I+D Project Proposal:

Kinodynamic Planning of Efficient and Agile Robot Motions

(KINODYN)

Kinematics and Robot Design Group
IRI (CSIC/UPC)

July 2017
# Contents

1 Objective ........................................... 1
2 Specific Objectives .................................. 2
3 Expected Contributions ................................. 3
4 Applications ......................................... 3
5 Background of the Project Team ....................... 4
6 Formal Problem Statement ......................... 4
7 State of the Art .................................... 6
   7.1 Decoupled Planning ............................... 6
   7.2 Direct Planning .................................. 7
8 Methodology ........................................ 9
9 Work Plan .......................................... 10
   9.1 Task Decomposition ............................ 10
   9.2 Benchmarks ...................................... 13
   9.3 Project Milestones and Temporal Chart .......... 14
C.1: PROPUESTA CIENTÍFICA

1 Objective

Robotics is flourishing. Innovative robot mechanisms constantly see the light of day, and their use may increase dramatically in the near future. Whether on Earth or in Space, from research labs, to medicine, or the industry, we see parallel and walking robots, flying manipulators, anthropomorphic hands and arms, humanoids, and other sophisticated machines in action. The capacity to autonomously plan and perform complex motions is key on such devices, and robotics has provided many solutions to this end. Despite the impressive advances, however, roboticists are beginning to recognize that robots still move far too conservatively [97], and accomplish only a fraction of the tasks and achieve only a fraction of the performance that they are mechanically capable of. This must be attributed to the fact that many robots are fundamentally limited by control technology that matured on rigid robotic arms in factory environments. Such robots use high-gain control loops, and therefore considerable joint torque, to cancel out their natural dynamics to strictly follow a desired trajectory. This approach to robot motion makes the problem tractable, but comes at a high price: a robot consumes much more energy than a human does to perform the same task, and it requires an oversized structure to support excessively large motors and resist their reactions. The result is a machine that is much less efficient and agile when compared to what a human, or an animal, would be in accomplishing a similar task (Figs. 1 and 2).

The objective of this project is to investigate how energy-efficient and agile robot motions can be planned and executed in an efficient and reliable way. While robot movements are usually rigid and stereotyped, our aim is to make them more graceful. This does not mean to avoid jagged movements by simply smoothing the trajectory, but to adapt each movement to the natural frequency of the robot parts and manipulated objects, taking advantage of gravity, inertia, and centripetal forces, and thus reducing the internal forces and global effort of the robot.

Figure 1: Top: The Atlas robot from Boston Dynamics loading a 4.5 Kg box onto a shelf (images courtesy of Boston Dynamics). Bottom: A man loading two heavy gas bottles on a truck. The man makes optimal use of its limited force by exploiting dynamics.
A departing hypothesis is the realisation that such motions can only be generated by (1) taking the full robot dynamics into account, and (2) making an optimal use of the limited power, energy, and strength capacities of the robot equipment. To a large extent, this calls for offloading lower-level control loops in their task to achieve feasible, conservative motions, transferring part of their duty to higher-level motion planners that, by considering the full robot dynamics, are able to achieve graceful natural motions compliant with motor torque, energy storage, or material resistance limitations. A second hypothesis is the observation that there are new computational tools from motion planning, numerical continuation, differential geometry, multibody dynamics, and robot singularity theory, that can be employed to devise a high-level motion planner taking all such limitations into account. The team members of this project accumulate substantial expertise in these tools, and thus they feel in a good position to reach the project objective successfully. The current proposal, in fact, is a natural extension of earlier developments by these members, which took place within the projects DPI2004-07358, DPI2007-60858, DPI2010-18449, and DPI2014-57220-C2-2-P of the Spanish National Plan.

2 Specific Objectives

This project pursues three specific objectives:

1. To develop a motion planner that is able to produce energy-efficient and agile motions compatible with all kinematic and dynamic constraints of a robot. Given an initial and a final state for the robot, this planner has to provide a feasible, locally-optimal state-space trajectory that connects the two states. In addition to the usual requirements of collision avoidance, shortest path, and joint limitations, this planner will have to satisfy limitations in the forces that can be applied to the joints and the stresses that can be supported by the links. At the same time, the planner will have to minimize a user-specified cost function involving aspects like energy consumption, speed and acceleration of the different parts, execution times, and the like. In particular, it will be able to address (1) serial or general open-chain robots, (2) parallel robots and closed kinematic chains of any topology, and (3) robots of any kind manipulating a known load, all of them moving with or without gravity.

2. To implement the planner as a self-contained software library. This library will be built upon the CUIK Suite, a software platform for motion analysis and planning of general multibody systems developed by the proposing group [77]. At present, this suite is in quite a mature state, but it can only handle kinematic constraints. Our goal is to extend it to also cope with the aforementioned dynamic constraints and trajectory cost functions.

3. To demonstrate the usefulness and performance of the planner by applying it to synthesize robot motions similar to those shown in Figs. 1 and 2 in illustrative test cases, both in simulation using virtual robots, and on a real state-of-the art robot torso equipped with multifingered hands (see Section 9.2 for details).
3 Expected Contributions

The main contribution of this research will be a kinodynamic motion planner able to synthesize energy-efficient and agile motion trajectories for a robot with arbitrary multibody structure. The term *kinodynamic* is used here in the usual sense in robotics [30, 59, 61, 61], to emphasize the fact that the mentioned trajectories will have to satisfy all kinematic and dynamic constraints of the problem at hand. Namely:

- **Kinematic constraints** are those only involving position and velocity coordinates of the robot. These include assembly constraints, constraints due to sliding and rolling contacts between bodies, closed kinematic loops inherent to the robot structure or to the task to be executed, collision avoidance, and joint or velocity limits.

