UNIVERSITAT POLITÈCNICA DE CATALUNYA

Doctoral Programme:

Doctorado en Ingeniería Mecánica, Fluídos y Aeronáutica

Research Plan:

Kinodynamic Planning and Control of Agile and Graceful Robot Motions

Siro Moreno

Advisors:

Enric Celaya, Dr.

Lluís Ros, Dr.

December, 2020

Contents

1 Introduction and Context

Agility and gracefulness are interesting notions in robotics. They are vague and lack a consistent technical definition, yet they represent objectives towards which a lot of effort is being invested by several researchers and institutions.

Usually, when we search for agility in the literature, we find definitions oriented towards rapid movement and accelerations. These definitions and metrics have a wide range of complexity, from very simple metrics to complex benchmarks. For example, in their 2009 paper [15], Bowling and Teng review some definitions and conclude that good metrics for agility are the ability to accelerate and the time to achieve full speed. In their 2016 work [24], Duperret, Kenneally, Pusey and Koditschek take a different approach and arrive at an interesting metric designed to compare the performance of legged robots, defining the *specific agility* in terms of mass-normalized change in extrinsic body energy. In their biomimetic development of a jumping robot [33], Haldane, Plecnik, Yim, and Fearing arrive to a metric for vertical jumping agility based on a repetitive jumping motion, by multiplying the height and frequency of the jumps. On the complex end of the formulation range, we find works such as that [25] of Eckert and Ijspeert, which approach the definition from a wide perspective and develop a benchmark for comparing legged robots based on multiple characteristics that they consider important in the definition of agility, such as speed, precision and energy cost in the performing various tasks. Our own definition for agility will be most similar to that last one, consisting in an umbrella term under which we will condense some different objectives. Since in our thesis we want to optimize trajectories and will not enter the robot design phase, our objective with agility is to translate its meaning into movement variables that we can measure and control. For example, we will understand that an aspect of agility is the capacity to perform accelerations in any direction of the task space. Through the use of models of the electric actuators, we will calculate acceleration and speed envelopes as functions of configuration and speed. Then, we will use them as a factor in the optimization cost function, penalizing trajectories that get too close to the limits of the robot capabilities. That will result in a trajectory that travels the task space through configurations that tend to keep a maximum implicit agility, in the sense that at any given point of the trajectory, the robot will have the ability to react with agility to sudden changes in the environment, like moving obstacles.

When we talk about graceful robots, we usually describe some quality in their movements: they are smooth, intuitive and beautiful, which are again abstract terms that anyone can interpret to their liking. This concept is used by authors such as Gulati, Kuipers and Park to describe the optimization objective they want to achieve in their works with robotic wheelchairs [31] [32] [52]. They usually define graceful movement in terms of being safe, smooth, comfortable, fast, and intuitive. They use these characteristics because they are all very relevant for the comfort of the person using the wheelchair. With their balancing column-like robot, Nagarajan, Kantor, and Hollis arrive to similar results using a more concrete definition for graceful movement: when its configuration variables' position, velocity and acceleration trajectories are continuous and bounded with low jerk [51] [50]. In that sense, it is also similar to the regularization term for smoothness and naturalness that Zhao, Lin and Tomizuka add to the cost function in their 2018 paper [61], which is the time integral of the jerk squared. In our thesis, an important objective will be to explore the different variables that we can use in order to find a concrete definition of gracefulness. We will use a multi-factorial approach that will include several concrete variables, such as the aforementioned jerk, actuator load or energy efficiency, as we will see in the next section 2.

Motion planning and control are two classic fields in robotics. Motion Planning can refer to calculating a trajectory or path that allows the robot to move between an initial and a final state, with or without time information respectively. Classically, the complete problem is solved by decomposing it in two subproblems: path planning, where a static-obstacle collision-free geometric path with no time information is calculated, and velocity planning, where the speed with which the path is traveled is adjusted so that force, torque and moving-obstacle collision restrictions are met [40]. In this project, we will understand motion planning in a complete sense, referred to a time including trajectory. In order to do so, we will approach it from an Optimal Control or Trajectory Optimization perspective. Control refers to how we can actuate the system, a robot in our case, in order to achieve and maintain the desired trajectory even when perturbations and unexpected complications appear. Many techniques can be used to this end, ranging from simple PID controllers [7] applied to each robot joint independently, to more complex controllers using the full dynamical model of the robot, like those based on linear quadratic regulators [6], computed-torque methods [49], or model-predictive schemes [28]. When we say kinodynamic planning, we mean that we will not only consider a kinematic approach (geometry and collisions), but that we will also take into consideration the dynamics of the system (effects of forces, actuator torque limits, inertia, and so on.) [22]

With the earlier context in mind, the aim of this thesis is to improve current motion planning and control methods in order to synthetize optimally-agile and graceful robot motions. In order to do so, we first need to convert the abstract notions of agility and gracefulness into concrete metrics and expressions that we can use in an algorithm. Then, we will optimize trajectories, taking into account comprehensive kinematic and dynamic constraints, with high-level optimization goals in which we use all these variables and developments. Lastly, we will measure and benchmark our results in order to demonstrate that trajectories generated using these techniques achieve better performances in the criteria we wanted to accomplish, such as efficiency, smoothness, agility, and so on. In the next section, we will discuss how we are planning to achieve these objectives.

As a particular problem to be solved, we have chosen the Waiter Motion Problem with Mobile Base as our main test case. The classical Waiter Motion Problem involves moving a surface with objects on top, from one position to another, as fast as possible, without the objects sliding or spilling their contents [27]. In order to achieve so, the normal to the surface can be rotated, like a human waiter would do, so that accelerations parallel to its surface are minimized. This problem is an evolution of previous acceleration compensation problems, in which a manipulator tries to compensate accelerations on its base in order to protect the payload using washout algorithms similar to those used in dynamic simulators [29]. Very elegant solutions have been proposed to the Waiter Motion Problem, such as the damped virtual pendulum proposed by Dang and Ebert [20]. However, most of these solutions consider either a fixed base, or a given robot movement to be compensated. Our proposal with the Waiter Motion Problem with Mobile Base is to optimize simultaneously the movement of the robot base and the end mechanism that controls the tray. We will also consider more complex optimization objectives apart from minimal time, such as energy consumption and actuator loads.

We have chosen this problem for two reasons: On one hand, it is clear that the movement of the tray has to align with our objectives of graceful and agile motion, as we want fast yet natural and smooth movements like those needed in this problem. On the other hand, as Dang and Ebert explain, there are multiple real world applications of this problem, such as transport of fragile equipment, liquids, and many others. However, before trying to solve this problem, more accessible, simpler ones will be tried. Simpler problems, such as the simple or double pendulum swing up, can still provide very valuable insights on how to operate with these complex optimization objectives. Interesting studies, such as the one by Zamzami and Ben Amar, has been done on an acrobot modeled as an under-actuated multiple pendulum. [58] Moreover, once we have reached a certain level of expertise in these problems, we will be able to apply the learned lessons to a wider array of problems where these optimization criteria naturally apply.

