, as follows

$$g = \begin{bmatrix} R & \rho \\ 0_{1\times 2} & 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & x \\ \sin\theta & \cos\theta & y \\ 0 & 0 & 1 \end{bmatrix}, \tag{4.11a}$$

$$\xi = \begin{bmatrix} \Omega & v \\ 0_{1\times 2} & 0 \end{bmatrix} = \begin{bmatrix} 0 & -\omega & v_x \\ \omega & 0 & v_y \\ 0 & 0 & 0 \end{bmatrix}, \tag{4.11b}$$

where $\rho = (x, y)^T$, $R \in SO(2)$, $v = (v_x, v_y)^T$ and $\Omega = R^T\dot{R} \in \mathfrak{se}(2)$ is the body angular velocity. The Matrix Lie group $SE(2)$ is the *configuration manifold*, also know as *configuration space* [110] of the robot and $\mathfrak{se}(2)$ is the corresponding Matrix Lie algebra. The state of the robot (g, ξ) can be seen as an element of $SE(2) \times \mathfrak{se}(2)$, see Appendix A.2, Chapter 5 of [22] and [64].

The *Kinetic Energy* of the robot is

$$T = \frac{1}{2}m(\dot{x}^2 + \dot{y}^2) + \frac{1}{2}I\dot{\theta}^2, \tag{4.12}$$

and Lemma 4.30 of [22] allows to derive the *kinetic energy metric*

$$\mathbb{G} = Ie_1^\vee \otimes e_1^\vee + m(e_2^\vee \otimes e_2^\vee + e_3^\vee \otimes e_3^\vee),$$

where $\otimes$ is the outer product of vectors and $e_i^\vee$ is the vector representation of the elements $\{e_1, e_2, e_3\}$ of the basis for $\mathfrak{se}(2)$, through the "vee map" $.^\vee$ defined in (A.7), as in Example A.2.0.2.

$\mathbb{G}$ is a Riemannian metric on $SE(2)$, i.e. a (0,2)-tensor whose matrix representation $[\mathbb{G}]$ is given

$$[\mathbb{G}] = \begin{bmatrix} I & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & m \end{bmatrix}, \tag{4.13}$$

therefore the couple $(SE(2), \mathbb{G})$ is a Riemannian manifold. We denote $[\mathbb{G}] \triangleq \mathbb{I}$ from now on, since such matrix representation corresponds to the inertia tensor of the robot, see Chapter 4 of [22].

The elements defined by (4.11a) and (4.11b), together with Theorem 5.45 of [22], allow us to write the dynamic equations of the robot in compact form as

$$\dot{g} = g\xi, \tag{4.14a}$$

$$\dot{\xi}^\vee = \mathbb{I}^\sharp[\mathrm{ad}_\xi^*]\mathbb{I}^\flat\xi^\vee + \left(\frac{f_1}{I}, \frac{f_2}{m}, \frac{f_3}{m}\right)^T \tag{4.14b}$$

where $\mathbb{I}^\sharp : \mathfrak{se}(2)^* \to \mathfrak{se}(2)$ and $\mathbb{I}^\flat : \mathfrak{se}(2) \to \mathfrak{se}(2)^*$ are the *sharp map* and the *flat map* associated to the inner product induced by $\mathbb{I}$, respectively, and $[\mathrm{ad}_\xi^*]$ is the matrix representation of co-adjoint operator as in Definition A.2.0.7.

The sharp and the flat map are one the inverse of the other and in our case, thanks to the isomorphisms with $\mathbb{R}^3$ we have, $\mathbb{I}^\flat = \mathbb{I}$, $\mathbb{I}^\sharp = \mathbb{I}^{-1}$ and $[\mathrm{ad}^*_\xi] = [\mathrm{ad}_\xi]^T$. The terms f_i incorporate the external forces acting on the system, namely

$$f_1 = \frac{1}{I}\left(\tau - \tau_{roll}\right), \quad f_2 = \frac{1}{m}\left(F - F_{roll}\right), \quad f_3 = \frac{1}{m}F_{lat}.$$

Equation (4.14a) yields

$$\begin{bmatrix} -\dot\theta\sin\theta & \dot\theta\cos\theta & \dot x \\ \dot\theta\cos\theta & -\dot\theta\sin\theta & \dot y \\ 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} -\omega\sin\theta & \omega\cos\theta & v_x\cos\theta - v_y\sin\theta \\ \omega\cos\theta & -\omega\sin\theta & v_x\sin\theta + v_y\cos\theta \\ 0 & 0 & 0 \end{bmatrix},$$

where a term-by-term comparison of the coefficients of the two matrices above allows to deduce the following equations

$$\begin{aligned} \dot\theta &= \omega, \\ \dot x &= v_x\cos\theta - v_y\sin\theta, \\ \dot y &= v_x\sin\theta + v_y\cos\theta. \end{aligned}$$

Considering equation (4.14b), we have

$$\begin{aligned} \mathbb{I}^\sharp[\mathrm{ad}^*_\xi]\mathbb{I}^\flat\xi^\vee &= \begin{bmatrix} 1/I & 0 & 0 \\ 0 & 1/m & 0 \\ 0 & 0 & 1/m \end{bmatrix}\begin{bmatrix} 0 & v_y & -v_x \\ 0 & 0 & \omega \\ 0 & -\omega & 0 \end{bmatrix}\begin{bmatrix} I & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & m \end{bmatrix}\begin{bmatrix} \omega \\ v_x \\ v_y \end{bmatrix} \\ &= \begin{bmatrix} 0 \\ v_y\omega \\ -v_x\omega \end{bmatrix}, \end{aligned}$$

and since $\dot\xi^\vee = (\dot\omega, \dot v_x, \dot v_y)^T$, we end up with

$$\begin{aligned} \dot\omega &= \frac{1}{I}(\tau - \tau_{roll}), \\ \dot v_x &= v_y\omega + \frac{1}{m}(F - F_{roll}), \\ \dot v_y &= -v_x\omega + \frac{1}{m}F_{lat}. \end{aligned}$$

By rearranging and summarizing the previous equations, the robot dynamics is given by

$$\dot x = v_x\cos\theta - v_y\sin\theta, \tag{4.15a}$$
$$\dot y = v_x\sin\theta + v_y\cos\theta, \tag{4.15b}$$
$$\dot\theta = \omega, \tag{4.15c}$$
$$\dot v_x = v_y\omega + \frac{1}{m}(F - F_{roll}), \tag{4.15d}$$
$$\dot v_y = -v_x\omega + \frac{1}{m}F_{lat}, \tag{4.15e}$$
$$\dot\omega = \frac{1}{I}(\tau - \tau_{roll}). \tag{4.15f}$$

that is identical to equations (4.10a) - (4.10f).

4.2.4 Hybrid System

It is a common choice, see e.g. [36, 50, 73, 93, 110, 137], to derive the dynamic model differential-drive mobile robots by assuming the following motion constraint

$$\dot{y}\cos\theta - \dot{x}\sin\theta = 0. \tag{4.16}$$

The non-holonomic bilateral constraint 4.16 implies no lateral motion of the robot, see [85, 104], meaning that $v_y = 0$. The motion of a system subject to this constraint is governed only by the forward velocity v_x and the angular velocity ω. Well known mechanical systems whose equations of motion are derived by assuming such constraint are, for example, the *rolling disk* [22, 110], the *falling rolling disk* [18] and the *Chaplygin sleigh* [16, 19]. This assumption is usually a good approximation of the real behaviour, nevertheless it does not model lateral sliding due to unpredictable external forces acting on the system. To allow lateral motion, we assume a different bilateral contraint given by

$$\dot{y}\cos\theta - \dot{x}\sin\theta = \rho\omega \tag{4.17}$$

where ρ is the absolute value of $\vec{\rho}$ in Figure 4.2 (a). The quantity $\rho\omega$ is the lateral velocity v_y of the body frame Σ_b, see Section 4.2.3 and [36]. Mechanical systems whose equations of motion are derived by assuming a motion constraint like (4.17) are the *rimmed wheel with transverse grooves* [85] and the *planar slider* [25, 56, 57].

We describe the deterministic model for a differential-drive robot as a hybrid system with two-state [54], which are identified as *grip*, state and *slip* state, being denoted by $\mathcal{M}_g$ and $\mathcal{M}_s$, respectively. The system switches from $\mathcal{M}_g$ to $\mathcal{M}_s$ whenever the condition $|F_{cf}| \leq |F_{cp}|$ is violated, where F_{cf} is the centrifugal force and F_{cp} is the centripetal force due to the friction of the wheels the with ground. The system switches back from $\mathcal{M}_s$ to $\mathcal{M}_g$ once the lateral velocity v_y returned to zero due to the action of the lateral friction force.

The system in *grip state* consists of five differential equations, derived by assuming the constraint given by (4.16). The equations of motion are

$$\mathcal{M}_g : \begin{cases} \dot{x} = v_x\cos\theta & x(0) = x_0, & \text{(4.18a)} \\ \dot{y} = v_x\sin\theta & y(0) = y_0, & \text{(4.18b)} \\ \dot{\theta} = \omega & z(0) = z_0, & \text{(4.18c)} \\ \dot{v}_x = 1/m\,(F - F_{roll}) & v_x(0) = v_{x,0}, & \text{(4.18d)} \\ \dot{\omega} = 1/I\,(\tau - \tau_{roll}) & \omega(0) = \omega_0, & \text{(4.18e)} \end{cases}$$

where F is the control force, F_{roll} is the rolling friction force opposing F, τ is the control torque and τ_{roll} is the friction momentum opposing τ.

The system in *slip state* consists of six differential equations, derived by

assuming (4.17)

$$\mathcal{M}_s : \begin{cases} \dot{x} = v_x \cos\theta - v_y \sin\theta & x(t_\xi) = x_\xi, \quad (4.19a) \\ \dot{y} = v_x \sin\theta + v_y \cos\theta & y(t_\xi) = y_\xi, \quad (4.19b) \\ \dot{\theta} = \omega & z(t_\xi) = z_\xi, \quad (4.19c) \\ \dot{v}_x = v_y\omega + 1/m\,(F - F_{roll}) & v_x(t_\xi) = v_{x,\xi}, \quad (4.19d) \\ \dot{v}_y = -v_x\omega + 1/m\,F_{lat} & v_y(t_\xi) = v_{y,\xi}, \quad (4.19e) \\ \dot{\omega} = 1/I\,(\tau - \tau_{roll}) & \omega(t_\xi) = \omega_\xi, \quad (4.19f) \end{cases}$$

where t_ξ is the time instant when $|F_{cf}| > |F_{cp}|$ and F_{lat} is the lateral friction acting on the robot. Section 4.2.2 describes in detail all the inertial and friction forces acting on the system.

4.3 Brownian-Markov Stochastic Model for WMR

We now provide a stochastic model for WMRs by assuming to have

(a) *no knowledge about the external friction forces* acting on the robot, therefore replacing them with terms proportional to independent reflected Brownian motions, and

(b) *no knowledge about the action of the centrifugal force F_{cf} and lateral friction F_{lat}*, replacing the state transition conditions $|F_{cf}| \leq |F_{cp}|$ and $|v_y| > 0$ with transition probabilities characterizing a homogeneous Markov chain with two states.

The choice of a reflected Brownian motion, instead of a standard one, is motivated by the need of modeling the dissipative external forces acting on the system as stochastic processes starting from an initial state and whose paths are confined between lower and upper barriers. We assume a reflected Brownian motion also for modeling uncertainty about the critical value for $v_x\omega$, starting from an initial guess $v_{x,0}\omega_0$. Reflection means that whenever a path or realization of the Brownian motion hits a barrier at time t, its dynamics is reflected, hence pushed back, with the same angle of incidence.

The uncertainty about inertial and external forces turns the two-state hybrid system described by $\mathcal{M}_g$ and $\mathcal{M}_s$ into a two-state hybrid system of SDEs, described by $\overline{\mathcal{M}}_g$ and $\overline{\mathcal{M}}_s$, respectively. In Section 4.3.1 we explain how to mathematically derive the SDEs governing both states $\overline{\mathcal{M}}_g$ and $\overline{\mathcal{M}}_s$ of the stochastic model, in Section 4.3.2 we point out the hybrid formulation, then in Section 4.3.3 we prove both existence and uniqueness of the solutions.