- **Dynamic constraints** are those involving position, velocity and acceleration coordinates, and the forces acting on the system. These correspond to the equations of motion of the robot, limits on the actuator or constraint forces, or existing acceleration bounds.

The resulting planner will incorporate state-of-the-art techniques for kinodynamic motion planning [59, 61], but also newer techniques under development by the project team of this proposal [9, 11, 13, 34, 65, 76]. These newer techniques will extend state-of-the-art ones to also cope with so-called constrained robotic systems, which are those involving closed kinematic chains in their multibody structure. Such systems cannot be managed by the latest techniques of motion planning (including those surveyed in [59, 91], or implemented in [55]), but arise frequently in many domains, like in parallel manipulators, robots in contact with objects or with the environment, or when virtual geometric constraints are needed to fulfill a given task. Loop closure constraints introduce singularities that complicate the planning, and induce a complex geometric structure on the robot state space. The newer techniques by the project team would cope with such difficulties (see Section 8). Thus, they would represent a significant advance in the field of motion planning.

A second contribution will be an open-source library implementing the planner. This library will be integrated with ROS [80] and it will be freely distributed to the robotics community as part of the CUIK suite, a software platform under continuous development by the research team of this proposal [77]. With the mentioned library, robot engineers will be able to generate energy-efficient and agile robot trajectories that take into account:

- The full dynamic model of the robot, allowing the generation of dynamically stable robot motions.
- The limited power deliverable by the actuators.
- The maximum forces that can be transmitted between the robot and its environment, e.g., between the robot hands, and the grasped or touched objects.
- The maximum constraint, gravity, friction, and inertia forces that the robot skeleton can absorb, without violating the materials resistance of its constituent parts.

4 Applications

The availability of a kinodynamic planner like the one we envisage would allow a robot to automatically convert high-level specifications of a task into low-level descriptions on how to achieve the task. This planner, in particular, would find applications in the following domains:

- In Robotics, the planner could be used to synthesize energy-efficient and agile robot trajectories for systems involving open or closed kinematic chains. This is crucial in tasks that are non-repetitive, where a different trajectory is needed for every scenario, for instance, in medical surgery, search-and-rescue operations, or ocean and space exploration. In these contexts, the planner could be used to find a reference trajectory to maneuver a robot around obstacles. This trajectory could be latter executed and stabilized in real-time by using a feedback policy. Moreover, planner trajectories could be used to evaluate the robot design in simulation in
order to make sure that it performs properly in different scenarios. In this way, the time and cost expenses of prototyping could be reduced. For example, the planner could conclude that a grasping device is not powerful enough to move a large object, thereby determining that a better design is needed.

- In the game and movie industries, the obtained trajectories could be used to automate the motion of virtual characters, while providing physical realism. For example, a game developer could program a task at a higher level, and the planner would automatically determine the movement of an animated character in an intelligent way. This also becomes useful in computer graphics, for instance, when hundreds of digital actors in a movie should move in an scenario with obstacles. The planner would avoid the time-consuming task of explicitly defining a motion for each actor.

5 Background of the Project Team

The project team involves the full staff of the Kinematics and Robot Design group of IRI, and several of its direct collaborators and postdocs at home or abroad. Together, the group accumulates substantial expertise on the topics involved in this project proposal. The group activities started over 30 years ago. In a series of national projects from 1988 to 2003, the group proposed position analysis tools based on group theory [98], interval techniques [21], Bernstein polynomials [12], multilinear equations [73], and coordinate-free formulations [74]. In the project “A Trajectory Planner for Robotic Systems of Arbitrary Topology” (DPI2004-07358) the group proposed a solver able to deal with robotic systems of general architecture (i.e., involving open or closed kinematic chains connected by means of arbitrary lower-pair joints). The solver was about two orders of magnitude faster than previous methods [72, 75]. Later, in the project “Analysis and motion planning of complex robotic systems” (DPI2007-60585) the technique was extended to deal with robots of a higher complexity, including, in particular, anthropomorphic hands [84]. More recently, in the context of the project “An Extension of Branch-and-Prune Techniques for Motion Analysis and Synthesis of Complex Robotic Systems” (DPI2010-18449), the group developed quite an advanced path planner [45]. This planner was based on modern higher-dimensional continuation techniques, and it was the first one able to efficiently explore configuration space manifolds like those arising when the robot is subject to loop-closure constraints, or geometric constraints imposed by a given task. The group proposed variants of the planner to deal with grasping constraints [85], singularity avoidance constraints [10], and force limit constraints appearing in cable-driven robots [9, 66]. Approximately since 2010, and building on the accumulated experience, the activity of the group has expanded to also cover other key aspects of motion planning. On this regard, it is worth to mention: (a) the development of a new method to generate closure polynomials for motion analysis of planar, spherical, and spatial mechanisms that greatly reduces the number of variable eliminations and avoids the need of trigonometric substitutions [81, 82, 83]; (b) the characterization of singularity-invariant leg rearrangements in fully parallel robots [14, 15]; (c) the design, implementation, path planning and control of parallel robots with lockable or non-holonomic joints [34, 35]; and (d), in the context of the project “RobCab: Control strategies for cable-driven robots for low-gravity simulation” (DPI2014-57220-C2-2-P), a kinodynamic planner for cable-driven robots [13].