We have selected an omnidirectional robot with mecanum wheels and a Stewart platform

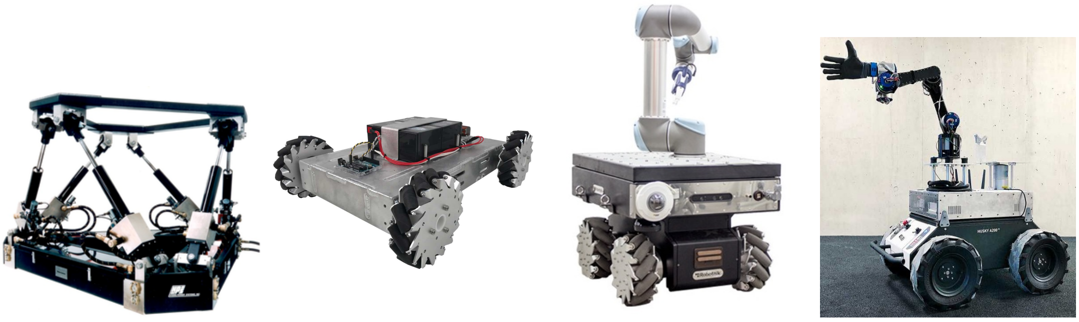

Figure 1: From left to right: a Stewart Platform, an omnidirectional robot with Mecanum wheels, an omnidirectional robot equipped with an articulated arm, a robot with conventional wheels and an articulated arm.

on top as the main physical support for this problem. Since their development in the 70's, the mecanum or directional sliding wheels [1] have captured the attention of robotics researchers and enthusiast due their omnidirectional movement capabilities and pleasing aesthetics. Using a robot with four of such wheels allows us to control movement in both axes of the ground plane and heading angle independently. This flexibility frees us of the non-holonomic constraints of the unicycle model of the wheelchair in the works of Gulati, Kuipers and Park [31] [32] [52]. The Stewart Platform is a parallel manipulator widely studied with many applications [26]. Its robustness and 6-DoF capabilities makes it a perfect candidate for the control of the tray. In fact, it has been used for active acceleration compensation since the beginnings [29]. Both systems, omnibot and Stewart Platform, have the additional advantage of being already available for use in IRI's lab, where we are to conduct our experiments.

However, the tests to be performed will not be restricted to this robotic hardware. If available, different combinations of mobile base and upper platform will be used. Among them, we expect to be able to use non holonomic platforms with articulated arms. In any case, simulations will be conducted to test cases of robots we cannot physically access or exercises that could be dangerous, like heavy machinery surrounded by people.

2 Objectives

With this thesis we first want to develop objective metrics and expressions in order to transform the abstract concepts of agility and gracefulness into concrete measurable values that we can use in our optimizations in a meaningful sense. we then wish to apply and improve algorithms and software tools for trajectory optimization and optimal control in order to optimize robot trajectories under the found metrics and expressions. Finally, we aim to to measure and demonstrate that trajectories generated through these methods achieve our objectives of better high-level performance in terms of as efficiency, smoothness and agility of the robot motions.

It is important to keep in mind that the whole time-dependent trajectory will be optimized at once (in contrast with a decoupled approach that optimizes a path and then calculates speeds along such a path) and that the result of the algorithm should be a control function, that is, the actuator forces or torques, as functions of time, that are required to produce the optimized system trajectory.

In each aspect of this work, the following aims and assumptions will be considered:

- **Constraints:** All kinematic and dynamic constraints of the robot will be taken into consideration. That means that the generated trajectories have to be not only geometrically feasible, but also respect the actuator limits, take into account internal forces and inertia, and respect the physical limits of the modeled machines.
- **Optimization goals definition:** In our optimizations, we will use multi-objective functions to achieve agile and graceful robot motions. End users should be able to balance the weight of each term in such functions so as to have solutions fit to their specific needs. These terms are abstract and vague, so the first critical step to achieve so is to translate them into concrete variables that can be measured and controlled through trajectory planning. Among the concrete parameters that we will measure, we will have:
  - **Jerk ($\dddot{q}$):** As we saw in the introduction, the time derivative of the accelerations are widely used as a measure of how natural and graceful movements are.
  - **Acceleration and speed margins:** In the context of agile movement, considering the available distance to the speed and acceleration envelopes will provide us margin to react to unexpected obstacles or changes in the environment.
  - **Motor torques:** We want our trajectories to produce low loads in our actuators, in order to protect them from wearing and overheating.
  - **Energy efficiency:** We want our movements to be natural and take advantage of the inherent dynamics of the robot, not artificially and blindly smoothing the trajectory at the cost of energy consumption. Strategies to achieve both this objective as well as the previous one could include the use of gravity, friction, or natural frequencies of the system to our advantage.
  - **Time:** Under certain circumstances, the trivial way to minimize loads and accelerations is to approach the destination arbitrarily slow, but we want our trajectories to span reasonable short time periods in order to be actually useful. Therefore, time requirements will be taken into account in our optimization.
- **Application:** We want our work to be able to produce results connected to real world applications. As we discussed in the previous section, we will use the Waiter Motion Problem with Mobile Base as a main test case, supported in the physical world with an omnidirectional robot with a Stewart platform. This will allow us to close the development loop and test our ideas against the real world.
- **Planning and control:** Once our software is mature enough, the project will fork into two components: A high-level optimization that considers the whole movement and a quick real-time model predictive control with receding horizon that allows the robot to react to unexpected changes in the environment.
- **Quality code paradigm:** The software we produce must comply with current standards in software development in order to have readable, maintainable, reliable and expandable code.
- **Open Science and dissemination:** Whenever possible, without compromising the quality of the process, open-source software and tools are to be preferred over closed-source ones in order to maximize the repeatability and reachability of the results. In compliance with the official UPC position with respect to Open Access [21], results and code will be published as open source whenever possible.

3 Expected Contributions

This PhD work seeks to contribute to the general field of robot motion planning and control by applying and expanding strategies, algorithms and software tools for trajectory optimization and optimal control, in order to achieve graceful and agile trajectories subject to kinodynamic constraints.

The results of our work should be both applicable to real world problems and be stable and robust enough to serve as a basis for further research and developments in the field. The main contributions we expect to achieve are:

- We will achieve graceful trajectories that feel natural and predictable to humans and can improve performance, safety and comfort in situations where humans and robots share the same space. This is even more important when the objective of the robot is to transport humans.
- We will translate the abstract concepts we seek into concrete expressions of physical variables that we can measure and control, which will allow us to perform advanced optimizations. This is a meaningful task that must be conducted carefully and thoroughly in order to produce valid and significant model constructs that can be used to produce optimizations whose results are qualitatively superior to those made with simple usual objectives.
- With our optimized trajectories, we will be able to reduce the wearing of components and reduce maintenance costs. They can also improve energy consumption, which is an important issue in the present and expected to become even more important in the future.
- The general cost function to be developed may include several factors expressed in different units and metrics. Combining them in one objective function and exploring their interrelationships is a very non-trivial task. Achieving expressions that can account for them in a balanced, significant and customizable way will allow our results to be useful for a wide array of users and applications.
- We will test and benchmark our developments in order to measure the difference between them and existing approaches. We will have to demonstrate that the results achievable with our methods are significantly improving the performance for a given robot.
- The Waiter Motion Problem with Mobile Base, one of our tests cases, has direct application in industry and can be used to improve safety and speed in automated transport of fragile, delicate or spillable products.
- Our results will be applicable to different problems and robots. This will be demonstrated in physical machines when possible, and in simulations otherwise.

4 State of the Art

We will now present a review of the state of the art in the relevant fields of this PhD work.

4.1 Agile and Graceful Robot Movement

Usually, agility in robotics is studied as a metric to compare different robots in motion [15] [24] [33] [25], or as a design objective in all phases of the robot design [33]. We will however approach it differently. We will go back to the concept itself and translate it into metrics that can be used as objective in a trajectory optimization problem for an existing robot, as discussed in sec. 2.

Gracefulness, or naturalness, on the other hand, has been in fact already used as a criterion in trajectory optimization. Nagarajan, Kantor, and Hollis [51] [50] use it in their *Graceful prepares graph* method, where they precompute some stereotypical graceful movements that they concatenate to generate the trajectory. The concept also plays a central role in the work with robot wheelchairs of Gulati, Kuipers and Park [31] [32] [52]. In their 2008 paper, Gulati and Kuipers [31] decouple the problem into path finding using splines, an then optimize the speed with which it is traveled. Later, in their 2011 dissertation, Gulati [32] optimizes the trajectory by defining it in terms of speed and orientation and applying a Finite Element discretization. In their 2016 dissertation, Park [52] uses a very interesting approach, filling the whole space with a field that contains information about the distance-to-go to the goal that takes into account the non-holonomic constraints of the wheelchair. Over this field, in a short term updating loop, predicts the probable movement of obstacles and tries to find a way to get nearer to the goal following the field while minimizing the chances of obstacle collision. While these approaches are all powerful and useful, and provide us with valuable insight about gracefulness in motions, they follow strategies different to the optimal control one that we want to explore, as we will explain in section 5.