4.3.1 Derivation of Stochastic Equations

Equations for the *grip state* (4.18a) - (4.18c) are re-arranged in matrix form as

$$\begin{bmatrix} dx \\ dy \\ d\theta \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} d\rho \\ d\theta \end{bmatrix}, \quad (4.20)$$

exploiting the fact that $v_x = d\rho/dt$ and $\omega = d\theta/dt$, where ρ is the absolute value of the position vector of Σ_b with respect to Σ_0. By adopting a procedure similar to [29, 138], we add uncertainty about position and orientation of the robot by re-defining $d\rho$ and $d\theta$ in (4.20) as

$$d\rho = v_x dt + \sigma_1 dW_1, \tag{4.21}$$

$$d\theta = \omega dt + \sigma_2 dW_2, \tag{4.22}$$

where $W_i = \{W_i(t) \,|\, t \geq 0\}$ for $i = 1, 2$ are two independent standard Brownian motions and σ_i are their diffusion coefficients. Substituting (4.21) and (4.22) into (4.20) we have

$$\begin{bmatrix} dx \\ dy \\ d\theta \end{bmatrix} = \begin{bmatrix} v_x \cos\theta \\ v_x \sin\theta \\ \omega \end{bmatrix} dt + \begin{bmatrix} \sigma_1 \cos\theta & 0 \\ \sigma_1 \sin\theta & 0 \\ 0 & \sigma_2 \end{bmatrix} \begin{bmatrix} dW_1 \\ dW_2 \end{bmatrix}, \tag{4.23}$$

which is a linear SDE with drift and diffusion coefficients represented by the vector-valued function $G_q^g : \mathbb{R}^3 \times \mathbb{R}^2 \to \mathbb{R}^3$ and the matrix-value function $H_q^g : \mathbb{R}^3 \times \mathbb{R}^2 \to \mathbb{R}^{3\times 2}$:

$$G_q^g(q, v) \triangleq \begin{bmatrix} v_x \cos\theta \\ v_x \sin\theta \\ \omega \end{bmatrix}, \tag{4.24}$$

$$H_q^g\left(q, \sigma_{\{1,2\}}\right) \triangleq \begin{bmatrix} \sigma_1 \cos\theta & 0 \\ \sigma_1 \sin\theta & 0 \\ 0 & \sigma_2 \end{bmatrix}. \tag{4.25}$$

Analogously, we derive the stochastic version of equations (4.18d) and (4.18e), by writing them as follows

$$\begin{bmatrix} dv_x \\ d\omega \end{bmatrix} = \begin{bmatrix} 1/m & 0 \\ 0 & 1/I \end{bmatrix} \begin{bmatrix} dp \\ Id\omega \end{bmatrix}, \tag{4.26}$$

and exploiting the fact that $F = dp/dt$ and $\tau = Id\omega/dt$, where p is the linear momentum of the robot. Then we redefine dp and $Id\omega$ as

$$dp = Fdt - \sigma_4 d\widetilde{W}_1, \tag{4.27}$$

$$Id\omega = \tau dt - \sigma_6 d\widetilde{W}_3, \tag{4.28}$$

where $\widetilde{W}_i = \left\{\widetilde{W}_i(t) \,|\, t \geq 0\right\}$ for $i = \{4, 6\}$ are two independent reflected Brownian motions and σ_i are their diffusion coefficients. Substituting the definitions (4.27) and (4.28) into the system (4.26) we get to

$$\begin{bmatrix} dv_x \\ d\omega \end{bmatrix} = \begin{bmatrix} F/m \\ \tau/I \end{bmatrix} dt - \begin{bmatrix} \sigma_4/m & 0 \\ 0 & \sigma_6/I \end{bmatrix} \begin{bmatrix} d\widetilde{W}_1 \\ d\widetilde{W}_3 \end{bmatrix}, \tag{4.29}$$

which is a linear SDE with drift and diffusion coefficients, represented by the vector-valued function $G_v^g : \mathbb{R}^2 \to \mathbb{R}^{2\times 1}$ and the matrix-valued function H_v^g :

$\mathbb{R}^2 \to \mathbb{R}^{2\times 2}$:

$$G_v^g(F,\tau) \triangleq \begin{bmatrix} F/m \\ \tau/I \end{bmatrix}, \tag{4.30}$$

$$H_v^g(\sigma_{\{4,6\}}) \triangleq \begin{bmatrix} \sigma_4/m & 0 \\ 0 & \sigma_6/I \end{bmatrix}. \tag{4.31}$$

We re-write the kinematics defined by *slip state* equations (4.19a) - (4.19c) in matrix form as

$$\begin{bmatrix} dx \\ dy \\ d\theta \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ \sin\theta & 0 & \cos\theta \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} d\rho \\ d\theta \\ \rho d\theta \end{bmatrix}, \tag{4.32}$$

exploiting the fact $v_x = \rho d\theta/dt$. Defining $\rho d\theta$ as

$$\rho d\theta = v_y dt + \sigma_3 dW_3, \tag{4.33}$$

where W_3 is a standard Brownian motion and σ_3 is a diffusion coefficient, and substituting equations (4.21), (4.22) and (4.33) into (4.32), we get

$$\begin{bmatrix} dx \\ dy \\ d\theta \end{bmatrix} = \begin{bmatrix} v_x\cos\theta - v_y\sin\theta \\ v_x\sin\theta + v_y\cos\theta \\ \omega \end{bmatrix} dt + \begin{bmatrix} \sigma_1\cos\theta & 0 & -\sigma_3\sin\theta \\ \sigma_1\sin\theta & 0 & \sigma_3\cos\theta \\ 0 & \sigma_2 & 0 \end{bmatrix} \begin{bmatrix} dW_1 \\ dW_2 \\ dW_3 \end{bmatrix} \tag{4.34}$$

where drift and diffusion terms are represented by the vector-value and the matrix-valued function $G_q^s : \mathbb{R}^3 \times \mathbb{R}^3 \to \mathbb{R}^3$ and $H_q^s : \mathbb{R}^3 \times \mathbb{R}^3 \to \mathbb{R}^{3\times 3}$ given by

$$G_q^s(q,v) \triangleq \begin{bmatrix} v_x\cos\theta - v_y\sin\theta \\ v_x\sin\theta + v_y\cos\theta \\ \omega \end{bmatrix}, \tag{4.35}$$

$$H_q^s(q,\sigma_{\{1,2,3\}}) \triangleq \begin{bmatrix} \sigma_1\cos\theta & 0 & -\sigma_3\sin\theta \\ \sigma_1\sin\theta & 0 & \sigma_3\cos\theta \\ 0 & \sigma_2 & 0 \end{bmatrix}. \tag{4.36}$$

We re-write the dynamics defined by *slip state* equations (4.19d)-(4.19f) in matrix form as

$$\begin{bmatrix} dv_x \\ dv_y \\ d\omega \end{bmatrix} = \begin{bmatrix} v_y\omega \\ -v_x\omega \\ 0 \end{bmatrix} dt + \begin{bmatrix} 1/m & 0 \\ 0 & 0 \\ 0 & 1/I \end{bmatrix} \begin{bmatrix} dp \\ I d\omega \end{bmatrix}. \tag{4.37}$$

We apply the same argument by assuming $dv_y = -v_x\omega dt - \sigma_5 d\widetilde{W}_2$ and by substituting it into (4.37) together with the definitions (4.27) and (4.28), obtaining

$$\begin{bmatrix} dx \\ dy \\ d\theta \end{bmatrix} = G_v^s(v,F,\tau)\,dt - H_v^s\left(\sigma_{\{4,5,6\}}\right) \begin{bmatrix} d\widetilde{W}_1 \\ d\widetilde{W}_2 \\ d\widetilde{W}_3 \end{bmatrix}, \tag{4.38}$$

where $G_v^s : \mathbb{R}^3 \times \mathbb{R}^2 \to \mathbb{R}^3$ and $H_v^s : \mathbb{R}^3 \to \mathbb{R}^{3\times 3}$ are the vector and the matrix valued function, respectively, given by

$$G_v^s(v, F, \tau) \triangleq \begin{bmatrix} v_y\omega + F/m \\ -v_x\omega \\ \tau/I \end{bmatrix}, \tag{4.39}$$

$$H_v^s\left(\sigma_{\{4,5,6\}}\right) \triangleq \begin{bmatrix} \sigma_4/m & 0 & 0 \\ 0 & \sigma_5/m & 0 \\ 0 & 0 & \sigma_6/I \end{bmatrix}. \tag{4.40}$$

The system (4.38) is a linear SDE with drift and diffusion coefficients given by G_v^s and H_v^s.

4.3.2 Hybrid Stochastic Brownian-Markov model

Letting $dq \triangleq (dx,\, dy,\, d\theta)^T$ and $dv \triangleq (dv_x,\, dv_y,\, d\omega)^T$, we summarize the two states of our stochastic model as follows:

$$\overline{\mathcal{M}}_g : \begin{cases} dq = G_q^g(q, v)\, dt + H_q^g\left(q, \sigma_{\{1,2\}}\right) dB_q^g \\ dv = G_v^g(F, \tau)\, dt + H_v^g\left(\sigma_{\{4,6\}}\right) dB_v^g \end{cases}$$

$$\overline{\mathcal{M}}_s : \begin{cases} dq = G_q^s(q, v)\, dt + H_q^s\left(q, \sigma_{\{1,2,3\}}\right) dB_q^s \\ dv = G_v^s(v, F, \tau)\, dt + H_v^s\left(\sigma_{\{4,5,6\}}\right) dB_v^s \end{cases}$$

where dB_q^g, dB_v^g, dB_q^s and dB_v^s are given by

$$\begin{aligned} dB_q^g &= [dW_1,\, dW_2]^T, \\ dB_v^g &= [d\widetilde{W}_1,\, d\widetilde{W}_2]^T, \\ dB_q^s &= [dW_1,\, dW_2,\, dW_3]^T, \\ dB_v^s &= [d\widetilde{W}_1,\, d\widetilde{W}_2, d\widetilde{W}_3]^T. \end{aligned}$$

The transition from $\overline{\mathcal{M}}_g$ to $\overline{\mathcal{M}}_s$ and viceversa is regulated by a two-state Markov chain defined by the couple (P, π_0)

$$P = \begin{bmatrix} p_{gg} & p_{gs} \\ p_{sg} & p_{ss} \end{bmatrix}, \quad \pi_0 = \begin{bmatrix} \pi_{0,1} \\ \pi_{0,2} \end{bmatrix}, \tag{4.41}$$

where P is the transition matrix and π_0 is the initial probability vector. The coefficient p_{gg} is the probability to remain in $\overline{\mathcal{M}}_g$, whereas $p_{gs} = 1 - p_{gg}$ is the transition probability from $\overline{\mathcal{M}}_g$ to $\overline{\mathcal{M}}_s$; p_{gs} maps $v_x\omega$ to $[0, 1]$, constructed in such a way to tend to 0 as $v_x\omega$ tends to 0, and to 1 as $v_x\omega$ tends to a critical value denoted with $\overline{v_x\omega}$. The probability to remain in $\overline{\mathcal{M}}_s$ is constructed in a similar way, as $p_{ss} = 1 - p_{sg}$ where p_{sg} maps the lateral velocity v_y to $[0, 1]$ in such a way that p_{sg} tends to zero as $v_y \gg \overline{v_y}$ and rapidly tends to 1 ad v_y approaches $\overline{v_y}$.

The functions that we choose as transition probabilities will be described in Section 4.4.2. The initial probability vector is assumed to be $\pi_0 = [1,\, 0]^T$, meaning that the initial state of the hybrid stochastic system is always $\overline{\mathcal{M}}_g$.

Systems $\overline{\mathcal{M}}_g$ and $\overline{\mathcal{M}}_s$, together with (P, π_0), constitute our novel hybrid stochastic Brownian-Markov model (HSBM).

4.3.3 Existence and Uniqueness of Solutions

A sufficient condition for the existence and uniqueness of solutions to SDEs is to verify that the drift and diffusion terms are Lipschitz functions, see, e.g., [6, 13, 37, 78, 88, 98]. We have

Proposition 4.3.3.1. *Let $(\Omega, \mathcal{F}, \{\mathcal{F}\}_{t\geq 0}, \mathbb{P})$ be a filtered probability space. Consider the stochastic model given by the following systems of SDEs*

$$\overline{\mathcal{M}}_g = \begin{cases} dq = G_q^g(q,v)\,dt + H_q^g\left(q, \sigma_{\{1,2\}}\right) dB_q^g \\ dv = G_v^g(F,\tau)\,dt + H_v^g\left(\sigma_{\{4,6\}}\right) dB_v^g \end{cases} \tag{4.42}$$

with initial state $x_0 = (q_0, v_0)^T$ where B_q^g is a vector of independent $\{\mathcal{F}\}_{t\geq 0}$-adapted Brownian motions, B_v^g is a vector of independent reflected $\{\mathcal{F}\}_{t\geq 0}$-adapted Brownian motions, and

$$\overline{\mathcal{M}}_s = \begin{cases} dq = G_q^s(q,v)\,dt + H_q^s\left(q, \sigma_{\{1,2,3\}}\right) dB_q^s \\ dv = G_v^s(v,F,\tau)\,dt + H_v^s\left(\sigma_{\{4,5,6\}}\right) dB_v^s \end{cases} \tag{4.43}$$

with initial state $x_\xi = (q_\xi, v_\xi)^T$ where B_q^s is a vector of independent $\{\mathcal{F}\}_{t\geq 0}$-adapted Brownian motions, B_v^s is a vector of independent reflected $\{\mathcal{F}\}_{t\geq 0}$-adapted Brownian motions. The two sets $\{G_q^g, G_v^q, G_q^s, G_v^s\}$, $\{H_q^g, H_v^q, H_q^s, H_v^s\}$ of drift and diffusion terms are Lipschitz-continuous matrix-valued functions.

We point out that we provide an explicit proof just for the system of SDEs (4.43) being the most interesting case; the Lipschitz-continuity of drift and diffusion coefficients in (4.42) follows the same argument.

We remark that the control force F and control torque τ are calculated by a digital controller, see Section 4.4.1, by comparing the reference value for v_x and ω with their actual measurements. On every interval $[t_k, t_{k+1}]$, $F = F(t_k)$ and $\tau = \tau(t_k)$ are then kept constant. This means that the body velocity vector $v = (v_x, v_y, \omega)^T$ can be thought as an element of $\mathcal{V} \triangleq [0, v_x^{max}] \times [0, v_y^{max}] \times [0, \omega^{max}] \subseteq \mathbb{R}^3$ that is closed and bounded, hence compact. We use this fact together with the results in [88] to show the Lipschitz-continuity of drift terms and diffusion terms, obtaining existence and uniqueness.

Proof. Consider the system of SDEs given by (4.43).
Dynamic equation: The Lipschitz-continuity of G_v^s requires that the Euclidean norms of the gradient of every component

$$\begin{aligned} g_v^{s,1}(v_x, v_y, \omega) &= v_y\omega + F/m, \\ g_v^{s,2}(v_x, v_y, \omega) &= \tau/I, \\ g_v^{s,3}(v_x, v_y, \omega) &= -v_x\omega, \end{aligned}$$

is bounded on $\mathcal{V}$. Calculations yield to the following vectors and their corresponding Euclidean norms

$$\begin{aligned} ||\nabla g_v^{s,1}||_2 &= ||\,(0, \omega, v_y)\,||_2 = \sqrt{\omega^2 + v_y^2}, \\ ||\nabla g_v^{s,2}||_2 &= ||\,(0,0,0)\,||_2 = 0, \\ ||\nabla g_v^{s,3}||_2 &= ||\,-(\omega, 0, v_x)\,||_2 = \sqrt{\omega^2 + v_x^2}. \end{aligned}$$

The norms are all continuous functions on $\mathcal{V}$ with an absolute minimum in the origin, hence bounded.