6 Formal Problem Statement

The kinodynamic planning problem is typically formulated using the state space of the robot, i.e., the set $\mathcal{X}$ of kinematically-valid states $\mathbf{x} = (\mathbf{q}, \dot{\mathbf{q}})$, where $\mathbf{q}$ is a vector of $n_q$ generalized coordinates describing the configuration of the robot, and $\dot{\mathbf{q}}$ is the time derivative of $\mathbf{q}$, which describes its velocity. The coordinates in $\mathbf{q}$ may be independent or not. In the former case, any $\mathbf{x} = (\mathbf{q}, \dot{\mathbf{q}}) \in \mathbb{R}^{2n_q}$ 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 $\mathbf{q}$ that satisfy a system of $n_c$
nonlinear equations

\[ \Phi(q) = 0 \]  

(1)

ing, 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 and only depend on position variables. By differentiating Eq. (1), the valid values of \( \dot{q} \) are those that fulfill

\[ \Phi_q(q) \dot{q} = 0, \]  

(2)

where \( \Phi_q = \partial \Phi / \partial q \). Likewise, the robot may also be subject to a system of \( n_h \) nonholonomic constraints

\[ A(q) \dot{q} = 0, \]  

(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

\[ \begin{bmatrix} \Phi_q(q) \\ A(q) \end{bmatrix} \dot{q} = B(q) \dot{q} = 0, \]  

(4)

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

Let \( F(x) = 0 \) 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 \( d_X = 2(n_q - n_e) - n_h \) defined implicitly as

\[ \mathcal{X} = \{ x : F(x) = 0 \}. \]  

(5)

Any trajectory 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 motion. These equations can be written in the form

\[ \dot{x} = g(x, u), \]  

(6)

where \( g(x, u) \) is an appropriate differentiable function, and \( u \) is an \( n_u \)-vector of actuator forces subject to lie in a bounded subset \( \mathcal{U} \subset \mathbb{R}^{n_u} \). For each value of \( u \), 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. Since in practice the actuator forces are limited, \( u \) 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 \( \dot{x} = g(x, u) \) at each \( x \in \mathcal{X} \). During its motion, moreover, the robot cannot incur in excessive constraint forces, or in collisions with itself or with the environment, so that the feasible states \( x \) will be those lying in a subset \( \mathcal{X}_{\text{feas}} \subseteq \mathcal{X} \) of non-collision states, where position, velocities, and constraint forces stay within allowable bounds.

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

- The system trajectory \( x = x(t) \) determined by Eqs. (1), (4) and (6) for \( x(0) = x_s \), fulfills \( x(t_f) = x_g \) for some time \( t_f > 0 \), and \( x(t) \in \mathcal{X}_{\text{feas}} \) for all \( t \in [0, t_f] \).
- \( x(t) \) 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

\[ C(x(0), u(t)) = \int_0^{t_f} c(x(t), u(t)) dt \]  

(7)

is, at least, locally optimal for some given instantaneous cost function \( c(x(t), u(t)) \) that models, for example, the time spent or energy consumed along \( x(t) \).
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. Observe however that, contrary to [30, 61] we allow the presence of Eqs. (1) and (4) in the problem, which makes it more general and challenging at the same time. Regarding these equations, it is also worth noting that complex interactions with the environment usually establish and break contacts during motion, either to manipulate an object, to provide extra support to the robot, or to interact with a human. These intermittent contacts change the form of Eqs. (1) and (4) during motion. In this project, however, we shall use the results in [2, 65, 99] to implement a higher-level manipulation planner that can decompose the whole planning problem into a sequence of simple motions in which Eqs. (1) and (4) are kept constant (Fig. 3). In this manner, we will be able to reduce the global planning problem to the planning of the individual constant-constraint motions.

7 State of the Art

Existing strategies for kinodynamic planning can be grouped into decoupled approaches, which search for a C-space path and then design a dynamic trajectory along this path; and direct approaches, which search for such a trajectory right into the state space.

7.1 Decoupled Planning

Decoupled methods concentrate initially on solving the path planning problem, which aims at finding a collision-free path in $C$ (instead of a full time-parametric trajectory in $X$). Algebraic approaches, like those based on silhouettes [20] or cell decompositions [16], are complete methods, i.e., they provide a path if one exists and show failure, otherwise. The former define a roadmap of the C-space, and the latter divide this space into collision-free cells. However, both approaches can only deal with low-dimensional problems. Approximate cell decomposition methods [58] only partially alleviate this issue.

Potential fields have better scalability [54] in comparison. They follow an attractive potential towards the goal, while avoiding repulsive potentials from the obstacles. However, they suffer from falling into local minima. This issue is addressed by the randomized potential planner proposed in [4], where random walks are used to escape from such minima. Nevertheless, potential field methods require a metric to measure the distance from the robot to the obstacles, which is not easy to obtain in general.

Sampling-based methods arise as an alternative, as they only require some means of checking whether a sampled configuration is in collision, instead of the actual distance to the obstacles. Such methods can cope with high-dimensional problems and are probabilistically complete, i.e., they guarantee to find a feasible solution, if one exists and sufficient computing time is available. The

---

Figure 3: A complex motion can be decomposed into simple motions, each of which exhibiting a constant topology in the robot-environment multibody system. The snapshots are representative of simple motions with three, two, and one contacts between the robot and the environment. (Image courtesy of Julia Borràs, Karlsruhe Institute of Technology).
two most popular methods among them are the probabilistic roadmap (PRM) [53] and the rapidly-exploring random tree (RRT) [60] methods. The PRM method takes random samples from the C-space and connects them to form a roadmap. Then, the start and goal configurations are added to this roadmap and a graph search algorithm, such as Dijkstra’s [29], is used to find a path between these configurations. The RRT method incrementally grows a tree rooted at the start configuration by generating random samples. Each sample is connected to the nearest configuration in the tree according to a given metric. The selection of this metric plays a fundamental role in the efficiency of the approach. The search is completed when the tree reaches the goal. Another relevant tree-based algorithm, the expansive search tree (EST) [43] is less metric-dependent. For every node, it simply measures the local density of neighboring nodes, and uses this density to grow the tree towards unsampled areas.

