View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by ScholarlyCommons@Penn

University of Pennsylvania
ScholarlyCommons

IRCS Technical Reports Series

Institute for Research in Cognitive Science

December 1996

Continuous Methods for Motion Planning

Milos Zefran
University of Pennsylvania

Follow this and additional works at: http://repository.upenn.edu/ircs_reports

Zefran, Milos, "Continuous Methods for Motion Planning" (1996). *IRCS Technical Reports Series*. 111.
http://repository.upenn.edu/ircs_reports/111

University of Pennsylvania Institute for Research in Cognitive Science Technical Report No. IRCS-96-34.

This paper is posted at ScholarlyCommons. http://repository.upenn.edu/ircs_reports/111
For more information, please contact libraryrepository@pobox.upenn.edu.

Continuous Methods for Motion Planning

Abstract
Motion planning for a robotic system addresses the problem of finding trajectory and actuator forces that are consistent with a given set of constraints and perform a desired task. In general, the problem is under determined and admits a large number of solutions. The main claim of this dissertation is that a natural way to resolve the indeterminacy is to define performance of a motion and find a solution with the best performance. The motion planning problem is thus formulated as a variational problem. The proposed approach is continuous in the sense that the motion planning problem is not discretized.

A distinction is made between dynamic and kinematic motion planning. Dynamic motion planning provides the actuator forces as part of the motion plan and requires finding a motion that is consistent with the dynamic equations of the system, satisfies a given set of equality and inequality constraints, and minimizes a chosen cost functional. In kinematic motion planning, dynamic equations of the system are not taken into account and it is therefore simpler.

For dynamic motion planning, a novel numerical method for solving variational problems is developed. The continuous problem is discretized by finite element methods and techniques from nonlinear programming are used to solve the resulting finite dimensional optimization problem. The method is employed to find smooth trajectories and actuator forces for two planar cooperating manipulators holding an object. The computed trajectories are used to model human motions in a two-arm manipulation task. The method is then extended for systems that change the dynamic equations as they move. An example of a simple grasping task illustrates that for such systems variational approach unifies motion planning and task planning.

Kinematic motion planning is formulated as a variational problem on the group of spatial rigid body displacements, *SE(3)*. A Riemannian metric and an affine connection are introduced to define cost functionals that measure smoothness of trajectories. Metrics and connections that are important for kinematic analysis are identified. It is then shown how the group structure of *SE (3)* can be used to find smooth trajectories that satisfy boundary conditions on positions, orientations, velocities or their derivatives, and have certain invariance properties with respect to the choice of the inertial and body fixed frames.

Comments
University of Pennsylvania Institute for Research in Cognitive Science Technical Report No. IRCS-96-34.

This thesis or dissertation is available at ScholarlyCommons: http://repository.upenn.edu/ircs_reports/111

Institute for Research in Cognitive Science

Continuous Methods for Motion Planning

Milos Zefran

University of Pennsylvania
3401 Walnut Street, Suite 400A
Philadelphia, PA 19104-6228

December 1996

Site of the NSF Science and Technology Center for Research in Cognitive Science

IRCS Report 96--34

CONTINUOUS METHODS FOR MOTION PLANNING

Miloš Žefran

A Dissertation
in

Computer and Information Science

Presented to the Faculties of the University of Pennsylvania in Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy.

1996

Prof. Vijay Kumar
Supervisor of Dissertation

Prof. Mark Steedman
Graduate Group Chairperson

Mojima staršema Nadi in Srečku ter bratu Rastku.

Acknowledgment

It is difficult to mention all the people who come to mind when I look upon the path that led to this work. Above all, I would like to express my admiration and gratitude to Professor Vijay Kumar for being my advisor and such an inspiring research collaborator. He never hesitated to work with me on a last-minute paper submission or to patiently listen when I tried to explain my often confused ideas. And he was a fun conference companion.

I am equally indebted to Professor Chris Croke. He introduced me to differential geometry and was prepared to get involved into research that was quite marginal to his research interests. Collaboration with him was one of the most satisfactory experiences in my graduate school.

I would also like to thank the members of my committee, Professors Ruzena Bajcsy, Max Mintz, Richard Murray, and Xiaoping Yun, for reading through the dissertation and making a number of helpful suggestions. Further, I am grateful to Professor Bajcsy, director of the GRASP Lab, for bringing me into the laboratory and for maintaining such a unique environment there. I thank Professor Mintz for many pleasant chats, be they about research or other things. Professor Murray kindly answered questions in my mails, and Professor Yun introduced me to the research in the GRASP Lab when I first came there.