Terms m and I in $H_v^s(\sigma_{\{4,5,6\}})$ are constant, so if σ_i is constant as well, then Lipschitz-continuity is straightforward. If σ_i depends on some parameters $(u_1, ..., u_n)$ for $n \in \mathbb{N}$, then it is sufficient to require

$$||\nabla \sigma_i(u)||_2 = \sqrt{\sum_{k=1}^{n} \frac{\partial \sigma_i^k}{\partial u_k}} \leq K_i \quad i = \{4, 5, 6\},$$

where K_i are positive constants, for guaranteeing Lipschitz-continuity of $H_v^s(\sigma_{\{4,5,6\}})$.
Kinematic equation: Consider the drift term $G_q^s(q, v)$ which elements are function of q and v

$$\begin{aligned} g_q^{s,1}(\theta, v_x, v_y) &= v_x \cos\theta - v_y \sin\theta, \\ g_q^{s,2}(\theta, v_x, v_y) &= v_x \sin\theta + v_y \cos\theta, \\ g_q^{s,3}(\omega) &= \omega. \end{aligned}$$

Their gradients are

$$\begin{aligned} \nabla g_q^{s,1} &= (-v_x \sin\theta - v_y \cos\theta, \cos\theta, -\sin\theta, 0)\,, \\ \nabla g_q^{s,2} &= (v_x \cos\theta - v_y \sin\theta, \sin\theta, \cos\theta, 0)\,, \\ \nabla g_q^{s,3} &= (0, 0, 0, 1)\,. \end{aligned}$$

and so we have

$$\begin{aligned} ||\nabla g_q^{s,1}||_2 &= \sqrt{1 + (v_x \sin\theta + v_y \cos\theta)^2}, \\ &\leq \sqrt{1 + (v_x + v_y)^2}, \end{aligned}$$

where the upper bound is a strictly positive continuous function on $\mathcal{V}$ with an absolute minimum in 1 achieved for $v_x = -v_y$ and thus bounded. An identical argument holds true also for $||\nabla g_q^{s,2}||_2$. Finally, $||\nabla g_q^{s,3}|| = 1$ and the Lipschitz-continuity of $G_q^s(q, v)$ is guaranteed.

Consider now the diffusion matrix $H_q^s\left(q, \sigma_{\{1,2,3\}}\right)$, if σ_i are constant, then each entry of the matrix is a bounded periodic differentiable function of θ, which derivative is upper bounded from the value of σ_i itself, thus Lipschitz-continuity of $H_q^s\left(q, \sigma_{\{1,2,3\}}\right)$ is given. Otherwise if $\sigma_i = \sigma_i(u)$ where $u = (u_1, ..., u_n)$ for $n \in \mathbb{N}$ then Lipschitz-continuity of $H_q^s\left(q, \sigma_{\{1,2,3\}}\right)$ is guaranteed if the following condition

$$||\nabla h_q^{s,ij}||_2 = \sqrt{\sum_{k=1}^{n} \left(\frac{\partial h_q^{s,ij}}{\partial u^k}\right)^2 + \left(\frac{\partial h_q^{s,ij}}{\partial \theta}\right)^2} \leq K_{ij}, \tag{4.44}$$

holds true, where K_{ij} are positive constant and $h_q^{s,ij}$ are the non identically zero entries of the diffusion matrix H_q^s. In other words, if σ_i are not constant it is sufficient to assume to be sufficiently regular to ensure the Lipschitz-continuity of the diffusion terms $H_q^s\left(q, \sigma_{\{1,2,3\}}\right)$ and $H_v^s\left(\sigma_{\{4,5,6\}}\right)$.

□

Remark 4.3.3.1. *We point out that requiring Lipschitz continuity of drift and diffusion coefficients was the first result about existence and uniqueness of solutions to SDEs due to Itô,* [66]. *It is a sufficient, but not necessary condition. In real-world applications, systems and phenomena may need to be modeled via SDEs with non-Lipschitz coefficients. In fact, in many cases of interest the diffusion coefficient is only Hölder-continuous, e.g. the Cox-Ingersoll-Ross (CIR) model. In case of non-Lipschitz coefficients* [39, 40] *the analysis becomes quite intricated. Noteworthy there exist results as the Yamada-Watanabe condition* [135] *for the existence and uniqueness of solution to one-dimensional SDEs with locally-Lipschitz drift and diffusion coefficients of the form* $\sigma(x) = |x|^{\alpha}$ *for* $\alpha \in [1/2, 1]$ *and its recent extension to the multidimensional case* [58].

4.4 Simulation Setup

We show realizations of the stochastic model and compare them with those obtained with the deterministic model. We do not require any particular assumption on the diffusion coefficients of terms $H^{g,s}_{q,v}$. We only assume that the coefficients guarantee the Lipschitz-continuity and thus existence and uniqueness of the characterizing SDE. Section 4.4.1 describes the control scheme, while Section 4.4.2 shows and comments the simulation results.

4.4.1 Robot Controller

The control force F and torque τ for the models are calculated by a Proportional-Integral-Derivative (PID) controller with sample time $T_s = 0.01s$, combined with the inverse dynamics of the system described by equations (4.18d) and (4.18e). The reference velocities (v^{ref}, ω^{ref}) are computed by a Pure Pursuit Controller [34], [131] and designed specifically to track a trajectory given by a Bézier curve $q^{ref}(t)$.

Let $t_k = kT_s$ with $k \in \mathbb{Z}_0^+$, the control force and torque at time t_k, $F(t_k), \tau(t_k)$ are computed according to

$$\begin{bmatrix} F(t_k) \\ \tau(t_k) \end{bmatrix} = \frac{1}{T_s} \begin{bmatrix} m & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} u_v(t_k) \\ u_w(t_k) \end{bmatrix}, \tag{4.45}$$

where

$$u_v(t_k) = u_v(t_{k-1}) + A_v e_v^k - B_v e_v^{k-1} + C_v e_v^{k-2}, \tag{4.46}$$

$$u_\omega(t_k) = u_\omega(t_{k-1}) + A_\omega e_\omega^k - B_\omega e_\omega^{k-1} + C_\omega e_\omega^{k-2}, \tag{4.47}$$

with $e_v^k = v^{ref}(t_k) - v(t_k)$ and $e_\omega^k = \omega^{ref}(t_k) - \omega(t_k)$ the errors with respect to the reference velocities at time t_k; the terms A_i, B_i and C_i are given by

$$A_i = \left(K_P^i + K_I^i T_s + \frac{K_D^i}{T_s} \right), \quad B_i = \left(K_P^i + 2\frac{K_D^i}{T_s} \right), \quad C_i = \frac{K_D^i}{T_s},$$

where K_P^i, K_D^i and K_I^i, for $i = v, w$, are the gains of the discrete-time PID regulator [47].

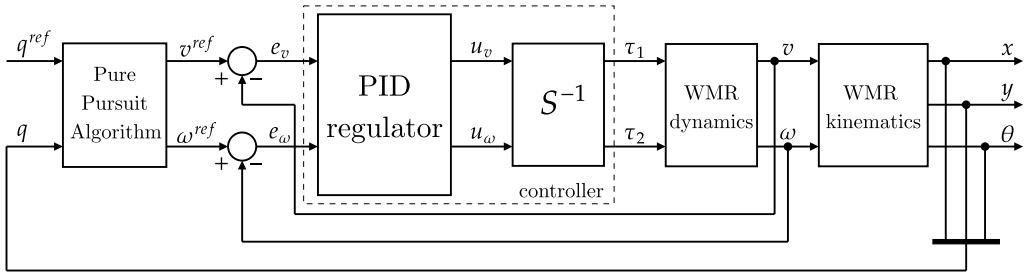


Figure 4.4: Block diagram of the controlled system. The dashed block refer to equations (4.45), (4.46) and (4.47).

The block diagram in Figure 4.4 describes the control loop. It is worth highlighting that the ODEs in $\mathcal{M}_g$ and $\mathcal{M}_s$ are solved by using the Forward Euler approximation scheme, while the system of SDEs in HSBM model are solved by adopting the Euler-Maruyama approximation scheme, see e.g [63], with discretization time-step $h = T_s/10$.

4.4.2 Setup Parameters

Table 4.2 summarizes the dynamic parameters for the deterministic model of the robot in our numerical simulations and the friction coefficients characterizing the external forces described in Section 4.2.2.

parameter	**value**	**description**
ℓ	0.760 m	length of robot
b	0.330 m	semi-axle
m_r	107 kg	robot mass (no wheels)
m_w	4 kg	one wheel mass
m	115 kg	robot total mass
c_{rr}	0.08	rolling friction coeff.
μ_s	0.70	lateral static friction coeff.
μ_k	0.56	lateral dynamic friction coeff.

Table 4.2: Dynamic parameters and friction coefficients.

Point	P_0	P_1	P_2	P_3	P_4	P_5	P_6	P_7
x	-4	0	4	8	8	4	0	-4
y	3	3	3	3	0	0	0	0

Table 4.3: The control points of the Bézier curve. The path starts in P_0 and ends in P_7.

The total inertia of the robot, given by equation (4.1), is therefore $I = 9.105\, kgm^2$. The robot follows the path in Figure 4.5 obtained via a planar 7 degrees Bézier curve whose control points are listed in Table 4.3. The robot travels along the path with a constant speed of $v_x = 2.20\, m/s$ and the maximum angular speed is $|\omega^{max}| = 4.80\, rad/s$.

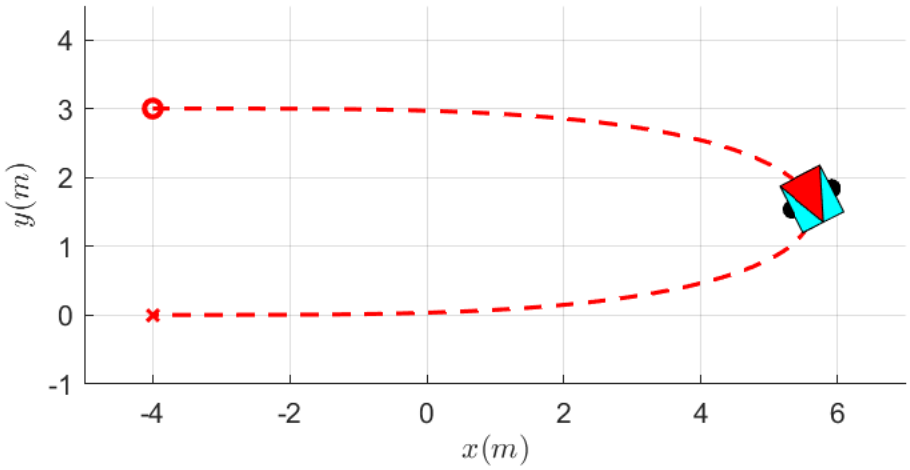


Figure 4.5: A snapshot of the robot while following the path, obtained via Bézier curve whose control path are reported in Table 4.3. The starting and ending points are put in evidence with a circle and a cross, respectively. The red triangle on top of the robot serves to show the orientation of the robot.

We assume constant diffusion coefficients for the stochastic model given by

$$\begin{aligned} \sigma_1 &= 0.03, \qquad \sigma_4 = 0.85, \\ \sigma_2 &= 0.03, \qquad \sigma_5 = 0.85, \\ \sigma_3 &= 0.03, \qquad \sigma_6 = 0.85. \end{aligned}$$

The transition probability p_{gs} in the transition matrix P in (4.41) is

$$p_{gs}(v_x\omega) \triangleq \frac{1}{1 + e^{-\alpha_1(|v_x\omega| - \gamma_1)}} \tag{4.48}$$

where $\alpha_1 \in \mathbb{R}_{>0}$ and $\gamma_1 = \overline{v_x\omega}$, i.e., a parametric sigmoid function on the absolute value of $v_x\omega$, centered in the deterministic critical value $\overline{v_x\omega}$. The parameter α_1 controls the rate of growth of p_{gs}, while γ_1 translates the function along the abscissa axis.

The choice of the parametric function (4.48) is motivated by the need of having a transition probability that is close to zero when $|v_x\omega| \ll \overline{v_x\omega}$ and also rapidly growing to 1 as $|v_x\omega| \to \overline{v_x\omega}$. Similarly, for p_{gs} we assume

$$p_{sg}(v_y) \triangleq \frac{-1}{1 + e^{-\alpha_2(|v_y| - \gamma_2)}} + 1, \tag{4.49}$$

where $\alpha_2 \in \mathbb{R}_{>0}$ and $\gamma_2 = \overline{v_y}$ choosen in such a way to have a function that rapidly grows to 1 as $|v_y| \to 0$.

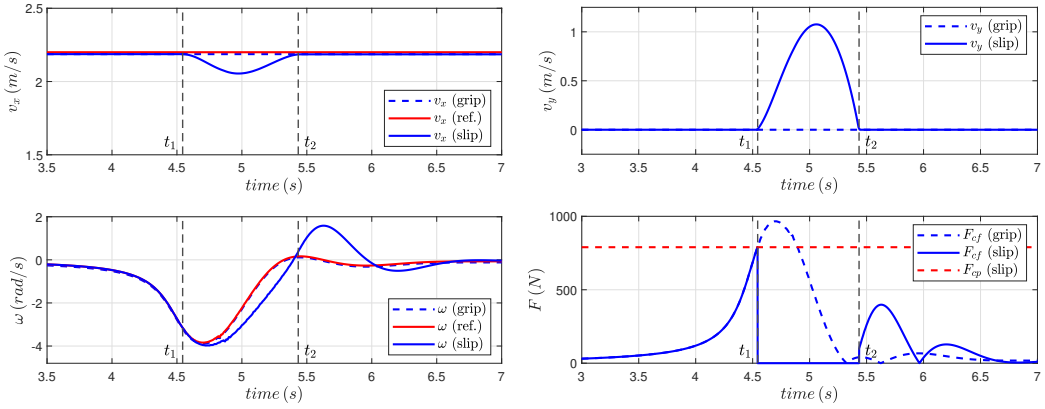
We model the behavior of transition probabilities (4.48) and (4.49) by assuming $\alpha_1 = \alpha_2 = 120$, $\overline{v_x\omega} = 6.90\,(m\,rad)/s^2$ and $\overline{v_y} = 0.05\,m/s$.