The work done by Zhao, Lin and Tomizuka in their 2018 paper [61], is also very relevant, because their approach is very similar to ours. They use a trajectory optimization paradigm and the tools that they use are the same as the ones that we will base our solution upon. The main differences are our emphasis in the design of a suitable cost function, that our test case is more challenging and that we will use a mobile robot instead of a fixed arm. Their results prove that this line of investigations can be fruitful.

4.2 Trajectory Optimization and Optimal Control

We will approach the objective modeled as a trajectory optimization or optimal control problem. This is a classic formulation that has been used for decades and has a mature and extensive literature corpus. We will present our formulation of the problem in this context in section 5. In our proposal we will follow structures stated in classic textbooks such as Planning Algorithms by Lavalle [47] or the Survey of Numerical Methods for Trajectory Optimization by Betts [8].

For our implementation, we will explore and utilize recent developments in the field such as CasADI (2019) [3], GEKKO (2018)[5], FORCES NLP (2017) [59] or Pyomo [35][36]. Most of these tools interact with independent solvers, among which the most widely used is IPOPT[56].

CasADI[3] is an open-source framework for numerical optimization, developed with optimal control problems in mind. Its core is written in C++ but is more easily used through its interfaces in Python, Matlab or Octave. It features a symbolic framework that allows algorithmic differentiation, translating the numerical problem into a Non Linear Programming formulation that is then solved with an independent solver. CasADI is distributed with some open-source solvers, like IPOPT.

GEKKO[5] is an open source optimization suite for python. It consist in an algebraic modeling language that uses automatic differentiation to translate the high-level numerical problem written in Python into a low level structure that can be passed to specific solvers like IPOPT. Its back-end is written in Fortran to speed up calculations.

FORCES NLP[59] is a software package that generates a tailored C code solver constructed for the specific problem modeled. In its paper, they report achieving speedups of up to an order of magnitude compared to IPOPT. It is a commercial closed-source software, though one of its licensing options can be obtained free of charge for academic purposes.

Pyomo[35][36] is a Python-based open-source software package that supports a diverse set of optimization capabilities for formulating, solving, and analyzing optimization models. A core capability of Pyomo is modeling structured optimization applications. Pyomo can be used to define general symbolic problems, create specific problem instances, and solve these instances using commercial and open-source solvers.

4.3 Kinodynamic Motion Planning

Kinodynamic motion planning is the sub-field of motion planning that takes into consideration kinematic (geometry) and dynamic (forces) constraints [22]. There are several strategies and approaches to this problem. We will use an optimal control/trajectory optimization one, as we will explain in section 5. However, this approach greatly benefits from initial values close to the optimal, so a good knowledge of alternative approaches will come handy in later stages of development. We can divide these methods in two categories: decoupled methods, that first obtain a geometric path with no time information, over which they later plan a trajectory, and coupled methods, that search for time-dependent trajectories directly.

Among the decoupled approaches we can find graph based techniques like the A* [23] or Dijkstra algorithms [38], iterative cell decomposition [16], Potential Fields [4] and sampling methods like the Probabilistic Roadmap (PRM) [41] or the Rapidly-expanding Random Tree (RRT) [46]. While these algorithms can be very useful to find geometrically viable paths, we can't use them as our main tool, because our objective is to optimize complex kinodynamic objectives that these algorithms can't take into account.

Apart from the Trajectory optimization/optimal control approach discussed in the previous point, among the coupled methods we can find Dynamic Programming and Kinodynamic RRTs. Dynamic Programming is based on subdivision of problems into smaller, easier ones. Algorithms that divide the space in a grid, find a path as a series of grid points and then optimize the point to point problem or chose precomputed trajectories to connect them are examples of this approach. This can be an interesting method to have in mind. However, when we consider problems with high numbers of degrees of freedom, these methods can become too expensive, as the grid scales exponentially with the dimension of the state space. Kinodynamic sampling based methods, like Kinodynamic Rapidly-expanding Random Trees, can also be of interest, and are currently under study by other members of our team [11][13][14][12].

4.4 Waiter Motion Problem

Earliest approaches to this problem were made from an Active Acceleration Compensation perspective [29], which filtered high frequencies in order to protect the payload. The active use of inclination in order to counteract long accelerations was also explored in the sibling problem of sloshing suppression [57]. The virtual dampened pendulum proposed by Dang and Ebert in 2004 [20] was a very elegant solution, but it only corresponds to a fraction of the complete problem we want to solve, as it assumes a mobile base over which there is no control and the platform can only react. More recent works have been made, such as Geu Flores and Kecskeméthy [27] or Nagy, Csorvási and Vajk [63]. However, most of these works rest on a physical model of a static-based arm, which again then represents a different problem than ours. That's the reason why we have decided to call our problem "Waiter Motion Problem with Mobile Base", in order to separate it from these similar problems.

4.5 Omnibot and Stewart Platform

Omnidirectional robots with mecanum wheels and Stewart platforms are classic mechanisms that have been widely used and studied in the last decades [1] [26]. The capabilities of the omnibot to travel in a plane without non-holonomic restrictions have made it a classical physical support of all kinds of pathfinding and trajectory planning algorithms, such as potential fields [48], A* cell based search connected by kinodynamic-optimized splines [44] and even machine learning based biomimetic controllers [60]. Our modeling is built over the work done in the department by Iñigo Moreno [19], a Lagrangian dynamics approach based on the work of Agulló et al [2].

5 Methodology and Formulation

We will construct this section using the nomenclature and definitions used in previous works by the Kinematics and Robot Design Team [43].

5.1 Trajectory Optimization

As we discussed before, we will approach the problem from a Trajectory Optimization / Optimal Control perspective. This means we aim to find state and actions trajectories $\boldsymbol{x}(t), \boldsymbol{u}(t)$ that minimizes a cost functional subject to a set of constraints including the system dynamics and all the bounds to which $\boldsymbol{x}(t)$ and $\boldsymbol{u}(t)$ are subject to. Given initial and goal states $\boldsymbol{x_s}$ and $\boldsymbol{x_g}$, respectively, and a running cost function $c(\boldsymbol{x}(t), \boldsymbol{u}(t))$ defined over a finite interval, $t \in [t_0, t_f]$, the basic problem can be formulated as follows:

$$
\begin{aligned}
\underset{\boldsymbol{u}(t)}{\text{minimize}} \quad & \int_{t_0}^{t_f} c(\boldsymbol{x}(t), \boldsymbol{u}(t)) dt \\
\text{subject to} \quad & \forall t, \dot{\boldsymbol{x}} = \boldsymbol{g}(\boldsymbol{x}(t), \boldsymbol{u}(t)), \\
& \forall t, \boldsymbol{h}(\boldsymbol{x}(t), \boldsymbol{u}(t)) \leq 0, \\
& \boldsymbol{x}(t_0) = \boldsymbol{x}_s, \\
& \boldsymbol{x}(t_f) = \boldsymbol{x}_g.
\end{aligned}
$$

Where $\boldsymbol{g}(\boldsymbol{x}(t), \boldsymbol{u}(t))$ codifies the dynamics of the system, and $\boldsymbol{h}(\boldsymbol{x}(t), \boldsymbol{u}(t))$ represent the path constraints, such as torque, speed and geometric limits. In order to express our problem in these terms, we need to transform the equations that control our models until they fit this formulation, and define a meaningful cost function to optimize.