In the last four years, GRASP Lab was my second home and I would like to thank all of its current and past denizens for making it such a vibrant place. Camaraderie of Luca Bogoni, Ulf Cahn von Seelen, Hany Farid, Ioannis Kakadiaris, Jana Košecká and the rest of the crowd in and out of the Lab made these years the experience of a lifetime. I assure everybody in the Lab that I did not work late at night to hide from them! Brian Madden was an entertaining officemate, Venkat Krovi was always ready for a discussion and kept me in touch with the social life at the University, Stamps Howard and Jaydev Desai were great coworkers, and Nilanjan Sarkar, Yoshio Yamamoto and Chau-Chang Wang gave me many wise advices about research.

These years in Philadelphia were very special thanks to Barbara Di Eugenio. Her attention and patience kept me going when I struggled with the research. And the wonderful trips we took together were the best imaginable way of spending the free time. She also kindly corrected many grammatical mistakes in the text.

I am grateful to my roommate Evangelos Kokkevis for his friendship and for reminding me that there is life beyond school. I appreciate his willingness to put up with my cooking. And I thank him for his help with the computer animations.

When I first started my research back at the University of Ljubljana, I found a wonderful mentor and friend in Professor Tadej Bajd. I thank him for all he has done for me.

I gratefully acknowledge the Institute for Research in Cognitive Science at the University of Pennsylvania for providing most of my funding during graduate school.

And finally, I would like to thank my parents, my brother, and his family for their encouragement and support during my long absence from home. This work is dedicated to them.

Abstract

CONTINUOUS METHODS FOR MOTION PLANNING

Miloš Žefran

Vijay Kumar

Motion planning for a robotic system addresses the problem of finding a trajectory and actuator forces that are consistent with a given set of constraints and perform a desired task. In general, the problem is under-determined and admits a large number of solutions. The main claim of this dissertation is that a natural way to resolve the indeterminacy is to define performance of a motion and find a solution with the best performance. The motion planning problem is thus formulated as a variational problem. The proposed approach is continuous in the sense that the motion planning problem is not discretized.

A distinction is made between dynamic and kinematic motion planning. Dynamic motion planning provides the actuator forces as part of the motion plan and requires finding a motion that is consistent with the dynamic equations of the system, satisfies a given set of equality and inequality constraints, and minimizes a chosen cost functional. In kinematic motion planning, dynamic equations of the system are not taken into account and it is therefore simpler.

For dynamic motion planning, a novel numerical method for solving variational problems is developed. The continuous problem is discretized by finite-element methods and techniques from nonlinear programming are used to solve the resulting finite-dimensional optimization problem. The method is employed to find smooth trajectories and actuator forces for two planar cooperating manipulators holding an object. The computed trajectories are used to model human motions in a two-arm manipulation task. The method is then extended for systems that change the dynamic equations as they move. An example of a simple grasping task illustrates that for such systems variational approach unifies motion planning and task planning.

Kinematic motion planning is formulated as a variational problem on the group of spatial rigid body displacements, $SE(3)$. A Riemannian metric and an affine connection are introduced to define cost functionals that measure smoothness of trajectories. Metrics and connections that are important for kinematic analysis are identified. It is then shown how the group structure of $SE(3)$ can be used to find smooth trajectories that satisfy boundary conditions on positions, orientations, velocities or their derivatives, and have certain invariance properties with respect to the choice of the inertial and body fixed frames.

Contents

List of Figures

Chapter 1

Introduction

In a basic robotic task, the robot has to move from a given initial configuration to a desired final configuration. Except for some degenerate cases, there will be infinitely many motions that achieve the desired transition. More complex tasks impose additional requirements on the motion, but in general the set of all possible motions will still be very large. A robotic task typically also entails manipulation of an object and interactions with the environment. To prevent damage to the manipulated object or to the robot, the manipulation and interaction forces have to be controlled. The purpose of motion planning is to select one motion from the set of all possible motions and to assure that the manipulation and interaction forces remain within given bounds.

The relation between a (kinematic) trajectory of a physical system and the actuator[1] forces that generate the motion along this trajectory is given by dynamic equations. In general, a motion plan thus consists of the kinematic trajectory for the system as well as the actuator forces that move the system along the trajectory. The actuator forces can be sometimes obtained from the given kinematic trajectory in a particularly simple way. In other instances we use the kinematic description of the system because the dynamic equations are difficult to derive. There are also cases when we employ the kinematic model of the system to abstract the details of the actuation scheme. In all these situations, we only need to plan the kinematic trajectory for the system. We will thus use the term *dynamic motion planning* when the actuator forces are part of the computed motion plan and *kinematic motion planning* when only the kinematic trajectory for the system is computed.

We say that motion planning is *explicit*, if the motion plan is computed before the motion is executed, and *implicit*, if the trajectory and the actuator forces are computed while the system moves. Explicit schemes must be used if we want to optimize the performance of the motion or guarantee certain properties of the trajectory. If it only matters that a desired configuration is reached, implicit schemes are sufficient.