In any case, the complexity of the path planning problem increases when the robot includes kinematic constraints in the form of Eq. (1), as the valid configurations define a manifold embedded in a given ambient space. Algebraic approaches can deal with such manifolds, but do not scale properly [16, 20], or are limited to particular robot architectures [90]. Thus, the usual approach to address these problems is to extend the common sampling-based methods. The performance of those methods heavily relies on being able to uniformly sample the manifold to be explored. In some robotic systems, distance-based formulations provide global parametrizations that can be used to uniformly sample the C-space [37, 96]. However, since global parametrizations are not available in general, alternative sampling strategies have been devised. For example, Han and Amato [36] sample a subset of joint variables and use inverse kinematics to find values for the remaining ones. Unfortunately, this strategy is not applicable to all robotic systems and, although some improvements have been proposed [27], the probability of generating invalid samples is significant. Also, the non-uniqueness of the solutions for the inverse kinematic problem and the presence of singularities complicate the approach [32]. Task-space planners [6, 88, 101] are similar in the sense that they sample a subset of the variables (those related with the end-effector), although they typically determine values for the remaining variables using numerical techniques instead of closed kinematic functions. Thus, they share the problems of kinematic-based approaches regarding the multiple solutions for the non-fixed variables. Another strategy is to sample in the ambient space and then try to converge to the C-space [5, 28, 93, 100, 102]. However, a uniform distribution of samples in the ambient space does not translate into a uniform distribution in the C-space [6]. A better alternative is to sample on an approximation of the constraint manifold, either learned from a collection of valid samples [41], inferred from the nodes of an exploration tree [102], or constructed from tangent-space parametrizations [45, 76, 95]. The latter technique provides better approximation and thus, a more uniform sampling and a more efficient exploration of the manifold.

In the aforementioned approaches, the obtained path is fed into a trajectory generator. Some methods gradually modify this path to obey the dynamic constraints by using optimization techniques [57]. Others find an optimal time scaling for the path subject to dynamic constraints [8, 63, 68, 87, 92]. These methods have been successfully applied to solve complex tasks, such as the coordination of mobile robots [67], or the stabilization of legged robots subject to balance constraints [40, 69]. In any case, decoupled approaches may lead to highly suboptimal solutions involving difficult maneuvers, or even worse, to dynamically unfeasible solutions.

### 7.2 Direct Planning

Under the direct approach, the kinodynamic planning problem becomes much harder because kinematic and dynamic constraints have to be fulfilled simultaneously. Moreover, the planning has to be done in the state space, whose dimension is twice that of the C-space. Existing techniques can be grouped into dynamic programming, optimization, and sampling-based methods.

Dynamic programming approaches search for a solution using a grid of cost-to-go values defined over the state space [3, 25, 64]. The main advantage of this approach is that it can find an optimal solution at the resolution of the grid. Such an approach, however, does not scale well to problems with many degrees of freedom, as the size of the grid grows exponentially in the dimension of the state space.

In contrast, trajectory optimization techniques can be applied to remarkably-complex problems.
They aim to find a trajectory \( x(t), u(t) \) that minimizes a cost function—generally the cost given by Eq. (7)—subject to a set of constraints including the system dynamics. Given an initial condition, \( x_s \), and an input trajectory \( u(t) \) defined over a finite interval, \( t \in [t_0, t_f] \), the basic problem can be formulated as follows:

\[
\begin{align*}
\text{minimize} & \quad \int_{t_0}^{t_f} c(x(t), u(t)) \, dt \\
\text{subject to} & \quad \forall t, \dot{x} = g(x(t), u(t)), \\
& \quad x(t_0) = x_s, \\
& \quad x(t_f) = x_g.
\end{align*}
\]

An advantage of this approach is that constraints of any kind can be added to the previous problem. Its drawbacks are that efficient methods to solve this optimization problem may converge to local minima depending on the initial guess employed, and that the problem size becomes huge for long time horizons or systems with many degrees of freedom [70]. Existing trajectory optimization techniques can be classified into transcription and collocation methods [7].

- **Transcription methods** [48, 86, 103] discretize the trajectory into multiple knot points \( x_1, \ldots, x_N, u_1, \ldots, u_N \), and then enforce the integral of the dynamics between these points as a constraint:

\[
\begin{align*}
\text{minimize} & \quad x_0, \ldots, x_N, u_0, \ldots, u_{N-1} \\
& \quad T_s \sum_{n=0}^{N-1} c(x_n, u_n) \\
\text{subject to} & \quad x_{n+1} = g_d(x_n, u_n) \quad \forall n \in [0, N - 1], \\
& \quad x_0 = x_s, \\
& \quad x_N = x_g.
\end{align*}
\]

Here, \( T_s \) is the time increment employed, and \( g_d(\cdot) \) 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** [38] alleviate this issue by avoiding numerical integration. Both the input \( u(t) \) and state \( x(t) \) trajectories are approximated explicitly by means of polynomial functions. Specifically, \( u(t) \) is described by a first-order polynomial defined by the \( u \) values at the knot points, while \( x(t) \) is described by an Hermitian spline defined by the \( x \) and \( \dot{x} \) values at such points (\( \dot{x} \) being computed using the dynamics in Eq. (6)). Finally, a constraint forces the satisfaction of Eq. (6) at the midpoint of the spline, also known as the collocation point. Thus, the optimization problem can be stated as:

\[
\begin{align*}
\text{minimize} & \quad x_0, \ldots, x_N, u_0, \ldots, u_{N-1} \\
& \quad T_s \sum_{n=0}^{N-1} c(x_n, u_n) \\
\text{subject to} & \quad 0 = h(x_n, u_n, x_{n+1}, u_{n+1}) \quad \forall n \in [0, N - 1], \\
& \quad x_0 = x_s, \\
& \quad x_N = x_g,
\end{align*}
\]