5.2 Problem Formulation

The kinodynamic planning problem typically takes place in the state space of the robot, i.e., the set $\mathcal{X}$ of kinematically-valid states $\boldsymbol{x} = (\boldsymbol{q}, \dot{\boldsymbol{q}})$, where $\boldsymbol{q}$ is a vector of n_q generalized coordinates describing the configuration of the robot, and $\dot{\boldsymbol{q}}$ is the time derivative of $\boldsymbol{q}$, which describes its velocity. The coordinates in $\boldsymbol{q}$ may be independent or not. In the former case, any pair $\boldsymbol{x} = (\boldsymbol{q}, \dot{\boldsymbol{q}})$ where $\boldsymbol{q}$ and $\dot{\boldsymbol{q}}$ are inside their domains is kinematically valid, and $\mathcal{X}$ becomes parametrically defined. The latter case is more complex. The configuration space (C-space) of the robot is the set $\mathcal{C}$ of points $\boldsymbol{q}$ that satisfy a system of n_e equations

$$\boldsymbol{\Phi}(\boldsymbol{q}) = \boldsymbol{0} \tag{1}$$

encoding, e.g., joint assembly, geometric, or contact constraints, either inherent to the robot design or necessary for task execution. The constraints in Eq. (1) are said to be holonomic constraints and only depend on $\boldsymbol{q}$, not on $\dot{\boldsymbol{q}}$. By differentiating Eq. (1), the valid values of $\dot{\boldsymbol{q}}$ are those that fulfill

$$\boldsymbol{\Phi}_{\boldsymbol{q}}(\boldsymbol{q})\, \dot{\boldsymbol{q}} = \boldsymbol{0}, \tag{2}$$

where $\mathbf{\Phi}_{\boldsymbol{q}} = \partial \mathbf{\Phi} / \partial \boldsymbol{q}$. Likewise, the robot may also be subject to a system of n_h nonholonomic constraints

$$\boldsymbol{A}(\boldsymbol{q})\dot{\boldsymbol{q}} = \mathbf{0}, \tag{3}$$

which are velocity constraints that cannot be integrated, i.e., they cannot be expressed as a position constraint like Eq. (1). The consequence of this property is that a nonholonomic robot has more degrees of freedom in position than in velocity. Eqs. (2) and (3) can be combined to form the velocity constraint

$$\begin{bmatrix} \mathbf{\Phi}_{\boldsymbol{q}}(\boldsymbol{q}) \\ \boldsymbol{A}(\boldsymbol{q}) \end{bmatrix} \dot{\boldsymbol{q}} = \boldsymbol{B}(\boldsymbol{q})\dot{\boldsymbol{q}} = \mathbf{0}, \tag{4}$$

where $\boldsymbol{B}(\boldsymbol{q})$ is an $(n_e + n_h) \times n_q$ matrix, which, under mild conditions, can be assumed to be full rank.

Let $\boldsymbol{F}(\boldsymbol{x}) = \mathbf{0}$ denote the system formed by Eqs. (1) and (4). Then, the state space $\mathcal{X}$ of the constrained system becomes a nonlinear manifold of dimension $d_{\mathcal{X}} = 2(n_q - n_e) - n_h$ defined implicitly as

$$\mathcal{X} = \{\boldsymbol{x} : \boldsymbol{F}(\boldsymbol{x}) = \mathbf{0}\}. \tag{5}$$

Any motion planned in $\mathcal{X}$ must also obey the dynamic equations of the robot, which arise from considering the forces and physical laws that determine the system movement. These equations can be written in the form

$$\dot{\boldsymbol{x}} = \boldsymbol{g}(\boldsymbol{x}, \boldsymbol{u}), \tag{6}$$

where $\boldsymbol{g}(\boldsymbol{x}, \boldsymbol{u})$ is an appropriate differentiable function, and $\boldsymbol{u}$ is a n_u-vector of actuator forces subject to lie in a bounded subset $\mathcal{U} \subset \mathbb{R}^{n_u}$. For each value of $\boldsymbol{u}$, Eq. (6) defines a vector field over $\mathcal{X}$, which can be used to integrate the robot motion forward in time, using proper numerical methods.

In order to obtain Eq. (6), constrained systems are usually modeled with the multiplier form of the Euler-Lagrange equations [30]. First the systems are treated as unconstrained systems by ignoring the constraints, and then they are enforced by using Lagrange multipliers. The dynamic equations then take the form

$$\frac{d}{dt}\frac{\partial K}{\partial \dot{\boldsymbol{q}}} - \frac{\partial K}{\partial \boldsymbol{q}} + \frac{\partial U}{\partial \boldsymbol{q}} + \boldsymbol{B}^{\mathsf{T}}\boldsymbol{\lambda} = \boldsymbol{\tau}, \tag{7}$$

where $\boldsymbol{\lambda}$ is a vector of $n_e + n_h$ Lagrange multipliers, $\boldsymbol{\tau}$ is the generalized force corresponding to the non-conservative forces applied on the system, and K and U are the expressions of the kinetic and potential energies of the robot.

The kinetic energy of the robot can always be defined compactly as a quadratic function of $\dot{\boldsymbol{q}}$, that is

$$K = \frac{1}{2}\dot{\boldsymbol{q}}^{\mathsf{T}}\boldsymbol{M}(\boldsymbol{q})\dot{\boldsymbol{q}}, \tag{8}$$

where $\boldsymbol{M}(\boldsymbol{q})$ is the so-called mass matrix, which is always symmetric and positive definite. The potential energy $U = U(\boldsymbol{q})$ is independent of $\dot{\boldsymbol{q}}$. These properties allow Eq. (7) to be written in the form

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \boldsymbol{G}(\boldsymbol{q}) + \boldsymbol{B}^{\mathsf{T}}\boldsymbol{\lambda} = \boldsymbol{\tau}, \tag{9}$$

where $\boldsymbol{G}(\boldsymbol{q})$ is a vector of conservative forces (e.g. gravity or spring forces) given by

$$\boldsymbol{G}(\boldsymbol{q}) = \frac{\partial U}{\partial \boldsymbol{q}}, \tag{10}$$

and $\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})$ is a vector corresponding to Coriolis and centrifugal forces, which is given by

$$\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}}) = (\boldsymbol{M}_{\boldsymbol{q}}\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} - \frac{1}{2}\dot{\boldsymbol{q}}^T\boldsymbol{M}_{\boldsymbol{q}}\dot{\boldsymbol{q}}. \tag{11}$$

Since Eq. (9) is a system of n_q equations in $n_q + (n_e + n_h)$ unknowns (the values of $\ddot{\boldsymbol{q}}$ and $\boldsymbol{\lambda}$), we need additional equations to be able to solve for $\ddot{\boldsymbol{q}}$. These can be obtained by differentiating Eq. (4), which yields

$$\boldsymbol{B}\ddot{\boldsymbol{q}} - \boldsymbol{\xi} = \boldsymbol{0}, \tag{12}$$

where $\boldsymbol{\xi} = -(\boldsymbol{B}_{\boldsymbol{q}}\dot{\boldsymbol{q}})\dot{\boldsymbol{q}}$. Eqs. (9) and (12) can then be written as

$$\begin{bmatrix} \boldsymbol{M}(\boldsymbol{q}) & \boldsymbol{B}^\top \\ \boldsymbol{B} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \ddot{\boldsymbol{q}} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\tau} - \boldsymbol{G}(\boldsymbol{q}) - \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}}) \\ \boldsymbol{\xi} \end{bmatrix}. \tag{13}$$

Clearly, if $\boldsymbol{B}$ is full rank, the matrix on the left-hand side of Eq. (13) is invertible, and thus we can write

$$\ddot{\boldsymbol{q}} = \boldsymbol{f}(\boldsymbol{q},\dot{\boldsymbol{q}},\boldsymbol{u}) = \begin{bmatrix} \boldsymbol{I}_{n_q} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{M}(\boldsymbol{q}) & \boldsymbol{B}^\top \\ \boldsymbol{B} & \boldsymbol{0} \end{bmatrix}^{-1} \begin{bmatrix} \boldsymbol{\tau} - \boldsymbol{G}(\boldsymbol{q}) - \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}}) \\ \boldsymbol{\xi} \end{bmatrix}. \tag{14}$$

If $\boldsymbol{B}$ is not full rank at the given $\boldsymbol{x}$, we say that $\boldsymbol{x}$ is a state-space singularity. At such a point, it is impossible to write Eq. (13) into the form of Eq. (14). It is clear, then, that state-space singularities should be avoided if Eq. (13) is to be solved during the planning process. Fortunately, state-space singularities are nongeneric phenomena that can be avoided by judicious mechanical design [10], or through the addition of singularity-avoidance constraints in Eq. (1) [9].