Motion of a physical system can be observed in three different spaces. The task is specified in the *task space*. The representation for this space is usually chosen to simplify the effort of the human in describing the task. In most applications, Cartesian coordinates are used to specify the position in space and some form of Euler angles is used to specify the orientation. Task space description is convenient for humans, but to properly represent

[1] Throughout the dissertation, the term *actuator force* will be used to denote the generalized force consisting of forces and torques produced by the actuators.

a motion of a system with multiple degrees of freedom, it is more appropriate to specify the motion in the *joint space* (in dynamics, the term used is *configuration space*). The joint space is the Cartesian product of the intervals describing the allowable range of motion for each degree of freedom. To achieve the motion, some degrees of freedom must be actuated. We must therefore compute the actuator forces that achieve the desired motion and these forces belong to the *actuator space*. An actuator is usually controlled by a single input (current or voltage in the case of electro-mechanical actuators), so the actuator space will be typically a subset of $\mathbb{R}^m$, where m is the number of actuators.

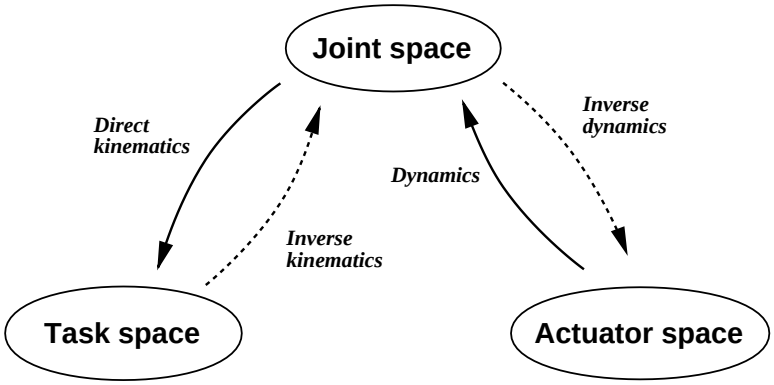


Figure 1.1: Spaces in which motion can be observed and mappings between them.

Mappings between the task space, the joint space and the actuator space (Figure 1.1) may not be invertible. For example, in the case of a kinematically redundant robot, there is a mapping from the joint space to the task space, but not vice-versa: a trajectory in the task space does not specify a unique trajectory in the joint space. Similarly, if the manipulator has actuator redundancy, the mapping from the actuator space to the joint space exists, but it does not in the other direction. In general, a curve in the actuator space can always be mapped to a curve in the joint space and a curve in the joint space can always be mapped to a curve in the task space. The mapping from the actuator space to the joint space is given by the equations of motion of the system and thus involves integration of a set of differential equations. In contrast, the mapping from the joint space to the task space is algebraic, although it might be still difficult to obtain (if the mechanism involves parallel modules). This implies that a trajectory in the actuator space completely characterizes the motion. However, for applications we usually also need the task space and the joint space trajectories and we require that they are part of the motion plan. It is also clear that with the exception of special cases in which the mappings between different spaces are invertible, the planner must provide the trajectory in the actuator space to completely specify the motion.

Motion planning has received much attention in the past. A short review of the literature is provided in the next two sections. Since humans seem to move in a very efficient way and since we often try to imitate humans by robots, we start with the models of motion planning in humans. We then discuss algorithms that were developed for solving the discretized motion planning problem. Discretization reduces the infinite-dimensional space of possible solutions to a finite-dimensional search space. One of the advantages of discrete algorithms is that they can address the question of completeness: if a solution in

the reduced space exists, the algorithm will find one; if there is no solution, the algorithm will return the negative answer. For continuous methods, that we describe next, the question of completeness is much more difficult and in general completeness can not be proved. Many continuous motion planning methods are based on optimal control. However, these methods were used to address very specific motion planning problems. At the end of the chapter, we argue that optimal control and variational methods represent a natural framework for motion planning and we propose them as a general method. We then formally define the dynamic and the kinematic motion planning problems that are addressed in the dissertation.

1.1 Motion planning in humans

Fascinating photographs by Muybridge, originally published at the end of last century [94, 95], are one of the earliest attempts to systematically study human and animal motion. In the '60s, developments in control theory sparked interest in modeling the organizing principles of human motor control. The models that were developed in the subsequent years can be divided into two major groups [55, 66, 136]. Models in the first group assume that the motion is generated through a feedback control law [24, 133]. Since the trajectory for the motion is not computed in advance, these models are examples of implicit planning. In contrast, models in the second group propose explicit schemes for motion planning and assume that the trajectory is planned in advance and then executed [38, 42, 143]. Experimental evidence exists for the feedback mechanisms as well as for the explicit trajectory planning, and none of the existing models seems to be able to account for all the characteristics of human motion. Nonetheless, there is enough evidence to believe that mechanisms for explicit motion planning do exist in the central nervous system.