where \( h \) refers to the collocation constraint. This method is powerful enough to be applied to challenging problems involving humanoids [17, 18, 42] and kinematic constraints [78]. In particular, in [78] 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 is 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. In some applications this might be acceptable, but in our constrained systems it would result in unwanted link penetrations, disassemblies, or contact losses.
A widely-used alternative to dynamic programming and optimization is to rely on kinodynamic sampling-based methods [26, 59]. These methods can cope with high-dimensional problems, and are probabilistically complete. Moreover, recent methods can even generate globally optimal trajectories [39, 50, 51, 62]. The kinodynamic RRT [61] and EST [44] methods stand out among them, due to their effectiveness and conceptual simplicity, since they mainly rely on motion simulation. However, it is well known that these planners can be inefficient in certain scenarios [22]. Part of the complexity arises from planning in the state space instead of in the lower-dimensional C-space [70]. The RRT method is easier to implement, but its main issue is the disagreement of the metric used to measure the distance between two states, and the actual cost of moving between such states, which must comply with the vector fields defined by Eq. (6) [23, 24, 46, 49, 56, 71, 89, 94]. However, none of the previous methods can directly deal with the implicitly-defined state spaces given by Eq. (5).

8 Methodology

Our planner will be based on sampling-based techniques, as these generally work well in high-dimensional problems. Moreover, they can easily cope with the many constraints of the problem, and with any integration method in principle. In particular, we envisage the extension of the classic kinodynamic RRT [61] and EST [44] methods to also deal with constrained robotic systems. To have an idea, we next see how this extension can be achieved with the method in [61]. The extension of [44] would be similar, only requiring adjustments in the expansion heuristics employed.

The planner in [61] assumes that \( \mathcal{X} \) is parametrically defined, i.e., that all tuples \( \mathbf{x} = (\mathbf{q}, \dot{\mathbf{q}}) \) are possible in principle. The planner looks for the desired trajectories \( \mathbf{u}(t) \) and \( \mathbf{x}(t) \) by constructing an exploration RRT over \( \mathcal{X} \), which in this case is \( \mathbb{R}^{2n_q} \). The RRT is rooted at \( \mathbf{x}_s \), and it is grown incrementally towards \( \mathbf{x}_g \) while staying inside \( \mathcal{X}_{\text{feas}} \). Every tree node stores a feasible state \( \mathbf{x} \in \mathcal{X}_{\text{feas}} \), and every edge stores the action \( \mathbf{u} \in \mathcal{U} \) needed to move between the connected states. This action is assumed to be constant during the move. The expansion of the RRT proceeds by applying three steps repeatedly (Fig. 4, top-left). First, a state \( \mathbf{x}_{\text{rand}} \in \mathcal{X} \) is randomly selected; then, the RRT state \( \mathbf{x}_{\text{near}} \) that is closest to \( \mathbf{x}_{\text{rand}} \) is computed according to some metric; finally, a movement from \( \mathbf{x}_{\text{near}} \) towards \( \mathbf{x}_{\text{rand}} \) is performed by applying an action \( \mathbf{u} \in \mathcal{U} \) during a fixed time \( \Delta t \). The movement from \( \mathbf{x}_{\text{near}} \) towards \( \mathbf{x}_{\text{rand}} \) is simulated by integrating Eq. (6) numerically, which yields a new state \( \mathbf{x} \) that may or may not be in \( \mathcal{X}_{\text{feas}} \). In the former case \( \mathbf{x} \) is added to the RRT, and in the latter it is discarded. To test whether \( \mathbf{x} \in \mathcal{X}_{\text{feas}} \), \( \mathbf{x} \) is checked for collisions by using standard algorithms [47], and the joint positions, velocities and constraint forces are computed to check whether they stay within bounds. The action \( \mathbf{u} \) applied is typically chosen as the one from \( \mathcal{U} \) that brings the robot closer to \( \mathbf{x}_{\text{rand}} \). One can either try all possible values in \( \mathcal{U} \) (if it is a discrete set) or only those of \( n_s \) random points on \( \mathcal{U} \) (if it is continuous). To force the RRT to extend towards \( \mathbf{x}_g \), \( \mathbf{x}_{\text{rand}} \) is set to \( \mathbf{x}_g \) once in a while, stopping the whole process when a RRT leaf is close enough to \( \mathbf{x}_g \). Usually, however, a solution trajectory can be found more rapidly if two RRTs respectively rooted at \( \mathbf{x}_s \) and \( \mathbf{x}_g \) are grown simultaneously towards each other (Fig. 4, left-bottom). The expansion of the tree rooted at \( \mathbf{x}_g \) is based on the integration of Eq. (6) backward in time.

The previous strategy is easy to implement when \( \mathcal{X} = \mathbb{R}^{2n_q} \), but in our case \( \mathcal{X} \) could be a \( d_\mathcal{X} \)-dimensional manifold defined implicitly by Eqs. (1) and (4). This complicates matters substantially, because there is no straightforward way to randomly select points \( \mathbf{x} = (\mathbf{q}, \dot{\mathbf{q}}) \) satisfying Eqs. (1) and (4), and the numerical integration of Eq. (6) easily drifts away from \( \mathcal{X} \) when standard methods for ordinary differential equations are used. In this project we shall circumvent these two issues by constructing an atlas of \( \mathcal{X} \) in parallel to the RRT.