To obtain Eq. (6), we finally transform Eq. (14) into a first-order ordinary differential equation using the change of variables $\dot{\boldsymbol{q}} = \boldsymbol{v}$, which yields

$$\dot{\boldsymbol{x}} = \begin{bmatrix} \dot{\boldsymbol{q}} \\ \dot{\boldsymbol{v}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{v} \\ \boldsymbol{f}(\boldsymbol{q},\boldsymbol{v},\boldsymbol{u}) \end{bmatrix} = \boldsymbol{g}(\boldsymbol{x},\boldsymbol{u}). \tag{15}$$

Since in practice the actuator forces are limited, $\boldsymbol{u}$ is always constrained to take values in some bounded subset $\mathcal{U}$ of $\mathbb{R}^{n_u}$, which restricts the range of possible state velocities $\dot{\boldsymbol{x}} = \boldsymbol{g}(\boldsymbol{x},\boldsymbol{u})$ at each $\boldsymbol{x} \in \mathcal{X}$. During its motion, moreover, the robot cannot incur in collisions with itself or with the environment, so that the feasible states $\boldsymbol{x}$ are those lying in a subset $\mathcal{X}_{\text{feas}} \subseteq \mathcal{X}$ of non-collision states, where position, velocities, and constraint forces are within given bounds.

With the previous definitions, the planning problem we confront can be phrased as follows. Given two states of $\mathcal{X}_{\text{feas}}$, $\boldsymbol{x}_s$ and $\boldsymbol{x}_g$, find an action trajectory $\boldsymbol{u} = \boldsymbol{u}(t) \in \mathcal{U}$ such that:

- The system trajectory $\boldsymbol{x} = \boldsymbol{x}(t)$ determined by Eqs. (1), (4) and (6) for $\boldsymbol{x}(0) = \boldsymbol{x}_s$, fulfills $\boldsymbol{x}(t_f) = \boldsymbol{x}_g$ for some time $t_f > 0$, and $\boldsymbol{x}(t) \in \mathcal{X}_{\text{feas}}$ for all $t \in [0, t_f]$.
- $\boldsymbol{x}(t)$ is once-differentiable at least, which implies that the computed trajectory will be smooth in position and velocity, and continuous in acceleration.
- The additive cost of executing the trajectory

$$C(\boldsymbol{x}(0),\boldsymbol{u}(t)) = \int_0^{t_f} c(\boldsymbol{x}(t),\boldsymbol{u}(t))\,dt \tag{16}$$

  is, at least, locally optimal for some given instantaneous cost function $c(\boldsymbol{x}(t),\boldsymbol{u}(t))$.
- Apart from geometric holonomic and non-holonomic constraints described in equations 1 and 3, additional path constraints, such as torque limits, have to be taken into account:

$$\forall t, \boldsymbol{h}(\boldsymbol{x}(t),\boldsymbol{u}(t)) \leq 0$$

We can observe that this formulation is equivalent to the expressions shown in section 5.1. Note that the previous problem can be considered as a full motion planning problem, as opposed to a *path planning* problem that only asks for a connecting curve in the C-space, without reference to the dynamics of the robot. Following [22, 45] we shall use the term kinodynamic planning to refer to such a planning problem. Note however that, contrary to [22, 45] we allow the presence of Eqs. (1) and (4) in the problem, which makes it more general and challenging at the same time.

5.3 Objective Function Calculation

As we saw in section 2, one of the central objectives of this work will be to find suitable cost functions that can balance different properties in order to achieve our goal of Agile and Graceful motions. Extensive and systematic testing and experimentation will be conducted in order to study how each of the measurable physical variables affects the optimization, both in terms of quality and adequacy of the obtained trajectory and the computational cost it requires. We will explore how each abstract adjectives could be translated into concrete mathematical functions of the real variables. We will study how these variables correlate and interact with each other in order to find efficient algorithms that can reach the desired results. We will develop parameter-based expressions that give the user power to balance the terms in order to fine-tune the results to their particular objective.

Among the parameters that we will study, there will be:

- **Jerk ($\dddot{q}$),** the time derivative of acceleration, plays a vital role in achieving graceful and natural movements.
- **Acceleration and speed margins:** At each $\boldsymbol{x}(t)$, the distance to the envelopes has to be taken into account. This will translate into trajectories with high implicit agility, that allow the robot to respond quickly to unexpected environmental changes.
- **Energy efficiency:** We have to consider instantaneous power spikes as well as the total energy spent. In order to do so, valid electric motor models have to be developed and included in the model. In itself, energy efficiency is a desirable objective, but it will contribute to natural movements as well, because in nature, animal movements tend to use their intrinsic dynamics in order to minimize energy loss.
- **Motor Torques:** We want to avoid torque spikes, but also try to minimize the mean torque load. Controlling the actuator loads will lead to graceful and natural movements, minimize energy consumption and reduce wearing damage to the system.
- **Time:** is an important factor in achieving useful trajectories. It will be important that the end user can balance the algorithm according to their own needs. In some cases, achieving the final configuration in minimum time will be an important optimization objective.

5.4 Discretization

Once we have a formulation for the problem in continuous variables, we have to discretize it in order to solve it numerically. There are two main approaches to the discretization: Direct and Indirect. According to Kelly [42], a common way to distinguish these two methods is that a direct method discretizes and then optimizes, while an indirect method optimizes and then discretizes.In general, it is considered that direct methods have a bigger convergence region, while indirect methods tend to be more accurate and will have a more reliable error estimate.[42] [55]

Another classification of trajectory optimization techniques is the distinction between shooting and collocation methods [8].

- Shooting methods [39, 54, 62] discretize the trajectories into multiple knot points $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$, $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_N$, and then enforce the integral of the dynamics between these points as a constraint:

$$\begin{aligned}
\underset{\boldsymbol{x}_0,\ldots,\boldsymbol{x}_N,\boldsymbol{u}_0,\ldots,\boldsymbol{u}_{N-1}}{\text{minimize}} \quad & T_s \sum_{n=0}^{N-1} c(\boldsymbol{x}_n, \boldsymbol{u}_n) \\
\text{subject to} \quad & \boldsymbol{x}_{n+1} = \boldsymbol{g}_d(\boldsymbol{x}_n, \boldsymbol{u}_n) \quad \forall n \in [0, N-1], \\
& \boldsymbol{x}_0 = \boldsymbol{x}_s, \\
& \boldsymbol{x}_N = \boldsymbol{x}_g.
\end{aligned}$$

  Here, T_s is the time increment employed, and $\boldsymbol{g_d}(\cdot)$ is a discrete approximation of the differential equation, either using an Euler method, or any higher-order method if more accuracy is necessary. Clearly, there is a trade-off between both the number of knot points and the integration method adopted, and the computational cost required to solve the resulting optimization problem.