4.5 Experimental Results

We show and comment results obtained from the numerical implementation of the deterministic model and the HSBM model. In particular, we discuss the velocity profiles and the transition conditions that allow the deterministic model to switch from $\mathcal{M}_g$ to $\mathcal{M}_s$, and the stochastic model from $\overline{\mathcal{M}}_g$ to $\overline{\mathcal{M}}_s$, and viceversa.

4.5.1 Deterministic Model

Figure 4.6a shows the forward velocity v_x and angular velocity ω of the deterministic model, obtained by ignoring the transition condition $|F_{cf}| \leq |F_{cp}|$, i.e., by considering only constraint (4.16) along all the path tracked until $T = 7s$ (blue dashed lines) and those obtained by allowing lateral slip phenomenon, i.e., allowing transitions from $\mathcal{M}_g$ to $\mathcal{M}_s$ and viceversa (blue continuous lines), compared with reference forward and angular velocities calculated by the Pure Pursuit controller (continuous red lines).



(a) (b)

Figure 4.6: (a): Forward and angular velocity time-series of the deterministic model on the time window $3.5s \leq t \leq 7s$, in case of perfect grip and in case of grip loss, compared with the reference ones; (b): Lateral velocity, on the time window $3s \leq t \leq 7s$, and centrifugal force profiles in case the perfect grip and in the case of grip loss, the latter compared with the maximum sustainable centripetal force.

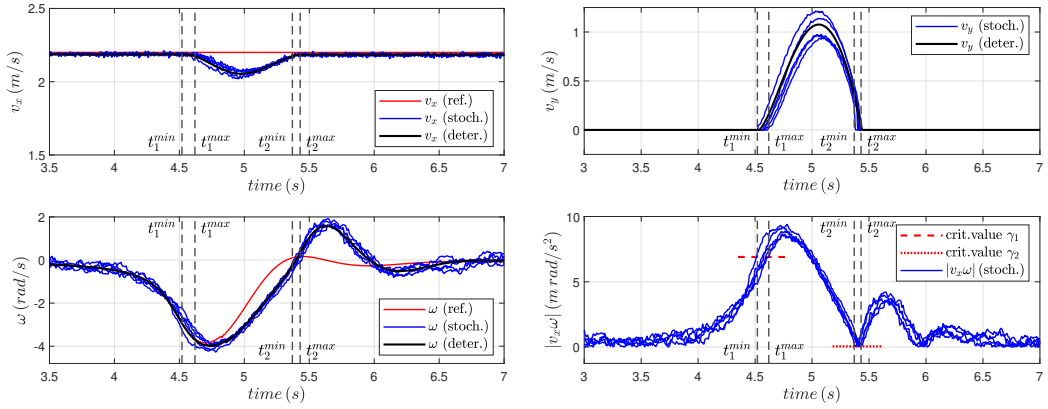
Till $t_1 = 4.54s$ the robot has perfect grip thus the reference velocities are perfectly tracked by the PID controller, subsequently the system switches from $\mathcal{M}_g$ to $\mathcal{M}_s$ and loss of grip phenomenon deviates v_x and ω from their reference values until $t_2 = 5.43s$. At $t = t_2$, the system switches back to $\mathcal{M}_g$, gripping is restored and v_x and ω converges again to the corresponding reference values.

Figure 4.6b shows the lateral velocity v_y and centrifugal force F_{cf} in the case of perfect grip (blue dashed line). Centrifugal force is compared with the maximum sustainable centripetal force F_{cp} (red dashed line). Ignoring condition $|F_{cf}| \leq |F_{cp}|$ implies that motion equations are given by $\mathcal{M}_g$, therefore the value of F_{cf} can overcome the maximum sustainable centripetal force, determined by μ_s in Table 4.2 and equal to $F_{cp} = 789.71N$. If slip phenomenon is allowed, the condition $|F_{cf}| \leq |F_{cp}|$ is taken into account and when it is violated, at $t_1 = 4.54s$, the model passes from $\mathcal{M}_g$ to $\mathcal{M}_s$ and it remains there, where F_{cf} has no role, until $t_2 = 5.43s$ when $|v_y|$ decayed to zero due to the action of the lateral friction force F_{lat}, determined by μ_k in Table 4.2.

4.5.2 HSBM Model

Figure 4.7a reports five forward velocity v_x time-series and angular velocity ω time-series as realizations of the Brownian-Markov model (blue irregular

lines), compared with the reference values calculated by the Pure Pursuit controller (red continuous line) and the velocities obtained as solutions of the deterministic model.



(a) (b)

Figure 4.7: (a): Forward and angular velocity profiles of 5 realizations of the Brownian-Markov model and 1 of the deterministic model, on the time window $3.5s \leq t \leq 7s$, compared with reference velocities; (b): Lateral velocity of 5 realizations of the Brownian-Markov model and 5 realizations of $|v_x\omega|$ and the critical values γ_1 and γ_2.

Figure 4.7b shows five realizations of the lateral velocity v_y (blue irregular lines) and the deterministic v_y (black continuous line); and five realizations of $|v_x\omega|$ (blue irregular lines) and the corresponding critical value (red dashed line). In Figure 4.7a every realization of the HSBM model has its own transition times from $\overline{\mathcal{M}}_g$ to $\overline{\mathcal{M}}_s$ and viceversa, all contained in the intervals $I_1 = [t_1^{min}, t_1^{max}]$ and $I_2 = [t_2^{min}, t_2^{max}]$, respectively (black vertical dashed lines).

As it is possible to notice in Figure 4.7b, for $t_1 \in I_1$, in every realization the absolute value $|v_x\omega|$ is very close to the critical value γ_1. The Brownian-Markov model then switches state from $\overline{\mathcal{M}}_g$ to $\overline{\mathcal{M}}_s$ and remains in slipping mode until time $t_2 \in I_2$, at which the lateral velocity is very close to γ_2.

In Figure 4.7b it is possible to observe two realizations above and three below the deterministic one. The under/over approximations are determined by the fact that transitions from $\overline{\mathcal{M}}_g$ to $\overline{\mathcal{M}}_s$ and viceversa occurs at different times within the transition intervals. In particular, the state transition times within I_1 and I_2 in our simulations, reported in ascending order, are

$$\begin{array}{rclrcl} t_1(min) & = & 4.53s & t_2(min) & = & 5.38s \\ t_1(2) & = & 4.58s & t_2(2) & = & 5.40s \\ t_1(3) & = & 4.59s & t_2(3) & = & 5.41s \\ t_1(4) & = & 4.61s & t_2(4) & = & 5.42s \\ t_1(max) & = & 4.63s & t_2(max) & = & 5.44s \end{array}$$

Finally, for $t > t_2^{max}$, all realizations of v_y of the Brownian-Markov model have switched back from $\overline{\mathcal{M}}_s$ to $\overline{\mathcal{M}}_g$, thus $v_y = 0$ for every $t \in (t_2^{max}, T]$ and v_x and ω return to converge to the corresponding reference values.

Figure 4.8a (a) shows a example of transition probabilities p_{gs} (red continuous line) and p_{sg} (blue continuous line) of the Markov chain (P, π_0) regulating

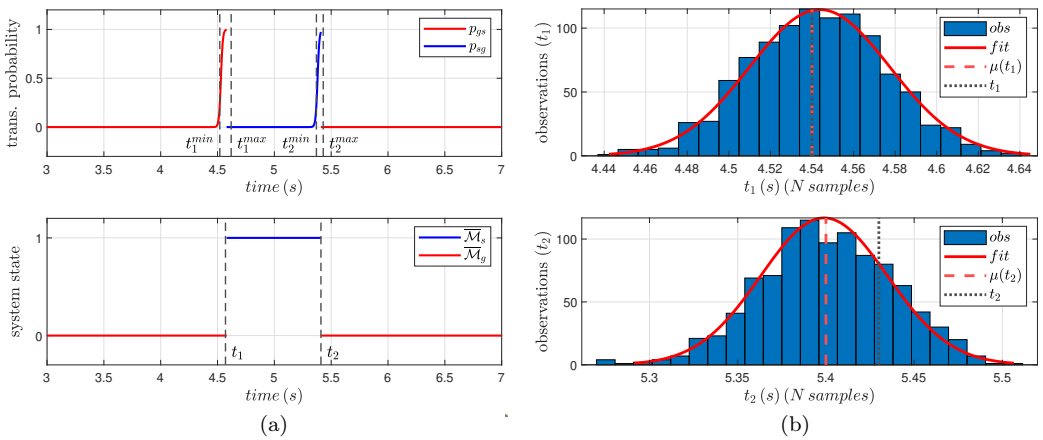


Figure 4.8: (a): An example of the transition probabilities p_{gs} in (4.48) and p_{sg} in (4.49) and the corresponding state transition diagram, on the time window $3s \leq t \leq 7s$; (b): Histograms of the N observations for t_1 and t_2.

the transitions between $\overline{\mathcal{M}}_g$ and $\overline{\mathcal{M}}_s$, for $3s \leq t \leq 7s$. The initial state is $\overline{\mathcal{M}}_g$ and for $t < t_1^{min}$, p_{gs} is close to zero. For $t \in I_1$, p_{gs} grows rapidly to 1 since $|v_x\omega|$ is close to γ_1 and a transition from $\overline{\mathcal{M}}_g$ to $\overline{\mathcal{M}}_s$ is likely to occur. The transition probability that now matters is p_{sg} and, for $t \in I_2$, it rapidly grows to 1 as the lateral velocity tends to γ_2. A transition back to $\overline{\mathcal{M}}_g$ happens and p_{gs} returns to zero. Figure 4.8a (b) shows the related state transition diagram together with the transition times t_1 and t_2.

The transition times are random variables since changes of system state depend on the transition probabilities (4.48) and (4.49) of the Markov chain (P, π_0), that in turn depends on the velocities affected by Brownian noise. We denote them with $\boldsymbol{t_1}$ and $\boldsymbol{t_2}$, respectively. The mean values $\mu(\boldsymbol{t_i})$ and the standard deviations $\sigma(\boldsymbol{t_i})$, $i = \{1, 2\}$, over $N = 10^3$ samples of transition times are given by

$$\mu(\boldsymbol{t_1}) = 4.54s \quad \sigma(\boldsymbol{t_1}) = 0.03s,$$
$$\mu(\boldsymbol{t_2}) = 5.40s \quad \sigma(\boldsymbol{t_2}) = 0.04s.$$

Note that $\mu(\boldsymbol{t_1})$ and $\mu(\boldsymbol{t_2})$ are close to the corresponding values for which the conditions $|F_{cf}| \leq |F_{cp}|$ and $|v_y| > 0$ are violated in the deterministic model, $t_1 = 4.54s$ and $t_2 = 5.43s$.

Figure 4.8b reports the histogram of the N observations of transition times (blue bars) obtained by simulating the HSBM model, together with two vertical lines representing their mean values $\mu(\boldsymbol{t_1})$ and $\mu(\boldsymbol{t_2})$ (red dashed), the corresponding values in the deterministic model t_1 and t_2 (black dashed) and the fitting curve (red continuous line).

It is possible to notice that the observations of the transition times follow a unimodal trend, that can be approximate by Gaussian distributions with parameters $\mu(\boldsymbol{t_1})$, $\sigma(\boldsymbol{t_1})$ and $\mu(\boldsymbol{t_2})$, $\sigma(\boldsymbol{t_2})$, respectively.

Chapter 5

Conclusions and Future Works

The generation of safe trajectories for both mobile robots and industrial manipulators is essential to improve production efficiency, a higher level of automation and reduction of the likelihood of undesirable and unpleasant events such as work accidents.

We started this thesis with an analysis of the state of the art on trajectory planning using the Velocity Obstacle. We constructed the taxonomy with the aim of proposing a broad and clear overview.

Next, we proposed a safe trajectory planner designed for pairs of holonomic robots that have to cooperate to transport a long and heavy payload in an automated warehouse, where other autonomous mobile agents are present. The planner is divided into two levels, global and local. The first, based on Voronoi diagrams, generates collision-free trajectories with respect to expected and fixed obstacles such as the shelves of the warehouse. The second, based on Velocity Obstacles, is designed to ensure the safety of the two constrained agents by calculating collision-free trajectories with respect to unforeseen obstacles or other mobile robots, taking into account the rigidity constraints that binds the pair of agents.

Then, since Velocity Obstacle has always been used only for the navigation of ground and aerial mobile robots, we focused on adapting the method to manipulators. We started with the planar case by simulating a two-link manipulator with two revolute degrees of freedom. The planner is based on the construction of the so-called Reduced Multiple Velocity Obstacle (RMVO), the set of all safe velocities for the end-effector and for the distal link of the manipulator. As future work, we aim to adapt the Velocity Obstacle to manipulators operating in the three-dimensional space by extending the proposed planner to the case of anthropomorphic manipulators with 3+ degrees of freedom.

The possibility to rely on reliable dynamic models is of equally importance in applications such as designing motion control loops, filters and computer simulation of robots. We designed a model-based feedforward control loop and a payload estimation routine for a D3-1200 Delta Robot made available by the company SIPRO Srl. We started from the standard dynamic model of the literature, that is derived by considering a significant simplification of the robot kinematics chains, and we enhanced it with a gray-box model parameter

identification technique in order to recover the inertia and gravity compensation effects lost due to the simplifications. Although the model with identified parameters provided better results in terms of estimated control torques, the parameter identification is a highly time-consuming procedure and furthermore parameters depend on the considered trajectory. For these reasons, using the Euler-Lagrange equations, we derived a novel mathematical model without any kind of simplification on the kinematic chains. The resulting model has proven to be more accurate than the previous one but paying the price of a considerably higher computational complexity. As future work, we aim to collect and analyze data from the implementation of the aforementioned feedfoward motion control loop applied to the D3-1200 Delta Robot and optimize the algorithm for high-perfomance computational tasks.