An atlas is a collection of charts mapping \( \mathcal{X} \) entirely, where each chart is a local diffeomorphism \( \psi \) from an open set \( P \subseteq \mathbb{R}^{d_\mathcal{X}} \) of parameters to an open set \( V \subset \mathcal{X} \) (Fig. 4, right). The \( V \) sets can be thought of as partially-overlapping tiles covering \( \mathcal{X} \), in such a way that every \( \mathbf{x} \in \mathcal{X} \) lies in at least one set \( V \). Such an atlas will be created with the tools in [45]. With such an atlas, the problem of sampling \( \mathcal{X} \) boils down to generating random values \( \mathbf{y} \) in the \( P \) sets, since these values can always be projected to \( \mathcal{X} \) using \( \mathbf{x} = \psi(\mathbf{y}) \). Also, the atlas allows the conversion of the vector field defined on \( \mathcal{X} \) by Eq. (6) into one in the coordinate spaces \( P \), which permits the integration of Eq. (6) using local
coordinates \[79\]. As a result, the RRT motions satisfy Eqs. (1) and (4) by construction, eliminating any drift from \(X\) to machine precision.

One could build a full atlas of the implicitly-defined state space and then use its local parameterizations to define a kinodynamic RRT. However, the construction of a complete atlas is only feasible for low-dimensional state spaces. Moreover, only part of the atlas is necessary to solve a given planning problem. Thus, we shall combine the construction of the atlas and the expansion of the RRT. In this approach, a partial atlas is used to generate random states and to add branches to the RRT. The atlas is initialized with two charts covering \(x_s\) and \(x_g\), respectively (Fig. 4, right). Then, these charts are used to pull the expansion of the RRT, which in turn adds new charts to the atlas as needed, until \(x_s\) and \(x_g\) become connected.

9 Work Plan

9.1 Task Decomposition

The project tasks are grouped into four work packages:

- Work package 1 ("Organization and kick off"): Tasks 1 and 2.
- Work package 2 ("Algorithmic Tools"): Tasks 3 to 7.
- Work package 3 ("Implementation and experiments"): Tasks 8 to 13.
- Work Package 4 ("Final Demonstrations and Dissemination"): Tasks 14 to 16.

Work package 1 encompasses basic organization and kick off tasks. Work package 2 develops the theoretical tools described in Section 8. Workpackage 3 consists in the software implementation of such tools, and in their validation both in simulation and on real benchmark experiments involving a real robotic torso (see Section 9.2 below). Workpackage 4 encompasses the final demonstrations and all dissemination tasks of the project. We next detail the contents of each work package.
Work package 1: Organization

**Work package coordinators:** Lluís Ros and Federico Thomas.

**Work package description:** Organization meetings and hiring of a mechatronics engineer.

**Task 1: Organization meetings**
- **Researchers in charge:** Lluís Ros and Federico Thomas.
- **Team members involved:** All of the team members.
- **Objective:** To keep the team organized. After an initial kick-off meeting, work package coordinators will prepare small and global meetings along the project, to keep the team organized and up to date with all the developments.

**Task 2: Hiring of a mechatronics engineer**
- **Researchers in charge:** Lluís Ros and Federico Thomas
- **Team members involved:** All of the team members.
- **Objective:** To hire a mechatronics engineer to support the project team. This engineer will be in charge of the set up, maintenance and low-level programming of the robot torso applied for in this project, and of the design and construction of the gripper tools needed in the various experiments (see Section 9.2). See also the project budget.

Work package 2: Algorithmic tools

**Work package coordinators:** Federico Thomas and Lluís Ros.

**Work package description:** Theoretical design of the planner algorithms.

**Task 3: Design of a dynamics engine**
- **Researcher in charge:** Lluís Ros
- **Other team members involved:** Josep M. Porta, Vicente Ruiz de Angulo, and Ricard Bordalba
- **Objective:** To design the dynamics engine necessary to support the motion planning methods of Section 8. Existing off-the-shelf engines do not cope well with implicitly-defined state spaces. The design of a proper engine for the project entails choosing a proper set of configuration coordinates, implementing Featherstone’s articulated body inertia methods [31], developing local coordinate methods for the integration of motion equations on manifolds [79], and using [1] to deal successfully with singularities. Mechanism symmetries will be taken into account to make the engine as efficient as possible [33].

**Task 4: Design of planning strategies based on RRTs and ESTs**
- **Researcher in charge:** Josep M. Porta
- **Other team members involved:** Lluís Ros, Enric Celaya, Ricard Bordalba.
- **Objective:** To develop the core motion planning strategies on which the planner will rely. In particular, this task will generalize the classical RRT and EST planning methods to also cope with systems with kinematic loop-closure constraints, as described in Section 8.

**Task 5: Trajectory optimization methods**
- **Researcher in charge:** Vicente Ruiz de Angulo
- **Other team members involved:** Enric Celaya, Josep M. Porta, and Nicolás Rojas.
- **Objective:** To smooth and optimize the position, velocity and acceleration trajectories resulting from the planner. Due to the randomized nature of the methods, these trajectories might be initially jerky, or suboptimal. Smooth, locally optimal trajectories could be obtained by feeding the output of the planner into optimization approaches like [19] or [39].

**Task 6: Computation of constraint forces**
- **Researcher in charge:** Lluís Ros
- **Other team members involved:** Enric Celaya, Júlia Borràs, and Alba Pérez.
Objective: To extend the dynamic engine so as to be able to compute, for each state undergone by the robot, all constraint forces transmitted at the various joints. This is necessary to ensure that internal forces suffered by the robot stay within admissible bounds along a trajectory.

Task 7: Handling of nonholonomic constraints  
Researcher in charge: Federico Thomas  
Other team members involved: Patrick Grosch and Julia Borràs  
Objective: To include the handling of nonholonomic constraints into the planner. These constraints can be included in Eq. (3), and then the robot motion can be simulated using Eq. (6).