- Collocation methods [34] alleviate this issue by avoiding numerical integration. Both the input $\boldsymbol{u}(t)$ and state $\boldsymbol{x}(t)$ trajectories are approximated explicitly by means of polynomial functions. Specifically, $\boldsymbol{u}(t)$ is described by a first-order polynomial defined by the $\boldsymbol{u}$ values at the knot points, while $\boldsymbol{x}(t)$ is described by an Hermitian spline defined by the $\boldsymbol{x}$ and $\dot{\boldsymbol{x}}$ values at such points ($\dot{\boldsymbol{x}}$ being computed using the dynamics in Eq. (9)). Finally, a constraint forces the satisfaction of Eq. (9) at the midpoint of the spline, also known as the collocation point. Thus, the optimization problem can be stated as:

$$\begin{aligned}
\underset{\boldsymbol{x}_0,\ldots,\boldsymbol{x}_N,\boldsymbol{u}_0,\ldots,\boldsymbol{u}_{N-1}}{\text{minimize}} \quad & T_s \sum_{n=0}^{N-1} c(\boldsymbol{x}_n, \boldsymbol{u}_n) \\
\text{subject to} \quad & 0 = \boldsymbol{h}(\boldsymbol{x}_n, \boldsymbol{u}_n, \boldsymbol{x}_{n+1}, \boldsymbol{u}_{n+1}) \quad \forall n \in [0, N-1], \\
& \boldsymbol{x}_0 = \boldsymbol{x}_s, \\
& \boldsymbol{x}_N = \boldsymbol{x}_g,
\end{aligned}$$

  where $\boldsymbol{h}$ refers to the collocation constraint. This method is powerful enough to be applied to challenging problems involving humanoids [17, 18, 37] and kinematic constraints [53]. In particular, in [53] constraints in the form of Eqs. (1) and (2) are enforced at the knot points and then added to the optimization problem. However, a large set of points are still needed to accurately approximate the constraint manifold and the dimension of the optimization problem increases considerably.Moreover, the kinematic constraints are fulfilled at these intermediate points but not necessarily along the whole trajectory.

6 Resources and Work Plan

The proposed research is framed within the KINODYN research project, code DPI2017-88282-P and is funded by an FPI grant associated with such project. The project will be developed at the Institut de Robòtica i Informàtica Industrial UPC-CSIC in Barcelona.

The software and algorithms developed are not expected to require special equipment apart from the standard desktop computer available in the office. Robotic components, such as the omnibot, the Stewart platform and additional external sensors will be reused from previous works at the department. A sufficiently large test area will be used to perform physical tests.

Since this PhD work entails developing appropriate motion planning and control algorithms, the typical work cycle consists in the following steps:

- Develop the theoretical foundation of the algorithms.
- Implement the algorithms in some computer language (usually Matlab, C, ROS, or Python)
- Test the algorithms in simulation.
- Test the algorithms in a real robot using the robot's sensors and actuators.

The work plan for the proposed research is divided into tasks, which are subdivided into subtasks, as described below. The schedule of this plan is presented in Fig. 2 as a Gantt chart. In this chart, **Q1**, ..., **Q4** stand for the four quarters of a year, and the work already completed is shown shaded in orange.

Task 1: Review of the state of the art and study of relevant material

Familiarization with the problem and the means to solve it is a fundamental first step. However, reading related material and literature will not stop with the start of the development, but it will be an ongoing process until the later stages of the thesis writing.

Task 2: Trajectory optimization and optimal control exploration

First, in parallel with literature review, small benchmarks and problems such as the pendulum swing-up are to be conducted in order to get familiar with the theoretical concepts and practical workflows. This also serves the purpose of trying and comparing different algorithms and tools before using them in the real problem, and therefore, this task is expected to continue intermittently as we test newly found ones before committing to them.

Task 3: Simple optimization problems

We will solve simple optimization problems, such as simple or under-actuated pendulums, in order to experiment and test with different tools and objective functions.

Task 4: Advanced objective functions

We will systematically study the factors explained in section 5 in order to achieve graceful, agile and efficient motions, and how to use them for our objective functions in meaningful ways.

Task 5: Problem-specific optimization

We will develop optimization algorithms specific to the Waiter Motion Problem with Mobile Base, modeled for the Omnibot with Stewart platform.

Task 5.1: Omnibot model characterization

A detailed study of the omnibot model is conducted in order to gain insight on its nature and behavior.

Task 5.2: First simplified kinodynamic planner

A first version of the optimizer, using a simplified problem, will be developed.

Task 5.3: Stewart Platform model characterization

A detailed study of the parallel manipulator model is conducted in order to gain insight on its nature and behavior.

Task 5.4: Complete problem

The complete optimization problem will be addressed and the test case will be completed.

Task 6: Physical implementation and feedback

Our models and results should be implemented in the physical robot and experimental measurements should be done in order to test their validity and find unexpected behaviors. The initial idea was to do this since the beginning, but sanitary circumstances suggest that it may be a good idea to wait until summer 2021.

Task 7: Simulations

Once validity is tested and experiments become too time-consuming, simulations will be conducted and established in the development loop.

Task 8: Application to different cases and problems

The algorithms developed for the test case will be widened and adapted to use for different problems and robots.

Task 9: Dissertation writing

This task entails the redaction of the thesis dissertation and the preparation of its public defense.

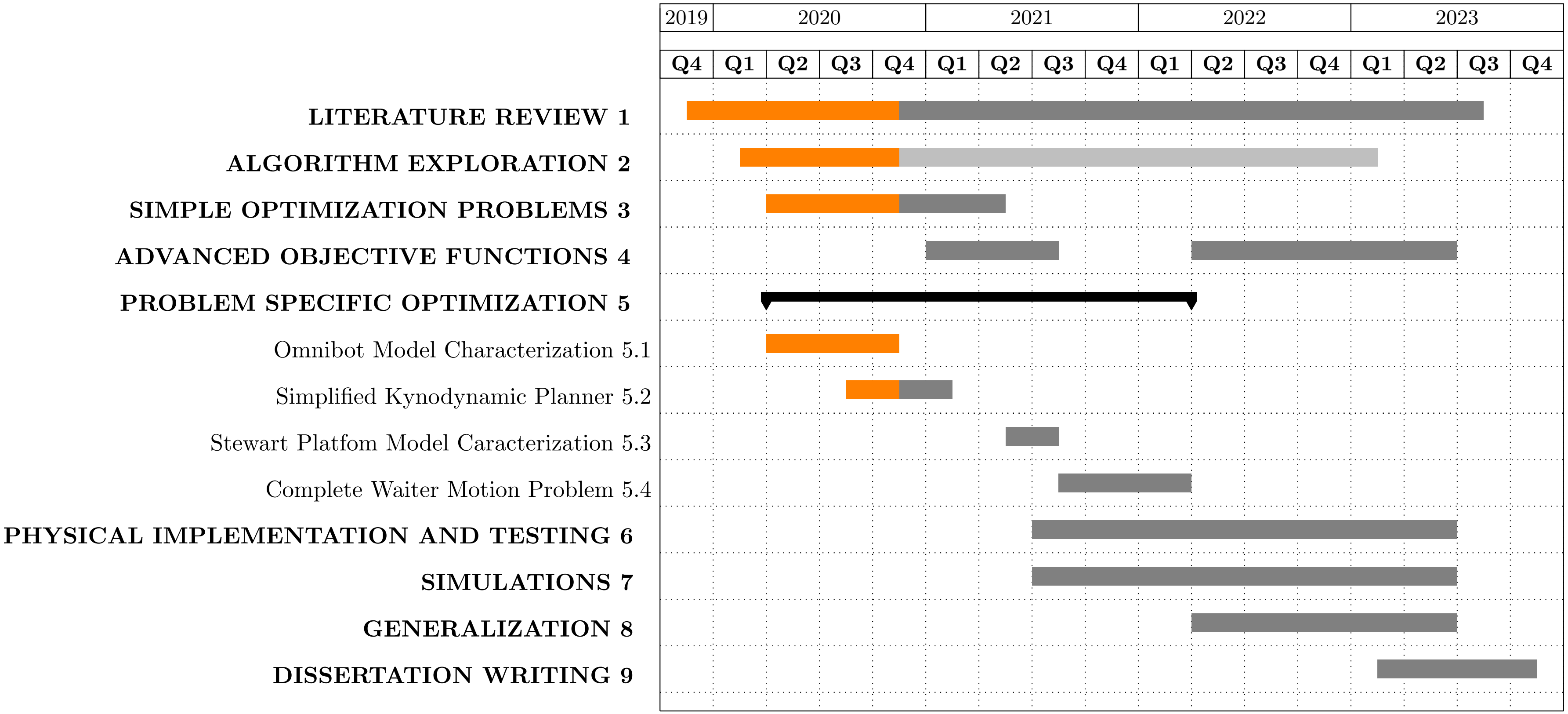

Figure 2: Work plan of the proposed work

References

[1] J Agulló, S Cardona, and J Vivancos. Kinematics of vehicles with directional sliding wheels. *Mechanism and Machine Theory*, 22(4):295 – 301, 1987. ISSN 0094-114X. doi: https://doi.org/10.1016/0094-114X(87)90018-8. URL `http://www.sciencedirect.com/science/article/pii/0094114X87900188`.