Finally, we developed a novel dynamic model for mobile robots. In particular, we presented a novel two-state hybrid stochastic dynamic model for differential-drive wheeled mobile robots, in which the action of dissipative forces is governed by independent reflected Brownian motions and the transition between two states, as consequence of the action of inertial external forces, is governed by a Markov chain. The hybrid stochastic model is important because, unlike the models known in the literature, it is able to take into account the action of inertial forces that can determine undesired lateral motion caused by losses of grip. We proved the existence and uniqueness of the solutions of the SDEs and provided numerical simulations to show the effectiveness of our model. As future work, we aim to experimentally validate the hybrid stochastic dynamic model on a real differential-drive robot.

Appendix A

Lie Groups Theory

A.1 Matrix Lie Groups

A Lie Group consists in a pair $(G, \star)$ where G has the structure of a differentiable manifold and $\star$ is a binary operation that enjoys certain group axioms. In what follows, we will refer to the plane with $\mathbb{R}^2$ and to the 3D space with $\mathbb{R}^3$. Matrix Lie Groups are a particular class of Lie Groups [22, 61, 64] that naturally arises in robotics and generally in rigid body mechanics [22]. In particular, it is useful to describe the position, orientation and velocity of a rigid body on the plane and in the space, exploiting the tools of analytical mechanics.

Definition A.1.0.1. ***(Group)*** *A set G endowed with a binary operation $\star : G \times G \to G$, $(a, b) \mapsto a \star b$ is a group if satisfies the following three properties:*

1. *$a \star (b \star c)$ for all $a, b, c \in G$;*
2. *there exists $e \in G$ such that $a \star e = e \star a = a$ for all $a \in G$;*
3. *there exists $a^{-1} \in G$ such that $a \star a^{-1} = e$ for all $a \in G$.*

In other words, $(G, \star)$ is a group if $\star$ is associative and there exist both the identity element e and inverse element a^{-1} for all a in G. If the $\star$ is also commutative, i.e., $a \star b = b \star a$, then $(G, \star)$ is called *Abelian group*. Intuitively, $H \subset G$ is a subgroup of G if $(H, \star)$ is a group with respect to the binary operation $\star$ defined on G.

Definition A.1.0.2. ***(Lie Group)*** *A Lie group $(G, \cdot)$ is a group in which G has the structure of a differentiable manifold and the operations $G \times G \to G$ such that $(a, b) \mapsto a \cdot b$ and $G \to G$ such that $a \mapsto a^{-1}$ are both differentiable.*

Before giving an explicit definition of what a Matrix Lie Group is, we need to introduce some other definitions and concepts.

Definition A.1.0.3. ***(Homogeneous Coordinates)*** *Consider a point $x \in \mathbb{R}^3$. The homogeneous coordinates associated to x are defined as $\bar{x} \triangleq (x, 1) \in \mathbb{R}^4$.*

Consider a spatial reference frame $\Sigma_s = \{O_s; s_1, s_2, s_3\}$ and a rigid body $\mathcal{B}$ in $\mathbb{R}^3$ on which we attach a so called *body frame*, $\Sigma_b = \{O_b; b_1, b_2, b_3\}$. Let x_s denote the pose of $\mathcal{B}$ in Σ_s and x_b the position of a particle of $\mathcal{B}$ with respect to the body frame Σ_b. Then

$$x_s = p + Rx_b \tag{A.1}$$

where R is the rotation matrix expressing the orientation of $\mathcal{B}$ and p is the position vector of its *center of mass* (CoM) expressed with respect to Σ_s. As in Definition A.1.0.3, we define $\bar{x}_s \triangleq (x_s, 1) \in \mathbb{R}^4$ and $\bar{x}_b \triangleq (x_b, 1) \in \mathbb{R}^4$, thus equation (A.1) can be rewritten as

$$\bar{x}_s = \begin{bmatrix} R & p \\ 0_{1\times 3} & 1 \end{bmatrix} \bar{x}_b, \quad \bar{x}_b = \begin{bmatrix} R^T & -R^T p \\ 0_{1\times 3} & 1 \end{bmatrix} \bar{x}_s. \tag{A.2}$$

The rotation matrix R belongs to the *Special Orthogonal group* $SO(n)$ [1]. Let $GL(n;\mathbb{R})$ (resp. $GL(n;\mathbb{C})$) be the *General Linear group*, i.e., the group of *all invertible matrices of order* n with real entries (resp. complex entries). It is a subset of $\mathcal{M}(n;\mathbb{R})$ (resp. $\mathcal{M}(n;\mathbb{C})$), the set of all $n \times n$ matrices with real (resp. complex) entries. $SO(n)$ is defined as follows.

Definition A.1.0.4. ***(Special Orthogonal Group)*** *The Special Orthogonal group $SO(n)$ is defined as*

$$SO(n) \triangleq \{R \in O(n) : \det(R) = 1\},$$

where $O(n)$ is the group of orthogonal matrices of dimension $n \in \mathbb{N}$, i.e., all the matrices $R \in GL(n, \mathbb{R})$ such that $R^T R = RR^T = I_d$.

In particular, $SO(2)$ is the group of rotations on the plane and $SO(3)$ is the group of rotations in space. The displacement of a rigid body is described by the position of its CoM and by its orientation on the plane or in the space, therefore we need to introduce the following definition.

Definition A.1.0.5. ***(Special Euclidean Group)*** *The Special Euclidean group $SE(n)$ is the group whose elements are expressed in the following matrix form*

$$g = \begin{bmatrix} R & p \\ 0_{1\times n} & 1 \end{bmatrix}, \tag{A.3}$$

where $R \in SO(n)$ and $p \in \mathbb{R}^n$.

In particular, $SE(2)$ and $SE(3)$ are the groups of rigid body displacements in $\mathbb{R}^2$ and in $\mathbb{R}^3$, respectively. Consider the usual matrix multiplication operation, denoted with " $\cdot$ ". We can now give a precise definition of Matrix Lie Group.

Definition A.1.0.6. ***(Matrix Lie Groups)*** *The groups $O(n)$, $SO(n)$ and $SE(n)$ equipped with the matrix multiplication operation are Matrix Lie groups.*

[1] we have omitted to specify the binary operation included in the definition of group, for the sake of simplicity.

Alternatively, we can assert that a Matrix Lie Group is any closed subgroup of $GL(n,\mathbb{R})$ or $GL(n,\mathbb{C})$ endowed with the matrix multiplication [64].

We remark that $O(n)$, $SO(n)$ and $SE(n)$ are the only Matrix Lie groups of interest in this thesis, there exist many other Matrix Lie Groups that we are not going to treat, see [61, 103]. We conclude this subsection with some examples.

Example A.1.0.1. Consider a fixed frame of reference $\Sigma_s = \{O; s_1, s_2, s_3\}$ in $\mathbb{R}^3$ and the group of three-dimensional rotations $SO(3)$ equipped with the usual matrix multiplication. A basis $B = \{e_1, e_2, e_3\}$ for $SO(3)$ is given by the following matrices

$$e_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_1 & -\sin\theta_1 \\ 0 & \sin\theta_1 & \cos\theta_1 \end{bmatrix} \quad e_2 = \begin{bmatrix} \cos\theta_2 & 0 & \sin\theta_2 \\ 0 & 1 & 0 \\ -\sin\theta_2 & 0 & \cos\theta_2 \end{bmatrix},$$

$$e_3 = \begin{bmatrix} \cos\theta_3 & -\sin\theta_3 & 0 \\ \sin\theta_3 & \cos\theta_3 & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

i.e., the elementary rotations matrices along the s_1, s_2 and s_3 axis.

Example A.1.0.2. Consider a fixed spatial frame of reference $\Sigma_s = \{O; s_1, s_2\}$ in $\mathbb{R}^2$ and a rigid body $\mathcal{B}$, whose CoM position in Σ_s is expressed by the vector $p = (x_1, x_2)^T$ and whose orientation is given by the angle θ with respect to Σ_s. Then, the displacement $g \in SE(2)$ in (A.3) of $\mathcal{B}$ is given by

$$g = \begin{bmatrix} R & p \\ 0_{1\times 2} & 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & x_1 \\ \sin\theta & \cos\theta & x_2 \\ 0 & 0 & 1 \end{bmatrix}. \tag{A.4}$$

A.2 Matrix Lie Algebras

In the previous subsection we introduced the concepts of Matrix Lie Group and displacement, that are fundamental to describe the position and the orientation of a rigid body. In order to have all the necessary, we must also introduce the concept of Matrix Lie Algebra, that arises when dealing with the velocity of a rigid body.

Definition A.2.0.1. ***(Matrix commutator)*** *Let $A, B \in M$ with $M = \mathcal{M}(n,\mathbb{R})$ or $\mathcal{M}(n,\mathbb{C})$. The matrix commutator is the $M \times M \to M$ operation such that*

$$[A, B] \triangleq AB - BA. \tag{A.5}$$

Proposition A.2.0.1. *Consider three matrices $A, B, C \in \mathcal{M}(n;\mathbb{R})$. The matrix commutator operator* (A.5) *enjoys the following properties:*

- $[A, B] = -[B, A]$ *(skew-symmetry), and*
- $[A, [B, C]] + [B, [C, A]] + [C, [A, B]] = 0$ *(Jacobi's Identity).*

Proof. See [64]. □

Definition A.2.0.2. ***(Skew-symmetric matrices)*** *The set of skew-symmetric matrices* $\mathfrak{so}(n)$ *with real entries is defined as*

$$\mathfrak{so}(n) \triangleq \{S \in \mathcal{M}(n, \mathbb{R}) : S^T = -S\}.$$

Consider now a rigid body $\mathcal{B}$ whose orientation is given by the matrix $R \in SO(n)$, $n = \{2, 3\}$. It is possible to define the

- **body angular velocity** $\hat{\Omega} = R^T \dot{R}$, and the
- **spatial angular velocity** $\hat{\omega} = \dot{R} R^T$,

where $\dot{R}$ is the derivative with respect to time of R. $\hat{\Omega}$ is the velocity of $\mathcal{B}$ in Σ_s, while $\hat{\omega}$ is the relative velocity of Σ_s with respect to Σ_b. The body and spatial angular velocities are represented by skew-symmetric matrices, i.e.

$$\hat{\Omega} = \begin{bmatrix} 0 & -\Omega_3 & \Omega_2 \\ \Omega_3 & 0 & -\Omega_1 \\ -\Omega_2 & \Omega_1 & 0 \end{bmatrix}, \quad \hat{\omega} = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix}, \tag{A.6}$$

where $\hat{\Omega}, \hat{\omega} \in \mathfrak{so}(3)$. They also admit a vector representation, obtained via the isomorphism between vector spaces given by $^\vee : \mathfrak{so}(3) \to \mathbb{R}^3$, whose inverse is the map $^\wedge : \mathbb{R}^3 \to \mathfrak{so}(3)$ [22]:

$$\hat{\Omega}^\vee = (\Omega_1, \Omega_2, \Omega_3) \triangleq \Omega, \qquad \hat{\omega}^\vee = (\omega_1, \omega_2, \omega_3) \triangleq \omega. \tag{A.7}$$

Example A.2.0.1. Consider the group element represented by the following matrix

$$R = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix} \in SO(3),$$

where $\theta \in [0, 2\pi]$. Let denote $\dot{\theta} = \omega$. We have

$$\hat{\Omega} = R^T \dot{R} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -\omega \\ 0 & \omega & 0 \end{bmatrix} \in \mathfrak{se}(3),$$

thus the corresponding vector representation is $\Omega = (\omega, 0, 0) \in \mathbb{R}^3$. In particular, in this case we have $\hat{\Omega} = \hat{\omega}$.

We can now define the space of Twists, with a construction similar to the one that allowed to construct $SE(n)$, whose elements admit the following matrix representation

$$\xi \triangleq \begin{bmatrix} \hat{\omega} & v \\ 0_{1\times n} & 0 \end{bmatrix}, \tag{A.8}$$

where $\hat{\omega} \in \mathfrak{so}(n)$ and $v \in \mathbb{R}^n$, for $n = \{2, 3\}$.

Definition A.2.0.3. ***(Twists)*** *The space of twists* $\mathfrak{se}(n)$ *is defined by*

$$\mathfrak{se}(n) \triangleq \{\xi : \xi \text{ as defined in (A.8)}\}.$$

Example A.2.0.2. The space $\mathfrak{se}(2)$ is the space of elements whose matrix representation is given by

$$\xi = \begin{bmatrix} 0 & -\omega & v_x \\ \omega & 0 & v_y \\ 0 & 0 & 0 \end{bmatrix} = \omega e_1 + v_x e_2 + v_y e_3,$$

where the elements

$$e_1 = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad e_2 = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad e_3 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix},$$

are a basis for $\mathfrak{se}(2)$.

Consider a mechanical system in $\mathbb{R}^3$. The concept of twist arises from the fact that the system of ODEs describing the motion of such system, given by

$$\begin{cases} \dot{g}(t) = g(t)A(t) \\ g(0) = g_0 \end{cases} \qquad t \in \mathbb{R},\ g, A \in \mathbb{R}^{4\times 4},$$

has solution[2] $g(t) \in SE(3)$ if and only if $A(t) \in \mathfrak{se}(3)$. It is possible to define the spatial and the body angular velocities in (A.6) in terms of g as follows

$$\xi_s(t) = \dot{g}(t)g^{-1}(t),$$
$$\xi_b(t) = g^{-1}(t)\dot{g}(t).$$

The isomorphism $^\vee : \mathfrak{se}(3) \to \mathbb{R}^3 \oplus \mathbb{R}^3$ and its inverse $^\wedge$ allow to represent ξ in vector form. For example, we have

$$\begin{bmatrix} \hat{\omega} & v \\ 0_{1\times 3} & 0 \end{bmatrix}^\vee = (\omega, v) \in \mathbb{R}^3 \oplus \mathbb{R}^3,$$

with $\omega = (\omega_1, \omega_2, \omega_3)$. At this point we can give the definition of Lie Algebra [22, 64].

Definition A.2.0.4. ***(Lie Algebra)*** *A (real) Lie Algebra is a (real) vector space* V *endowed with a bilinear operation* $[\cdot,\cdot] : V \times V \to V$*, called* ***Lie Brackets****. Consider* $v_1, v_2, v_3 \in V$*, the Lie Brackets satisfy the following properties*

1. $[v_1, v_2] = -[v_2, v_1]$ *(skew-symmetry), and*
2. $[v_1, [v_2, v_3]] + [v_2, [v_3, v_1]] + [v_3, [v_1, v_2]]$ *(Jacobi's Identity)*

A non-empty subset U of V is a Lie Subalgebra of V if it is a Lie Algebra with respect to the bracket operation defined on V.