Work package 3: Implementation and experiments

Work package coordinators: Josep M. Porta and Enric Celaya.  
Work package description: Implementation and experimental validation of the planner.

Task 8: Basic planner implementation  
Researcher in charge: Josep M. Porta  
Other team members involved: Ricard Bordalba, Nicolás Rojas, Julia Borràs.  
Objective: To implement an initial planner module allowing to solve planning queries without nonholonomic constraints, force limit constraints, or intermittent contacts. This module implements the algorithms designed in tasks 3, 4, and 5. The module will be tested using benchmark “B1” in simulation (see Section 9.2).

Task 9: Set up of experiments  
Researcher in charge: Federico Thomas  
Other team members involved: Hired engineer, Patrick Grosch, and Alba Pérez  
Objective: To set up the robot torso in the different experimental scenarios. For each experiment, this includes the mounting of the torso on its base, the design and construction of the grasping devices needed to interact with the environment, the environment itself, and any security elements needed.

Task 10: Low-level control module  
Researcher in charge: Federico Thomas  
Other team members involved: Hired engineer, Patrick Grosch, and Lluís Ros  
Objective: To develop a low-level control module to execute the planned motions with the real torso. Although we plan to maximally exploit the built-in controllers of the acquired torso (see the project budget), these might need redesigns or adjustments to better take the dynamic model into account. The module will be tested using benchmark “B1” in simulation (see Section 9.2), and then on the real torso.

Task 11: Internal force computation module  
Researcher in charge: Josep M. Porta  
Other team members involved: Hired engineer, Enric Celaya, Júlia Borràs, and Alba Pérez.  
Objective: To implement the methods resulting from Task 6. This module will then be used by the planner to avoid surpassing any internal force limits applicable to the robot. This module will be tested using benchmark “B1” (see Section 9.2), taking the force limits of the robot into account.

Task 12: Nonholonomic constraints module  
Researcher in charge: Vicente Ruiz de Angulo  
Other team members involved: Hired engineer, Federico Thomas, Patrick Grosch, and Júlia Borràs.  
Objective: To implement the methods resulting from Task 7. The resulting module will allow the planner to take nonholonomic constraints into account. This module will be tested using benchmark “B2” (see Section 9.2).
Task 13: Motion decomposition module

Researcher in charge: Vicente Ruiz de Angulo

Other team members involved: Hired engineer, Júlia Borràs and Enric Celaya.

Objective: To implement a higher-level planner based on [65, 99], to be able to decompose a complex motion involving intermittent contacts, into simple motions with constant multibody system topology. This module will be tested using benchmark “B3” (see Section 9.2).

Work package 4: Final demonstrations and dissemination

Work package coordinators: Vicente Ruiz de Angulo and Josep M. Porta.

Work package description: Final demos, documentation and dissemination tasks.

Task 14: Final demonstrations

Researcher in charge: Vicente Ruiz de Angulo

Other team members involved: Hired engineer, Ricard Bordalba, Daniel González.

Objectives: To demonstrate the developed technology before Beta robots, the company interested in the project results. To study and discuss in some depth any potential industrial applications.

Task 15: Software and documentation deliverables

Researchers in charge: Lluís Ros and Federico Thomas

Other team members involved: The rest of the team

Objectives: To clearly document the software library developed and to provide a users’ manual for it. To openly distribute this documentation and the software, together with a tutorial, through a dedicated website at IRI.

Task 16: Conference and journal publications

Researchers in charge: Lluís Ros and Federico Thomas

Other team members involved: The rest of the team.

Objective: To disseminate the scientific and technological achievements of the project through internationally recognized conferences and specialised journals. A wrap-up paper summarizing the project results will also be prepared for the IEEE Robotics and Automation Magazine.

9.2 Benchmarks

The planner will be able to deal with general multibody systems, but for validation purposes we shall apply it to the following benchmark problems. They all involve a two-limbed robot torso equipped with grasping devices tailored to the needs of each experiment (see Task 9). The resulting trajectories will first be validated in simulation, and then they will be executed on the real torso:

B1 “Swing-up pick-and-place task”: In this experiment, the robot torso will have to perform a motion similar to the one in Fig 1, bottom. With its base fixed, the torso will have to pick up a heavy load from a given location, and it will have to deposit it on a destination shelf in a higher position. The load will be large enough so as to force the torso to swing it back and forth, in order to gain momentum to reach the goal.

B2 “Object-in-tray transportation”: The torso will grasp a tray with its both hands. An object will be positioned on a given location in the tray. The goal of the robot will be to move the tray to a second location, keeping the object safely inside the tray during the whole motion. The object will be a cylinder, or a ball, or any other object establishing a nonholonomic contact with the tray. Collisions with the tray walls will have to be avoided. This experiment will allow us to validate the planner in situations involving nonholonomic constraints.

B3 “Monkey-bar exercise”: In this third experiment the torso will have to emulate the motion shown in Fig. 2. A ladder will be installed horizontally on the ceiling of the lab, and the robot torso will have to traverse it from one end to the other using pendulum-like motions.
9.3 Project Milestones and Temporal Chart

The following milestones will be used as project indicators:

- **MS1**: Design of basic planning algorithms completed.
- **MS2**: Design of advanced planning algorithms completed
- **MS3**: Basic planner module implemented and tested in simulation.
- **MS4**: Low-level control module implemented and tested for the 1st benchmark.
- **MS5**: 1st benchmark executed successfully on the real torso without constraint force bounds.
- **MS6**: 1st benchmark executed successfully on the real torso with constraint force bounds.
- **MS7**: 2nd benchmark executed successfully on the real robot torso.
- **MS8**: 3rd benchmark executed successfully on the real robot torso.
- **MS9**: Final demonstrations before Beta robots done.
- **MS10**: Software and documentation deliverables finished.