[2] J Agulló, S Cardona, and J Vivancos. Dynamics of vehicles with directionally sliding wheels. *Mechanism and Machine Theory*, 24(1):53 – 60, 1989. ISSN 0094-114X. doi: https://doi.org/10.1016/0094-114X(89)90083-9. URL `http://www.sciencedirect.com/science/article/pii/0094114X89900839`.

[3] Joel A. E. Andersson, Joris Gillis, Greg Horn, James B. Rawlings, and Moritz Diehl. Casadi: a software framework for nonlinear optimization and optimal control. *Mathematical Programming Computation*, 11(1):1–36, Mar 2019. ISSN 1867-2957. doi: 10.1007/s12532-018-0139-4. URL `https://doi.org/10.1007/s12532-018-0139-4`.

[4] J. Barraquand, B. Langlois, and J. . Latombe. Numerical potential field techniques for robot path planning. *IEEE Transactions on Systems, Man, and Cybernetics*, 22(2):224–241, 1992. doi: 10.1109/21.148426.

[5] Logan Beal, Daniel Hill, R. Martin, and John Hedengren. Gekko optimization suite. *Processes*, 6(8):106, Jul 2018. ISSN 2227-9717. doi: 10.3390/pr6080106. URL `http://dx.doi.org/10.3390/pr6080106`.

[6] Alberto Bemporad, Manfred Morari, Vivek Dua, and Efstratios N. Pistikopoulos. The explicit linear quadratic regulator for constrained systems. *Automatica*, 38(1):3 – 20, 2002. ISSN 0005-1098. doi: https://doi.org/10.1016/S0005-1098(01)00174-1. URL `http://www.sciencedirect.com/science/article/pii/S0005109801001741`.

[7] S. Bennett. Development of the pid controller. *IEEE Control Systems Magazine*, 13(6): 58–62, 1993. doi: 10.1109/37.248006.

[8] John T. Betts. Survey of numerical methods for trajectory optimization. *Journal of Guidance, Control, and Dynamics*, 21(2):193–207, 1998. doi: 10.2514/2.4231. URL `https://doi.org/10.2514/2.4231`.

[9] O. Bohigas, M. E. Henderson, L. Ros, M. Manubens, and J. M. Porta. Planning singularity-free paths on closed-chain manipulators. *IEEE Transactions on Robotics*, 29(4):888–898, 2013.

[10] O. Bohigas, M. Manubens, and L. Ros. *Singularities of robot mechanisms: numerical computation and avoidance path planning*, volume 41 of *Mechanisms and Machine Science*. Springer, 2016.

[11] R. Bordalba, L. Ros, and J. M. Porta. Randomized kinodynamic planning for constrained systems. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7079–7086, 2018. doi: 10.1109/ICRA.2018.8460753.

[12] R. Bordalba, L. Ros, and J. M. Porta. A randomized kinodynamic planner for closed-chain robotic systems. *IEEE Transactions on Robotics*, pages 1–17, 2020. doi: 10.1109/TRO.2020.3010628.

[13] Ricard Bordalba, Josep M. Porta, and Lluís Ros. Randomized kinodynamic planning for cable-suspended parallel robots. In Clément Gosselin, Philippe Cardou, Tobias Bruckmann, and Andreas Pott, editors, *Cable-Driven Parallel Robots*, pages 195–206, Cham, 2018. Springer International Publishing.

[14] Ricard Bordalba, Lluís Ros, and Josep M. Porta. Randomized planning of dynamic motions avoiding forward singularities. In Jadran Lenarcic and Vincenzo Parenti-Castelli, editors, *Advances in Robot Kinematics 2018*, pages 170–178, Cham, 2019. Springer International Publishing.

[15] Alan Bowling and Shih-Chien Teng. Performance measures of agility for mobile robots. 01 2009. doi: 10.1145/1865909.1865949.

[16] R. A. Brooks and T. Lozano-Pérez. A subdivision algorithm in configuration space for findpath with rotation. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-15(2): 224–233, 1985. doi: 10.1109/TSMC.1985.6313352.

[17] T. Buschmann, S. Lohmeier, M. Bachmayer, H. Ulbrich, and F. Pfeiffer. A collocation method for real-time walking pattern generation. In *International Conference on Humanoid Robots*, pages 1–6. IEEE, 2007.

[18] T. Buschmann, S. Lohmeier, and H. Ulbrich. Humanoid robot lola: Design and walking control. *Journal of physiology-Paris*, 103(3):141–148, 2009.

[19] Iñigo Moreno Caireta. Model predictive control for a mecanum-wheeled robot in dynamical environments, 2019.

[20] A. X. H. Dang and I. Ebert-Uphoff. Active acceleration compensation for transport vehicles carrying delicate objects. *IEEE Transactions on Robotics*, 20(5):830–839, 2004. doi: 10.1109/TRO.2004.832791.

[21] Universidad Politècnica de Catalunya. Open acces approach and positions of the upc. Open Acces approach and positions of the UPC, 2009.

[22] Bruce Donald, Patrick Xavier, John Canny, and John Reif. Kinodynamic motion planning. *J. ACM*, 40(5):1048–1066, November 1993. ISSN 0004-5411. doi: 10.1145/174147.174150. URL `https://doi.org/10.1145/174147.174150`.

[23] František Duchoň, Andrej Babinec, Martin Kajan, Peter Beňo, Martin Florek, Tomáš Fico, and Ladislav Jurišica. Path planning with modified a star algorithm for a mobile robot. *Procedia Engineering*, 96:59 – 69, 2014. ISSN 1877-7058. doi: https://doi.org/10.1016/j.proeng.2014.12.098. URL `http://www.sciencedirect.com/science/article/pii/S187770581403149X`. Modelling of Mechanical and Mechatronic Systems.

[24] Pusey J.L. Duperret J.M., Kenneally G.D. and Koditschek D.E. Towards a Comparative Measure of Legged Agility. In *Hsieh M., Khatib O., Kumar V. (eds) Experimental Robotics. Springer Tracts in Advanced Robotics, vol 109.* Springer, 2016.

[25] P. Eckert and A. J. Ijspeert. Benchmarking agility for multilegged terrestrial robots. *IEEE Transactions on Robotics*, 35(2):529–535, 2019. doi: 10.1109/TRO.2018.2888977.

[26] E.F. Fichter. A stewart platform- based manipulator: General theory and practical construction. *The International Journal of Robotics Research*, 5(2):157–182, 1986. doi: 10.1177/027836498600500216. URL `https://doi.org/10.1177/027836498600500216`.

[27] Francisco Geu Flores and Andrés Kecskeméthy. *Time-Optimal Path Planning for the General Waiter Motion Problem*, pages 189–203. Springer International Publishing, Heidelberg, 2013. ISBN 978-3-319-00398-6. doi: 10.1007/978-3-319-00398-6_14. URL https://doi.org/10.1007/978-3-319-00398-6_14.

[28] Carlos E. García, David M. Prett, and Manfred Morari. Model predictive control: Theory and practice—a survey. *Automatica*, 25(3):335 – 348, 1989. ISSN 0005-1098. doi: https://doi.org/10.1016/0005-1098(89)90002-2.

[29] R. Graf and R. Dillmann. Active acceleration compensation using a stewart-platform on a mobile robot. In *Proceedings Second EUROMICRO Workshop on Advanced Mobile Robots*, pages 59–64, 1997. doi: 10.1109/EURBOT.1997.633569.

[30] D. T. Greenwood. *Advanced dynamics*. Cambridge University Press, 2006.

[31] S. Gulati and B. Kuipers. High performance control for graceful motion of an intelligent wheelchair. In *2008 IEEE International Conference on Robotics and Automation*, pages 3932–3938, 2008. doi: 10.1109/ROBOT.2008.4543815.