[2]The solution is a curve in $SE(3)$, i.e., a map $g : \mathbb{R} \to SE(3)$ such that $t \mapsto g(t) \in SE(3)$.

Example A.2.0.3. The following are vector spaces Lie Algebras:

- $V = \mathbb{R}^3$ equipped with the usual cross product " $\times$ " operation;
- the space $\{f : V \to V : \ f \text{ is linear}\}$ equipped with the operation $[f, g] = f \circ g - g \circ f$, where $\circ$ denotes the composition of functions;
- the space $\Gamma(\mathcal{M})$ of all smooth vector fields on the smooth manifold $\mathcal{M}$, equipped with the Jacobi-Lie brackets operation $[X, Y] = \mathcal{L}_X Y$ for $X, Y \in \Gamma(\mathcal{M})$, where $\mathcal{L}_X Y$ is the Lie derivative of Y with respect to X.

Definition A.2.0.5. ***(Matrix Lie Algebras)*** *The spaces $\mathfrak{so}(n)$ and $\mathfrak{se}(n)$ endowed with the matrix commutator operation* (A.5) *are Matrix Lie Algebras.*

As we did in the previous section, we remark those appearing in the previous definition are the only Lie Algebras of interest. There exist many more Matrix Lie Algebras that we are not going to deal with, see [22, 64].

Proposition A.2.0.2. *For any Matrix Lie Group G, the tangent space at the identity $T_I G$ is a Matrix Lie Algebra $\mathfrak{g}$, called the Lie Algebra of G. On the other hand, any Matrix Lie Algebra is the Lie Algebra of some Matrix Lie Group.*

Proof. Omitted. Proposition 5.8 of [64]. □

Proposition A.2.0.3. *A Matrix Lie Algebra is a vector subspace of $\mathcal{M}(n, \mathbb{R})$ endowed with the usual operations of scalar multiplication and matrix addition that is also closed with respect to the matrix commutator operation* (A.5)*. Let G be a group, it is indeed possible to prove that $[A, B] \in T_I G$ for any $A, B \in T_I G$.*

Proof. Omitted. Propositions 5.6 and 5.7 of [64]. □

We identify the Lie Algebra $\mathfrak{g}$, of a Lie Group G, with the tangent space at the identity element $e \in G$ endowed with the Lie Brackets operation $[u, v] = [u_L, v_L](e)$[3]. It is possible to prove that $T_e G \simeq T_g G$ for any element $g \in G$, i.e., there exists an isomorphism between the tangent space at the identity and any other tangent space. This fact is at the basis of the Proposition A.2.0.2. Furthermore, it is possible to demonstrate that such isomorphism between vector spaces induces a vector bundle isomorphism from the tangent bundle[4] TG to $G \times T_e G$, allowing us to write $TG \simeq G \times \mathfrak{g}$. For further details, we refer to Chapter 5 of [22] and [64].

Example A.2.0.4. The space $\mathfrak{se}(3)$ of all the matrices of the form

$$\begin{bmatrix} A & v \\ 0_{1\times 3} & 0 \end{bmatrix}$$

where A is skew-symmetric and $v \in \mathbb{R}^3$ is the Lie Algebra of the group of rigid body displacements $SE(3)$.

[3] u_L and v_L are left-invariant vector fields whose values at e are u and v, respectively.
[4] TG is the disjoint union of all the tangent spaces $T_g M$, for $g \in G$.

Note that to a Lie group corresponds only one Lie Algebra, but a single Lie Algebra can correspond to more than one Lie group.

Example A.2.0.5. Consider the Matrix Lie Group $(O(n), \cdot)$. The tangent space to $O(n)$ at the identity, $T_I O(n)$, is the vector space of skew-symmetric matrices $\mathfrak{so}(n)$. Consider any pair $A, B \in \mathfrak{so}(n)$, we have

$$\begin{aligned}[A,B]^T + [A,B] &= (AB - BA)^T + (AB - BA) \\ &= B^T A^T - A^T B^T + AB - BA = 0_{n\times n},\end{aligned}$$

since $A^T = -A$ and $B^T = -B$, where $0_{n\times n}$ is the null matrix of order n. This proves that $\mathfrak{so}(n)$ is closed under the matrix commutator operation, therefore $(\mathfrak{so}(n), [\cdot,\cdot])$ is the Lie Algebra of $(O(n), \cdot)$, but it is also possible to prove that $\mathfrak{so}(n)$ is also the Lie Algebra of the Special Orthogonal group $SO(n)$, see Example 5.12 of [64].

We conclude the section with the definitions of *adjoint operator* and *co-adjoint operator*, the last concept about Lie Groups theory that we need to introduce. Let V^* denote the *dual space* of V, i.e., the space of all linear maps from V to $\mathbb{R}$.

Definition A.2.0.6. ***(Adjoint Operator)*** *Let $(V, [\cdot,\cdot])$ be a Lie Algebra and $v \in V$. Define $\mathrm{ad}_v : V \to V$ such that*

$$\mathrm{ad}_v u = [v, u] \text{ for any } u \in V.$$

The linear map $\mathrm{ad}_v u$ is called adjoint operator corresponding to $v \in V$.

Definition A.2.0.7. ***(Co-adjoint Operator)*** *The co-adjoint operator ad_v^* associated to ad_v is the linear map $\mathrm{ad}_v^* : V^* \to V^*$ such that, for every $v \in V$ the following inequality holds true*

$$\langle \mathrm{ad}_v^* \alpha, w\rangle = \langle \alpha, \mathrm{ad}_v w\rangle,$$

for any $w \in V$ and $\alpha \in V^*$.

Let $v \in V$, both operators admit a matrix representation, denoted with $[\mathrm{ad}_v]$ and $[\mathrm{ad}_v^*]$, respectively.

Appendix B

Stochastic Processes

B.1 Brownian Motion

A stochastic process in continuous time [37, 88, 109] can be viewed as a sequence of random variables $\{X_t\}_{t\geq 0}$ indexed with a real parameter $t \in \mathbb{R}_{\geq 0}$, representing the time. A stochastic process takes value in a state space usually denoted with Ω. The space state is equipped with a sigma-algebra Ψ.

Definition B.1.0.1. ***(Sigma-algebra)*** *A sigma-algebra Ψ is a collection of subsets of Ω that satisfies the following properties*

1. *Ω itself belongs to Ψ,*
2. *if a set $A \in \Omega$ belongs to Ψ, also its complementary set A^c belongs to Ψ,*
3. *if $\{A_n\}_{n\in\mathbb{N}}$ is a family of sets in Ψ, then*

$$\bigcup_{n\in\mathbb{N}} A_i \in \Psi. \tag{B.1}$$

Property (B.1) is also known as *stability for countable unions* of sets, that in turn implies stability for countable intersections, thanks to the properties of set operations. Indeed, we have

$$\bigcap_{n\in\mathbb{N}} A_i = \left(\bigcup_{n\in\mathbb{N}} A_i^c\right)^c \in \Psi.$$

The space Ω endowed with a sigma-algebra Ψ is a measurable state space and the elements within Ψ are measurable sets. Consider the state $X = X_t$ of a stochastic process. All the information available until time t is represented by its *Filtration* $\{\mathcal{F}\}_{t\geq 0}$.

Definition B.1.0.2. ***(Filtration)*** *The filtration $\{\mathcal{F}\}_{t\geq 0}$ of a stochastic process X_t is an increasing sequence of sigma-algebras defined on Ω such that*

1. *for every $s < t$, $\mathcal{F}_s \subset \mathcal{F}_t$ and*

2. $\mathcal{F}_t = \bigcap_{u<t} \mathcal{F}_u$.

Definition B.1.0.3. ***(Martingale)*** *A stochastic process X_t is a martingale when it satisfies the following properties*

1. *X_t is $\mathcal{F}_t$-measurable,*

2. *$\mathbb{E}[X_t] < +\infty$ for every $t \in \mathbb{R}_{\geq 0}$,*

3. *$\mathbb{E}[X_t|\mathcal{F}_s] = X_s$.*

If $\mathbb{E}[X_t|\mathcal{F}_s] \geq X_s$ then the stochastic process is a *submartingale*, while if $\mathbb{E}[X_t|\mathcal{F}_s] \leq X_s$ then it is a *supermartingale*. $\mathcal{F}_t$-measurability means that the counterimage of Borel sets through X belongs to the sigma-algebra Ψ. The Borel sets are the elements of the so called "Borel sigma-algebra" $\mathcal{B}$, that is the smallest sigma-algebra that one can put on a topological space compatible with the topology itself, in the sense that $\mathcal{B}$ contains all the sets of the topology. The 4-uple $(\Omega, \Psi, \{\mathcal{F}\}_{t\geq 0}, \mathbb{P})$ is called *filtered probability space*, where $\mathbb{P}$ is the probability measure[1] on Ω.

Definition B.1.0.4. ***(Brownian motion)*** *The Brownian motion B_t, also known as Wiener process and denoted with W_t, is a continuous-time Gaussian stochastic process: $B_t \sim N(0, t)$ for every $t \in \mathbb{R}_{\geq 0}$.*

In other words, a Brownian motion is a sequence of independent Gaussian variables with mean equal to 0 and variance equal to t.

Proposition B.1.0.1. *The Brownian motion satisfies the following properties*

1. *$B_0 = 0$, i.e. the process always starts from zero,*

2. *it has independent increments: $B_s \perp B_t$ for $s < t$,*

3. *$B_{t-s} \sim N(0, t-s)$ for $s < t$,*

4. *the covariance function is defined by $a(s,t) = min(s,t)$ for $s, t \in \mathbb{R}_{\geq 0}$,*

5. *it is a Markov process,*

6. *it is a Martingale,*

7. *the trajectories have unbounded variations,*

8. *$[B_t] = t$: the quadratic variation is equal to t.*

Proof. See [88]. □

[1] A map from Ω to $[0, 1]$ that associated a probability to every element of the state space.

B.2 Stochastic Differential Equations

A *Stochastic Differential Equation* (SDE) is a differential equation containing one or more terms that are stochastic processes. Various phenomena whose evolution is not deterministic, such as stock prices and the evolution of interest rates, are modeled using stochastic differential equations. We will focus our attention on linear SDEs with Brownian noise, however other types of random behaviour are possible, such as jump processes [13].

Definition B.2.0.1. ***(Linear SDE)*** *Let $(\Omega, \Psi, \{\mathcal{F}\}_{t\geq 0}, \mathbb{P})$ be a filtered probability space. Let $t \in \mathbb{R}_{\geq 0}$ and $x \in \mathbb{R}$ be two variables, where in particular t represents time. A (one-dimensional) linear SDE in differential form is expressed as follows*

$$\begin{cases} dX_t = a(t, X_t)dt + b(t, X_t)dB_t \\ X_0 = x_0 \end{cases} \qquad t \in \mathbb{R}_{\geq 0}, \tag{B.2}$$

where $a : \mathbb{R}_{\geq 0} \times \mathbb{R} \to \mathbb{R}$ and $b : \mathbb{R}_{\geq 0} \times \mathbb{R} \to \mathbb{R}$ are non-random well defined real-valued functions called "drift" and "diffusion" coefficients and B_t is a (standard) $\mathcal{F}_t$-adapted Brownian motion. Equation (B.2) *can be expressed in integral form as follows*

$$X_t = X_0 + \int_0^t a(s, X_s)ds + \int_0^t b(s, X_s)dB_s \quad t \in \mathbb{R}_{\geq 0}. \tag{B.3}$$

The solution of an SDE is also a stochastic process. A stochastic process X_t solves (B.2), or equivalently (B.3), if the processes

$$X_t \text{ and } \xi_t \triangleq X_0 + \int_0^t a(s, X_s)ds + \int_0^t b(s, X_s)dB_s$$

are *indistinguishable*, i.e., defined over the same filtered probability space and such that

$$\mathbb{P}\left[X_t = \xi_t\right] = 1 \text{ for every } t \in \mathbb{R}_{\geq 0}.$$

There exists a vast amount of literature about existence and uniqueness of solutions to stochastic differential equations, e.g., [6, 37, 78, 88, 98], whose first result is due to Itô [66]. Usually one requires at least the Lipschitz-continuity of drift and diffusion coefficients, but it is a sufficient but not necessary condition: it is possible to prove existence and uniqueness of solutions also in case of weaker assumptions on the coefficients.

We remark that the existence of a closed-form solution is not always guaranteed; in this case in order to calculate one it is appropriate to rely on approximation schemes, e.g., Euler-Maruyama method [63], or quadrature formulas.

Lemma B.2.0.1. ***(Itô-Doeblin formula)*** *Let $(\Omega, \Psi, \{\mathcal{F}\}_{t\geq 0}, \mathbb{P})$ be a filtered probability space. Let $t \in \mathbb{R}_{\geq 0}$ and $x \in \mathbb{R}$ be two variables, where in particular t represents time. Consider a non-random function $F(t, x) \in C^{1,2}(\mathbb{R}_{\geq 0} \times \mathbb{R})$ and a SDE of the form* (B.3)*.*

The (one-dimensional) Itô-Doeblin formula in integral form is given by

$$F(t, X_t) = F_0 + \int_0^t \left[\frac{\partial F(s, X_s)}{\partial s} + \frac{1}{2} \frac{\partial^2 F(s, X_s)}{\partial X_s^2} b^2(s, X_s) \right] ds + \int_0^t \frac{\partial F(s, X_s)}{\partial X_s} dX_s$$

where $F_0 = F(0, X_0)$*. It can be also expressed in differential form as follows*

$$dF(t, X_t) = \left[\frac{\partial F(t, X_t)}{\partial t} + \frac{1}{2} \frac{\partial^2 F(t, X_t)}{\partial X_t^2} b^2(t, X_t) \right] dt + \frac{\partial F(t, X_t)}{\partial X_t} dX_t.$$

Proof. See [88]. □

The Itô-Doeblin formula, also known as Itô's Lemma, provides the basic rules by which the differential of the composition of functions, that depends both on deterministic and stochastic variables, should be calculated. In other words, the Itô-Doeblin formula is the *stochastic equivalent of the chain rule* in traditional calculus. It is usually referred as Itô formula for simplicity although it has been proved that the first man that has ever derived this formula was a French-German mathematician named Wolfgang Doeblin (1915-1940).