The diagram in Fig. 5 shows the temporal distribution of all tasks and milestones of the project.
References


C.2: IMPACTO ESPERADO DE LOS RESULTADOS

Since the early days of robotics, motion planners of all kinds have been developed. Until recently, however, most of them have disregarded the dynamics of the problem. Taking the dynamics into consideration is mandatory to plan trajectories that are effectively executable by a specific robot. The scientific challenge taken in this project is to contribute to the solution of the complete kinodynamic motion planning problem, so as to allow a robot to perform its tasks in an energy-efficient and agile way. The envisaged theoretical developments are expected to significantly push the state of the art forward in motion planning. The treatment of dynamic systems with closed kinematic chains has been identified as a main open problem in robotics [52], and thus, the project developments would imply a significant scientific advance at the international level.

The ability to generate energy-efficient robot trajectories is another aspect of the project that could have a significant impact in the general development of robotics. Increasing the efficiency of a robot widens its possible applications and reduces its energy storage needs, and thus allows a cost reduction both in its operation and construction. The robot, as a result, can be designed with less powerful actuators and lighter components.

Another important impact of the project may take place in the area of human-robot interaction. Robots showing movements that are perceived as mechanical or rude by a human are difficult to interact with, and have lesser opportunities to be integrated in our homes and daily lives. Note that humans tend not to feel confident with them in the measure in which they are perceived as extraneous artifacts with somewhat unpredictable actions. Instead, if a robot showed a human-like behavior, in the sense that its motions looked natural and similar to those we would do when performing a similar task, we would be more prone to interact and stay confident with it. This kind of behavior is just what is expected from the energy-efficient and agile motions that we intend to plan in this project.
C.3: CAPACIDAD FORMATIVA DEL EQUIPO SOLICITANTE

The group has a long experience advising Master and PhD students. Some of the group members have long been collaborating in the Robotics Master and PhD programmes of the UPC, coordinating and teaching courses in these programmes. In recent years, the group members have supervised several Master theses by students from the Industrial Engineering School and the Computer Science and Mathematics Faculties at UPC (Barcelona), and from the Ecole Nationale Supérieure de l’Aéronautique et de l’Espace (Toulouse).

The group members have advised several PhD theses whose quality is proved by the high quality publications derived from them. Notably, the thesis of Oriol Bohigas got the award to the 2013 Spanish Best Thesis in Robotics, an award given in the XXXV Jornadas de Automatica held in Valencia and organized by the Comité Español de Automática. Moreover, the thesis of Carlos Rosales was finalist in this same event.

Besides advisory support, the group can grant enough computing facilities, robots, sensors, and a robotics lab with other equipment which, combined with its expertise and motivation, provides a fertile environment for pursuing leading-edge research. The group can also provide an extensive network of solid academic links with foreign institutions, allowing to complement the students’ formation with research stages abroad. Among the most relevant connections we cite those with the Robotics Group of the LAAS-CNRS lab in Toulouse (France), the Robotics Group of the Mechanical Engineering Department at University of Leuven (Belgium), the Robotics Group at University of Amsterdam (The Netherlands), the Robotics Group at University of Cassino (Italy), the Computational Geometry Group at University of Tokyo (Japan), the Structural Topology Group at York University (Canada), the Robotics Group at University of Illinois at Urbana-Champain (USA), and the Institute for the Research in Communications and Cybernetics in Nantes (France). The group members have been actively collaborating with such institutions through pre- or post-doc stays, within research projects funded by the European Community, or within bilateral programmes. We next list the PhD theses advised recently by the group members:

- **Student:** Júlia Borràs. **Advisor:** Federico Thomas. **Year:** 2011. **Thesis:** Singularity-invariant leg rearrangements in Stewart-Gough platforms. **Current position:** Postdoc at the Karlsruhe Institute of Technology, Germany.

- **Student:** Alejandro Agostini. **Advisor:** Enric Celaya. **Year:** 2011. **Thesis:** Q-Learning with a degenerate function approximation. **Current position:** Postdoc at the University of Gottingen, Germany.

- **Student:** Nicolás Rojas. **Advisor:** Federico Thomas. **Year:** 2012. **Thesis:** Distance-based formulations for the position analysis of kinematic chains. **Current position:** Lecturer at Imperial College London, USA.

- **Student:** Carlos Rosales. **Advisors:** Lluís Ros, Raúl Suárez. **Year:** 2013. **Thesis:** Grasp planning under task-specific contact constraints. **Current position:** Co-founder and Chief Technology Officer of Beta Robots.

- **Student:** Oriol Bohigas. **Advisors:** Lluís Ros, Montserrat Manubens. **Year:** 2013. **Thesis:** Numerical computation and avoidance of manipulator singularities. **Current position:** Co-founder and Chief Science Officer of Beta Robots.

- **Student:** Patrick Grosch. **Advisor:** Federico Thomas. **Year:** 2016. **Thesis:** Parallel robots with unconventional joints to achieve under-actuation and reconfigurability. **Current position:** Postdoc at IRI.

- **Student:** Aleix Rull. **Advisors:** Federico Thomas, Josep M Porta. **Year (expected):** 2018. **Thesis:** Distance bound smoothing under orientation constraints.

- **Student:** Ricard Bordalba. **Advisors:** Lluís Ros, Josep M Porta. **Year (expected):** 2020. **Thesis:** Motion planning under differential constraints.

C.4: IMPLICACIONES ÉTICAS Y/O DE BIOSEGURIDAD

This project does not raise any ethic or bio-security issue.