[32] Shilpa Gulati. *A Framework for Characterization and Planning of Safe, Comfortable, and Customizable Motion of Assistive Mobile Robots*. PhD thesis, University of Texas at Austin, 2011.

[33] Duncan W. Haldane, M. M. Plecnik, J. K. Yim, and R. S. Fearing. Robotic vertical jumping agility via series-elastic power modulation. *Science Robotics*, 1(1), 2016. doi: 10.1126/scirobotics.aag2048. URL https://robotics.sciencemag.org/content/1/1/eaag2048.

[34] C. R. Hargraves and S. W. Paris. Direct trajectory optimization using nonlinear programming and collocation. *Journal of Guidance, Control, and Dynamics*, 10(4):338–342, 1987.

[35] William E Hart, Jean-Paul Watson, and David L Woodruff. Pyomo: modeling and solving mathematical programs in python. *Mathematical Programming Computation*, 3(3):219–260, 2011.

[36] William E. Hart, Carl D. Laird, Jean-Paul Watson, David L. Woodruff, Gabriel A. Hackebeil, Bethany L. Nicholson, and John D. Siirola. *Pyomo–optimization modeling in python*, volume 67. Springer Science & Business Media, second edition, 2017.

[37] A. Hereid, E. A. Cousineau, C. M. Hubicki, and A. D. Ames. 3D dynamic walking with underactuated humanoid robots: A direct collocation framework for optimizing hybrid zero dynamics. In *IEEE International Conference on Robotics and Automation*, pages 1447–1454. IEEE, 2016.

[38] Huijuan Wang, Yuan Yu, and Quanbo Yuan. Application of dijkstra algorithm in robot path-planning. In *2011 Second International Conference on Mechanic Automation and Control Engineering*, pages 1067–1069, 2011. doi: 10.1109/MACE.2011.5987118.

[39] M. Kalakrishnan, S. Chitta, E. Theodorou, P. Pastor, and S. Schaal. STOMP: Stochastic trajectory optimization for motion planning. In *2011 IEEE International Conference on Robotics and Automation*, pages 4569–4574, 2011.

[40] Kamal Kant and Steven W. Zucker. Toward efficient trajectory planning: The path-velocity decomposition. *The International Journal of Robotics Research*, 5(3):72–89, 1986. doi: 10.1177/027836498600500304. URL https://doi.org/10.1177/027836498600500304.

[41] L. E. Kavraki, P. Svestka, J. . Latombe, and M. H. Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Transactions on Robotics and Automation*, 12(4):566–580, 1996. doi: 10.1109/70.508439.

[42] Matthew Kelly. An introduction to trajectory optimization: How to do your own direct collocation. *SIAM Review*, 59(4):849–904, 2017. doi: 10.1137/16M1062569. URL `https://doi.org/10.1137/16M1062569`.

[43] Kinematics and Robot Design Group. Kinodynamic planning ofefficient and agile robot motions. Kinodyn Project web page in IRI's portal, 2017.

[44] F. Kuenemund, D. Hess, and C. Roehrig. Energy efficient kinodynamic motion planning for holonomic agvs in industrial applications using state lattices. In *Proceedings of ISR 2016: 47st International Symposium on Robotics*, pages 1–8, 2016.

[45] S. M. LaValle and J. J. Kuffner. Randomized kinodynamic planning. *International Journal of Robotics Research*, 20(5):378–400, 2001.

[46] Steven M LaValle. Rapidly-exploring random trees: A new tool for path planning. 1998.

[47] Steven M. Lavalle. *Plannin Algorithms*. Cambridge, 2006.

[48] Weihao Li, Chenguang Yang, Yiming Jiang, Xiaofeng Liu, and Chun-Yi Su. Motion planning for omnidirectional wheeled mobile robot by potential field method. *Journal of Advanced Transportation*, 2017:4961383, Mar 2017. ISSN 0197-6729. doi: 10.1155/2017/4961383. URL `https://doi.org/10.1155/2017/4961383`.

[49] R. H. Middletone and G. C. Goodwin. Adaptive computed torque control for rigid link manipulators. In *1986 25th IEEE Conference on Decision and Control*, pages 68–73, 1986. doi: 10.1109/CDC.1986.267156.

[50] Umashankar Nagarajan. *Fast and Graceful Balancing Mobile Robots*. PhD thesis, Carnegie Mellon University, 2012.

[51] Umashankar Nagarajan, George Kantor, and Ralph Hollis. Integrated motion planning and control for graceful balancing mobile robots. *The International Journal of Robotics Research*, 32(9-10):1005–1029, 2013. doi: 10.1177/0278364913488011. URL `https://doi.org/10.1177/0278364913488011`.

[52] Jong Jin Park. *Graceful Navigation for Mobile Robots in Dynamic and Uncertain Environments*. PhD thesis, University of Michigan, 2016.

[53] M. Posa, S. Kuindersma, and R. Tedrake. Optimization and stabilization of trajectories for constrained dynamical systems. In *IEEE International Conference on Robotics and Automation*, pages 1366–1373, 2016.

[54] J. Schulman, Duan Y, J. Ho, A. Lee, I. Awwal, H. Bradlow, J. Pan, S. Patil, K. Goldberg, and P. Abbeel. Motion planning with sequential convex optimization and convex collision checking. *International Journal of Robotics Research*, 33(9):1251–1270, 2014.

[55] O. von Stryk and R. Bulirsch. Direct and indirect methods for trajectory optimization. *Annals of Operations Research*, 37(1):357–373, Dec 1992. ISSN 1572-9338. doi: 10.1007/BF02071065. URL `https://doi.org/10.1007/BF02071065`.

[56] Andreas Wächter and Lorenz T. Biegler. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming*, 106(1):25–57, Mar 2006. ISSN 1436-4646. doi: 10.1007/s10107-004-0559-y. URL `https://doi.org/10.1007/s10107-004-0559-y`.

[57] K. Yano and K. Terashima. Robust liquid container transfer control for complete sloshing suppression. *IEEE Transactions on Control Systems Technology*, 9(3):483–493, 2001. doi: 10.1109/87.918901.

[58] Ziad Zamzami and Faïz Ben Amar. Dynamic coupling map: Acceleration space analysis for underactuated robots. In *Proceedings of the 13th International Conference on Informatics in Control, Automation and Robotics*, ICINCO 2016, page 548–557, Setubal, PRT, 2016. SCITEPRESS - Science and Technology Publications, Lda. ISBN 9789897581984. doi: 10.5220/0006012405480557. URL `https://doi.org/10.5220/0006012405480557`.

[59] A. Zanelli, A. Domahidi, J. Jerez, and M. Morari. Forces nlp: an efficient implementation of interior-point methods for multistage nonlinear nonconvex programs. *International Journal of Control*, 93(1):13–29, 2020. doi: 10.1080/00207179.2017.1316017. URL `https://doi.org/10.1080/00207179.2017.1316017`.

[60] Weibin Zhang. Design and adaptive control of omni-directional mobile platform with four driving wheels. *American Journal of Scientific Research and Essays*, 3(7), 2018. doi: 10.28933/ajsre-2018-06-2801.

[61] Y. Zhao, H. Lin, and M. Tomizuka. Efficient trajectory optimization for robot motion planning. In *2018 15th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, pages 260–265, 2018. doi: 10.1109/ICARCV.2018.8581059.

[62] M. Zucker, N. Ratliff, A. D. Dragan, M. Pivtoraiko, M. Klingensmith, C. M. Dellin, J. A. Bagnell, and S. S. Srinivasa. CHOMP: Covariant Hamiltonian optimization for motion planning. *International Journal of Robotics Research*, 32(9-10):1164–1193, 2013.

[63] Nagy Á., Csorvási G., and Vajk I. Path tracking algorithms for non-convex waiter motion problem. *Periodica Polytechnica Electrical Engineering and Computer Science*, 62(1):16–23, 2018. doi: https://doi.org/10.3311/PPeecs.11606.