We conclude by pointing out that both Definition B.2.0.1 and Lemma B.2.0.1 hold true also in the multi-dimensional case, where drift and diffusion coefficients are vector-valued and matrix-valued functions, respectively, and B_t is an n-dimensional Brownian motion.

Bibliography

[1] J. Alonso-Mora, S. Baker, and D. Rus. "*Multi-robot Formation Control and Object Transport in Dynamic Environments via Constrained Optimization*". In: *The International Journal of Robotics Research* 36.9 (2017), pp. 1000–1021.

[2] J. Alonso-Mora et al. "*Optimal Reciprocal Collision Avoidance for multiple Non-holonomic Robots*". In: *Distributed autonomous robotic systems*. Springer, 2013, pp. 203–216.

[3] R. Andreani et al. "*On augmented Lagrangian methods with General Lower-level Constraints*". In: *SIAM Journal on Optimization* 18.4 (2008), pp. 1286–1309.

[4] L. Angel and J. Viola. "*Parametric Identification of a Delta Type Parallel Robot*". In: *2016 IEEE Colombian Conference on Robotics and Automation (CCRA)*. IEEE. 2016, pp. 1–6.

[5] L. Angel et al. "*RoboTenis System part II: Dynamics and Control*". In: *Proceedings of the 44th IEEE Conference on Decision and Control*. IEEE. 2005, pp. 2030–2034.

[6] D. Applebaum. *Lévy Processes and Stochastic Calculus*. Cambridge university press, 2009.

[7] F. Asadi and A. Heydari. "*Analytical Dynamic Modeling of Delta Robot with Experimental Verification*". In: *Proceedings of the Institution of Mechanical Engineers, Part K: Journal of Multi-body Dynamics* 234.3 (2020), pp. 623–630.

[8] K. J. Åström and B. Wittenmark. *Computer-controlled Systems: Theory and Design*. Courier Corporation, 2013.

[9] F. Aurenhammer. "*Voronoi Diagrams: A Survey of a Fundamental Geometric Data Structure*". In: *ACM Computing Surveys (CSUR)* 23.3 (1991), pp. 345–405.

[10] T. Balch and R. C. Arkin. "*Behavior-Based Formation Control for Multirobot Teams*". In: *IEEE transactions on robotics and automation* 14.6 (1998), pp. 926–939.

[11] D. Bareiss and J. Van den Berg. "*Generalized Reciprocal Collision Avoidance*". In: *The International Journal of Robotics Research* 34.12 (2015), pp. 1501–1514.

[12] J. Barraquand and J.C. Latombe. "*Robot Motion Planning: A Distributed Representation Approach*". In: *The International Journal of Robotics Research* 10.6 (1991), pp. 628–649.

[13] R.F. Bass. "*Stochastic differential equations with jumps*". In: *Probability Surveys* 1 (2004), pp. 1–19.

[14] T. Battisti and R. Muradore. "*A Velocity Obstacles approach for Autonomous Landing and Teleoperated Robots*". In: *Autonomous Robots* 44.2 (2020), pp. 217–232.

[15] J. S. Bellingham et al. "*Cooperative Path-Planning for multiple UAVs in Dynamic and Uncertain Environments*". In: *Proceedings of the 41st IEEE Conference on Decision and Control, 2002.* Vol. 3. IEEE. 2002, pp. 2816–2822.

[16] I.A. Bizyaev, A.V. Borisov, and S.P. Kuznetsov. "*The Chaplygin Sleigh with Friction Moving due to Periodic Oscillations of an Internal Mass*". In: *Nonlinear Dynamics* 95.1 (2019), pp. 699–714.

[17] A. Bloch. *Nonholonomic Mechanics and Control*. Springer Science & Business Media, 2003.

[18] A.M. Bloch. "*Nonholonomic Mechanics*". In: *Nonholonomic Mechanics and Control*. Springer, 2003, pp. 207–276.

[19] A.V. Borisov and I.S. Mamayev. "*The Dynamics of a Chaplygin Sleigh*". In: *Journal of Applied Mathematics and Mechanics* 73.2 (2009), pp. 156–161.

[20] S Bortoff. "*Object-Oriented Modeling and Control of Delta Robots*". In: *2018 IEEE Conference on Control Technology and Applications (CCTA)*. IEEE. 2018, pp. 251–258.

[21] O. Brock and O. Khatib. "*High-speed navigation using the global dynamic window approach.*" In: 1 (1999), pp. 341–346.

[22] F. Bullo and A. Lewis. *Geometric Control of Mechanical Systems: Modeling, Analysis, and Design for Simple Mechanical Control Systems*. Vol. 49. Springer, 2019.

[23] G. Campion, G. Bastin, and B. D'Andréa-Novel. "*Structural Properties and Classification on Kinematic and Dynamic Models of Wheeled Mobile Robots*". In: *Russian Journal of Nonlinear Dynamics* 7.4 (2011), pp. 733–769.

[24] L. A. Castañeda, A. Luviano-Juárez, and I. Chairez. "*Robust Trajectory Tracking of a Delta Robot through Adaptive Active Disturbance Rejection Control*". In: *IEEE Transactions on Control Systems Technology* 23.4 (2014), pp. 1387–1398.

[25] A. Cavallo et al. "*Modeling and Slipping Control of a Planar Slider*". In: *Automatica* 115 (2020), p. 108875.

[26] J. Chen and D. Sun. "*Resource Constrained Multirobot Task Allocation based on Leader-Follower Coalition Methodology*". In: *The International Journal of Robotics Research* 30.12 (2011), pp. 1423–1434.

[27] J. Chen et al. "*Leader-Follower Formation Control of Multiple Nonholonomic Mobile Robots incorporating a Receding-horizon Scheme*". In: *The International Journal of Robotics Research* 29.6 (2010), pp. 727–747.

[28] GS. Chirikjian. *Stochastic Models, Information Theory and Lie Groups (Vol.1)*. Springer Science & Business Media, 2009.

[29] GS. Chirikjian. *Stochastic Models, Information Theory and Lie Groups (Vol.2)*. Springer Science & Business Media, 2009.

[30] Reymond Clavel. *Dispositif pour le dÉplacement et le Positionnement d'un Élément dans l'Espace*. 1987.

[31] A. Codourey. "*Dynamic modelling and mass matrix evaluation of the DELTA parallel robot for axes decoupling control*". In: *Proceedings of IEEE/RSJ international conference on intelligent robots and systems. IROS'96*. Vol. 3. IEEE. 1996, pp. 1211–1218.

[32] A. Codourey. "*Dynamic Modeling of Parallel Robots for Computed-torque Control Implementation*". In: *The International Journal of Robotics Research* 17.12 (1998), pp. 1325–1336.

[33] A. Codourey and E. Burdet. "*A body-oriented method for finding a linear form of the dynamic equation of fully parallel robots*". In: *Proceedings of international conference on robotics and automation*. Vol. 2. IEEE. 1997, pp. 1612–1618.

[34] R.C. Coulter. *Implementation of the Pure Pursuit Path Tracking Algorithm*. Tech. rep. Carnegie-Mellon UNIV Pittsburgh PA Robotics INST, 1992.

[35] C. De La Cruz and R. Carelli. "*Dynamic Model based Formation Control and Obstacle Avoidance of Multi-robot Systems*". In: *Robotica* 26.3 (2008), pp. 345–356.

[36] R. Dhaouadi and A. Hatab. "*Dynamic Modelling of Differential-Drive Mobile Robots using Lagrange and Newton-Euler Methodologies: A Unified Framework*". In: *Advances in Robotics & Automation* 2.2 (2013), pp. 1–7.

[37] L.C. Evans. *An Introduction to Stochastic Differential Equations*. Vol. 82. American Mathematical Soc., 2012.

[38] F. Falezza et al. "*Gray-Box Model Identification and Payload Estimation for Delta Robots*". In: *2020 IFAC World Congress-PapersOnLine* 53.2 (2020), pp. 8771–8776.

[39] S. Fang and T. Zhang. "*Stochastic Differential Equations with non-Lipschitz Coefficients: I. Pathwise Uniqueness and Large Deviations*". In: *arXiv preprint math/0311032* (2003).

[40] S. Fang and T. Zhang. "*Stochastic Differential Equations with non-Lipschitz Coefficients: II. Dependence with Respect to Initial Values*". In: *arXiv preprint math/0311034* (2003).

[41] S. Farsoni et al. "*Real-time Identification of Robot Payload using a Multirate Quaternion-Based Kalman Filter and Recursive Total Least-Squares*". In: *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2018, pp. 2103–2109.

[42] P. Fiorini and Z. Shiller. "*Motion Planning in Dynamic Environments using the Relative Velocity Paradigm*". In: *[1993] Proceedings IEEE International Conference on Robotics and Automation*. IEEE. 1993, pp. 560–565.

[43] P. Fiorini and Z. Shiller. "*Robot Motion Planning in Dynamic Environments*". In: *Robotics Research* (1996), pp. 237–248.

[44] P. Fiorini and Z. Shiller. "*Motion Planning in Dynamic Environments using Velocity Obstacles*". In: *The International Journal of Robotics Research* 17.7 (1998), pp. 760–772.

[45] M. Foskey et al. "*A Voronoi-based Hybrid Motion Planner*". In: *Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the the Next Millennium (Cat. No. 01CH37180)*. Vol. 1. IEEE. 2001, pp. 55–60.

[46] D. Fox, W. Burgard, and S. Thrun. "*The Dynamic Window Approach to Collision Avoidance*". In: *IEEE Robotics & Automation Magazine* 4.1 (1997), pp. 23–33.

[47] G.F. Franklin, J.D. Powell, and M.L. Workman. *Digital Control of Dynamic Systems*. Vol. 3. Addison-wesley Reading, MA, 1998.

[48] A. Fumagalli and P. Masarati. "*Real-time Inverse Dynamics Control of Parallel Manipulators using General-Purpose Multibody Software*". In: *Multibody System Dynamics* 22.1 (2009), pp. 47–68.

[49] S. Sam Ge and Y. J. Cui. "*Dynamic Motion Planning for Mobile Robots using Potential Field method*". In: *Autonomous robots* 13.3 (2002), pp. 207–222.

[50] A. Gholipour and M.J. Yazdanpanah. "*Dynamic Tracking Control of Nonholonomic Mobile Robot with Model Reference Adaptation for Uncertain Parameters*". In: *2003 European Control Conference (ECC)*. IEEE. 2003, pp. 3118–3122.

[51] M. Ginesi, N. Sansonetto, and P. Fiorini. "*Overcoming some Drawbacks of Dynamic Movement Primitives*". In: *Robotics and Autonomous Systems* 144 (2021), p. 103844.

[52] M. Ginesi et al. "*Dynamic Movement Primitives: Volumetric Obstacle Avoidance*". In: *2019 19th International Conference on Advanced Robotics (ICAR)*. IEEE. 2019, pp. 234–239.

[53] M. Ginesi et al. "*Dynamic Movement Primitives: Volumetric Obstacle Avoidance using Dynamic Potential Functions*". In: *Journal of Intelligent & Robotic Systems* 101.4 (2021), pp. 1–20.

[54] R. Goebel, R. Sanfelice, and A. Teel. *Hybrid Dynamical Systems*. Princeton University Press, 2012.

[55] J. V. Gómez et al. "Planning Robot formations with Fast Marching Square including Uncertainty Conditions". In: *Robotics and Autonomous Systems* 61.2 (2013), pp. 137–152.

[56] S. Goyal, A. Ruina, and J. Papadopoulos. "*Planar Sliding with Dry Friction, Part 1: Limit Surface and Moment Function*". In: *Wear* 143.2 (1991), pp. 307–330.

[57] S. Goyal, A. Ruina, and J. Papadopoulos. "*Planar Sliding with Dry Friction, Part 2. Dynamics of motion*". In: *Wear* 143.2 (1991), pp. 331–352.

[58] P. Graczyk and J. Małecki. "*Multidimensional Yamada-Watanabe Theorem and its Applications to Particle Systems*". In: *Journal of Mathematical Physics* 54.2 (2013), p. 021503.

[59] S. Guy et al. "*Clearpath: Highly Parallel Collision Avoidance for Multi-Agent Simulation*". In: *Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2009), pp. 177–187.

[60] H. Hahn. *Dynamics of Planar and Spatial Rigid-Body Systems*. Springer Verlag GmbH, 2002.

[61] B. Hall. *Lie Groups, Lie Algebras, and Representations: An Elementary Introduction*. Vol. 10. Springer, 2003.

[62] D. Helbing and P. Molnar. "*Social Force Model for Pedestrian Dynamics*". In: *Physical review E* 51.5 (1995), p. 4282.

[63] D.J. Higham. "*An Algorithmic Introduction to Numerical Simulation of Stochastic Differential Equations*". In: *SIAM review* 43.3 (2001), pp. 525–546.

[64] T. Holm D. Schmah and C. Stoica. *Geometric Mechanics and Symmetry: from Finite to Infinite Dimensions*. Vol. 12. Oxford University Press, 2009.

[65] S. Hu and D. Sun. "*Automatic Transportation of Biological Cells with a Robot-Tweezer Manipulation System*". In: *The International Journal of Robotics Research* 30.14 (2011), pp. 1681–1694.

[66] K. Itô. "*On a Stochastic Integral Equation*". In: *Proceedings of the Japan Academy* 22.1-4 (1946), pp. 32–35.

[67] H. Jiang et al. "*Obstacle Avoidance of Autonomous Vehicles with CQP-based Model Predictive Control*". In: *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE. 2016, pp. 001668–001673.

[68] T. Keviczky, F. Borrelli, and G. J. Balas. "*A Study on Decentralized Receding Horizon Control for Decoupled Systems*". In: *Proceedings of the 2004 American Control Conference*. Vol. 6. IEEE. 2004, pp. 4921–4926.

[69] Z.A. Khan, V. Chacko, and H. Nazir. "*A Review of Friction Models in Interacting Joints for Durability Design*". In: *Friction* 5.1 (2017), pp. 1–22.

[70] O. Khatib. "*Real-time Obstacle Avoidance for Manipulators and Mobile Robots*". In: *Autonomous robot vehicles*. Springer, 1986, pp. 396–404.

[71] M. Kim and J. Oh. "*Study on Optimal Velocity selection using Velocity Obstacle (OVVO) in Dynamic and Crowded Environment*". In: *Autonomous Robots* 40.8 (2016), pp. 1459–1470.

[72] D.E. Koditschek. "*Robot Planning and Control via Potential Functions*". In: *The robotics review* (1989), p. 349.

[73] S. Konduri, E.O.C. Torres, and P.R. Pagilla. "*Dynamics and Control of a Differential Drive Robot with Wheel Slip: Application to Coordination of Multiple Robots*". In: *Journal of Dynamic Systems, Measurement, and Control* 139.1 (2017).

[74] Y. Kuo. "*Mathematical Modeling and Analysis of the Delta Robot with Flexible Links*". In: *Computers & Mathematics with Applications* 71.10 (2016), pp. 1973–1989.

[75] Y. Kuo and P. Huang. "*Experimental and Simulation Studies of Motion Control of a Delta Robot using a Model-Based Approach*". In: *International Journal of Advanced Robotic Systems* 14.6 (2017), pp. 1–14.

[76] M. A. Lewis and K.H. Tan. "*High precision Formation Control of Mobile Robots using Virtual Structures*". In: *Autonomous robots* 4.4 (1997), pp. 387–403.

[77] D.J.N. Limebeer and M. Massaro. *Dynamics and Optimal Control of Road Vehicles*. Oxford University Press, 2018.

[78] P.L. Lions and A.S. Sznitman. "*Stochastic Differential Equations with Reflecting Boundary Conditions*". In: *Communications on Pure and Applied Mathematics* 37.4 (1984), pp. 511–537.

[79] Y. Liu and R. Bucknall. "*A Survey of Formation Control and Motion Planning of Multiple Unmanned Vehicles*". In: *Robotica* 36.7 (2018), pp. 1019–1047.

[80] T. Lozano-Perez. "*Spatial Planning: A Configuration Space Approach*". In: *Autonomous robot vehicles*. Springer, 1990, pp. 259–271.

[81] R. Luna and K. E. Bekris. "*Efficient and Complete Centralized Multi-robot Path Planning*". In: *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE. 2011, pp. 3268–3275.

[82] D. Q. Mayne. "*Control of Constrained Dynamic Systems*". In: *European Journal of Control* 7.2-3 (2001), pp. 87–99.

[83] N. Michael, J. Fink, and V. Kumar. "*Cooperative Manipulation and Transportation with Aerial Robots*". In: *Autonomous Robots* 30.1 (2011), pp. 73–86.

[84] K. Miller. "*Experimental Verification of Modeling of DELTA Robot Dynamics by Direct Application of Hamilton's Principle*". In: *Proceedings of 1995 IEEE international conference on robotics and automation*. Vol. 1. IEEE. 1995, pp. 532–537.

[85] J.I. Neimark and N. A. Fufaev. *Dynamics of Nonholonomic Systems*. Vol. 33. American Mathematical Soc., 2004.

[86] P. Ogren and N.E. Leonard. "*A Convergent Dynamic Window approach to Obstacle Avoidance.*" In: *IEEE Transactions on Robotics* 21.2 (2005), pp. 188–195.

[87] K.K. Oh, M.C. Park, and H.S. Ahn. "*A Survey of Multi-Agent Formation Control: Position, Displacement and Distance-based approaches*". In: *Number: Gist DCASL TR* 2 (2012).

[88] B. Oksendal. *Stochastic Differential Equations: An Introduction with Applications*. Springer Science & Business Media, 2013.

[89] G. Oriolo, A. De Luca, and M. Vendittelli. "*WMR Control via Dynamic Feedback Linearization: Design, Implementation, and Experimental Validation*". In: *IEEE Transactions on control systems technology* 10.6 (2002), pp. 835–852.

[90] J.M. Park et al. "*Obstacle Avoidance of Autonomous Vehicles based on Model Predictive Control*". In: *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering* 223.12 (2009), pp. 1499–1516.

[91] S.B. Park et al. "*Dynamics Modeling of a Delta-type Parallel Robot*". In: *IEEE ISR 2013*. IEEE. 2013, pp. 1–5.

[92] Z. Peng et al. "*Leader-Follower Formation Control of Non-holonomic Mobile Robots based on a Bioinspired Neurodynamic based approach*". In: *Robotics and autonomous systems* 61.9 (2013), pp. 988–996.

[93] P. Petrov. "*Modeling and Adaptive Path Control of a Differential Drive Mobile* Robot". In: *Proceedings of the 12th WSEAS international conference on Automatic control, modelling & simulation*. Vol. 6. World Scientific, Engineering Academy, and Society (WSEAS). 2010, pp. 403–408.

[94] N. Piccinelli and R. Muradore. "*Hybrid Motion Planner Integrating Global Voronoi Diagrams and Local Velocity Obstacle Method*". In: *2018 European Control Conference (ECC)*. IEEE. 2018, pp. 26–31.

[95] N. Piccinelli, F. Vesentini, and R. Muradore. "*Planning with Real-Time Collision Avoidance for Cooperating Agents under Rigid Body Constraints*". In: *2019 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE. 2019, pp. 1261–1264.

[96] F. Pierrot, P. Dauchez, and A. Fournier. "*HEXA: A Fast Six-DoF Fully-Parallel Robot*". In: *Fifth International Conference on Advanced Robotics' Robots in Unstructured Environments*. IEEE. 1991, pp. 1158–1163.

[97] F. Pierrot et al. "*Optimal Design of a 4-DOF Parallel Manipulator: From Academia to Industry*". In: *IEEE Transactions on Robotics* 25.2 (2009), pp. 213–224.

[98] A. Pilipenko. *An Introduction to Stochastic Differential Equations with Reflection*. Vol. 1. Universitätsverlag Potsdam, 2014.

[99] W.L.V. Pollard. "*Position-Controlling Apparatus*". In: *United States Patent Office* (1938), pp. 1–5.

[100] A. Poty, P. Melchior, and A. Oustaloup. "*Dynamic Path Planning by Fractional Potential*". In: *Second IEEE International Conference on Computational Cybernetics, 2004. ICCC 2004.* IEEE. 2004, pp. 365–371.

[101] C. Qixin, H. Yanwen, and Z. Jingliang. "*An Evolutionary Artificial Potential Field algorithm for Dynamic Path Planning of Mobile Robot*". In: *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE. 2006, pp. 3331–3336.

[102] A. Rao. *Dynamics of Particles and Rigid Bodies: A Systematic Approach*. Cambridge University Press, 2006.

[103] W. Rossmann. *Lie Groups: An Introduction through Linear Groups*. Vol. 5. Oxford University Press on Demand, 2006.

[104] V.V. Rumyantsev. "*Variational principles for systems with unilateral constraints*". In: *Journal of applied mathematics and mechanics* 70.6 (2006), pp. 808–818.

[105] A. E. Scheflen and N. Ashcraft. "*Human Territories: How we behave in Space-Time.*" In: (1976).

[106] G. Sharon et al. "*Conflict-based Search for Optimal Multi-agent Pathfinding*". In: *Artificial Intelligence* 219 (2015), pp. 40–66.

[107] Z. Shiller, O. Gal, and T. Fraichard. "*The Nonlinear Velocity Obstacle revisited: The Optimal Time Horizon*". In: *Guaranteeing safe navigation in dynamic environments workshop*. 2010.

[108] Y. Shoukry et al. "*SMC: Satisfiability Modulo Convex Optimization*". In: *Proceedings of the 20th International Conference on Hybrid Systems: Computation and Control*. 2017, pp. 19–28.

[109] S. E. Shreve. *Stochastic Calculus for Finance II: Continuous-time Models*. Vol. 11. Springer Science & Business Media, 2004.

[110] B. Siciliano et al. "*Robotics: Modelling, Planning and Control*". In: *Advanced Textbooks in Control and Signal Processing Springer* (2009).

[111] J. Snape and D. Manocha. "*Goal Velocity Obstacles for Spatial Navigation of Multiple Autonomous Robots or Virtual Agents*". In: *Autonomous Robots and Multi robot Systems, St. Paul, Minn* (2013), pp. 1–17.

[112] J. Snape et al. "*The Hybrid Reciprocal Velocity Obstacle*". In: *IEEE Transactions on Robotics* 27.4 (2011), pp. 696–706.

[113] A. Sozzi et al. "*Dynamic Motion Planning for Autonomous Assistive Surgical Robots*". In: *Electronics* 8.9 (2019), p. 957.

[114] S. Staicu. "*Relations matricielles de récurrence en dynamique des mécanismes*". In: *Revue Roumaine des Sciences Techniques-Série de Mécanique Appliquée* 50.1-3 (2005), pp. 15–28.

[115] S. Staicu. "*Recursive Modelling in Dynamics of Delta Parallel Robot*". In: *Robotica* 27.2 (2009), pp. 199–207.

[116] R.E. Stamper. *A Three Degree of Freedom Parallel Manipulator with only Translational Degrees of Freedom*. University of Maryland, College Park, 1997.

[117] Doug Stewart. "*A Platform with Six Degrees of Freedom*". In: *Proceedings of the institution of mechanical engineers* 180.1 (1965), pp. 371–386.

[118] P. Stone and M. Veloso. "*Task Decomposition, Dynamic Role Assignment, and low-bandwidth Communication for Real-Time Strategic Teamwork*". In: *Artificial Intelligence* 110.2 (1999), pp. 241–273.

[119] D. Sun et al. "*A Synchronization approach to Trajectory Tracking of Multiple Mobile Robots while maintaining Time-varying Formations*". In: *IEEE Transactions on Robotics* 25.5 (2009), pp. 1074–1086.

[120] O. Takahashi and R. J. Schilling. "*Motion Planning in a Plane using Generalized Voronoi diagrams*". In: *IEEE Transactions on robotics and automation* 5.2 (1989), pp. 143–150.

[121] S. Tang, J. Thomas, and V. Kumar. "*Hold or Take Optimal Plan (Hoop): A Quadratic Programming approach to Multi-robot Trajectory Generation*". In: *The International Journal of Robotics Research* 37.9 (2018), pp. 1062–1084.

[122] L. Tsai. *Robot Analysis: the Mechanics of Serial and Parallel Manipulators*. John Wiley & Sons, 1999.

[123] L.W. Tsai and R. Stamper. "*A Parallel Manipulator with only Translational Degrees of Freedom*". In: *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. Vol. 97584. American Society of Mechanical Engineers. 1996, pp. 1–18.

[124] M. Turpin, N. Michael, and V. Kumar. "*Capt: Concurrent Assignment and Planning of Trajectories for Multiple Robots*". In: *The International Journal of Robotics Research* 33.1 (2014), pp. 98–112.

[125] J. Van den Berg, M. Lin, and D. Manocha. "*Reciprocal Velocity Obstacles for Real-Time Multi-Agent Navigation*". In: *2008 IEEE International Conference on Robotics and Automation*. IEEE. 2008, pp. 1928–1935.

[126] J. Van Den Berg et al. "*Reciprocal Collision Avoidance with Acceleration-Velocity Obstacles*". In: *2011 IEEE International Conference on Robotics and Automation*. IEEE. 2011, pp. 3475–3482.

[127] J. Van Den Berg et al. "*Reciprocal n-body Collision Avoidance*". In: *Robotics research*. Springer, 2011, pp. 3–19.

[128] P.K.C. Wang. "*Navigation Strategies for multiple Autonomous Mobile Robots moving in formation*". In: *Journal of Robotic Systems* 8.2 (1991), pp. 177–195.

[129] D. Watson. "*Spatial Tessellations: Concepts and Applications of Voronoi Diagrams.*" In: *Computers and Geosciences* 19.8 (1993), pp. 1209–1210.

[130] D. Wilkie, J. Van Den Berg, and D. Manocha. "*Generalized Velocity Obstacles*". In: *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE. 2009, pp. 5573–5578.

[131] J. Wit, C.D. Crane III, and D. Armstrong. "*Autonomous Ground Vehicle Path Tracking*". In: *Journal of Robotic Systems* 21.8 (2004), pp. 439–449.

[132] J. Wu et al. "*Dynamics and Control of a Planar 3-DOF Parallel Manipulator with Actuation Redundancy*". In: *Mechanism and Machine Theory* 44.4 (2009), pp. 835–849.

[133] J. Wu et al. "*Mechatronics Modeling and Vibration Analysis of a 2-DOF Parallel Manipulator in a 5-DOF Hybrid Machine Tool*". In: *Mechanism and Machine Theory* 121 (2018), pp. 430–445.

[134] J. Wu et al. "*An Iterative Learning Method for realizing Accurate Dynamic Feedforward Control of an Industrial Hybrid Robot*". In: *Science China Technological Sciences* (2021), pp. 1–12.

[135] T. Yamada and S. Watanabe. "*On the Uniqueness of Solutions of Stochastic Differential Equations*". In: *Journal of Mathematics of Kyoto University* 11.1 (1971), pp. 155–167.

[136] J. Yu and S. M. LaValle. "*Optimal Multirobot Path Planning on Graphs: Complete Algorithms and Effective Heuristics*". In: *IEEE Transactions on Robotics* 32.5 (2016), pp. 1163–1177.

[137] X. Yun and Y. Yamamoto. "*Stability Analysis of the Internal Dynamics of a Wheeled Mobile Robot*". In: *Journal of Robotic Systems* 14.10 (1997), pp. 697–709.

[138] Y. Zhou and G. S. Chirikjian. "*Probabilistic Models of Dead-Reckoning error in Nonholonomic Mobile Robots*". In: *2003 IEEE International Conference on Robotics and Automation (Cat. No. 03CH37422)*. Vol. 2. IEEE. 2003, pp. 1594–1599.