The human body can be modeled as an articulated linkage of rigid bodies. This is also how we usually model robots. Principles that govern human motion planning therefore directly bear on motion planning for robotic systems. Redundancies that make the mappings between task, joint, and actuator spaces non-invertible are abundant in the human body (as well as other biological systems) suggesting that humans posses mechanisms for resolving these redundancies. Studies of human motion therefore represent an important source of ideas for robot motion planning schemes. Of particular relevance to robotic applications are studies of human arm motions.

An important quantitative property of human target directed motions was discovered by Fitts [41]. Known as Fitts' Law, it relates the movement time to the traveled distance and the size of the target. Invariant properties of human arm trajectories were studied by Soechting and Lacquanity [133] and Viviani *et al.* [145]. In [133], it was observed that the trajectory for spatial motions is independent of movement velocity and that ratios of velocities of elbow and shoulder joints are fixed in the acceleratory and decceleratory phases. Two-thirds power law was formulated in [145] to describe the relation between the radius of curvature of the trajectory and the tangential velocity. Morasso [90] showed that the shape of the velocity profiles for human arm trajectories is invariant in the extracorporal (task) space but not in the joint space and concluded that planning takes place in the extracorporal space.

Feldman [38] focussed on motions involving only the elbow joint. He treated the muscles

as nonlinear springs and showed that humans are able to modulate a parameter that corresponds to the zero-length (equilibrium point) of the spring. Based on these observations, he postulated the so called equilibrium point hypothesis for the generation of voluntary arm motions. According to this hypothesis, to reach a goal configuration the equilibrium point of the system of springs representing muscles of the human arm is instantaneously shifted to the goal configuration. The arm subsequently moves because of the stiffness of the system.

The hypothesis that the system only specifies the target equilibrium point was refuted by Bizzi *et al.* [10]. They showed that if the arm of a deafferented monkey is displaced from the initial position, it moves towards the initial position when the target at the final position is illuminated and only then changes the direction of motion and moves towards the target. This shows that a series of intermediate positions is planned and implies that the final-position control (according to which the motion is driven by the error between the current position and the goal position) and the equilibrium point hypotheses are not sufficient to describe goal directed motions. This study provides the strongest evidence for explicit trajectory planning.

The results of Bizzi *et al.* and the apparent "ease" and "efficiency" of human movements prompted Flash and Hogan [42] to suggest that trajectories for human planar reaching motions are chosen so that they minimize the integral of the square norm of jerk (jerk is the derivative of acceleration):

$$J = \frac{1}{2} \int_{t_0}^{t_1} (\dddot{x}^2 + \dddot{y}^2)\, dt. \tag{1.1}$$

In the equation, t_0 and t_1 are the fixed initial and final times while x and y are the Cartesian coordinates of the hand in the plane of motion. The *minimum-jerk model*, as it was called by the authors, agreed with the measured human arm trajectories during medium speed, large amplitude, unconstrained motions in a horizontal plane. Hogan [57] used the minimum-jerk model to postulate the equilibrium trajectory hypothesis for the formation of human motions. In contrast to Feldman's equilibrium point hypothesis, Hogan suggests that the equilibrium point of the system of springs representing the muscles is gradually shifted along a prescribed trajectory and not simply switched between the initial and the final configuration. The trajectory along which the equilibrium point moves is a minimum-jerk trajectory.

In parallel with the work of Flash and Hogan, Nelson [97] also speculated that motions are planned so that they minimize some global cost functional. He discussed how cost functionals such as time, energy, integral of impulse and integral of jerk can be used to model trajectories for violin bowing and jaw movements.

Following these ideas, Uno *et al.* [143] suggested another cost functional to model generation of human single arm reaching motions. They suggested that the motions minimize the integral of the squared norm of the vector of torque derivatives:

$$J = \frac{1}{2} \int_{t_0}^{t_1} \sum_{i=1}^{n} \dot{\tau}_i^2\, dt. \tag{1.2}$$

As before, t_0 and t_1 are the fixed initial and final time while τ is the vector of the joint torques. The authors presented experiments in which the *minimum torque-change model*,

as it is usually called, fitted the data better than the minimum-jerk model. The model also predicted kinematic features of the unrestrained vertical arm motions for which the minimum-jerk hypothesis failed [2]. In addition, they showed that the minimum torque-change trajectories can be computed with a layered neural network and argued that such mechanism could be implemented in the central nervous system [65].

The minimum-jerk [42] and minimum torque-change [143] models agree quite well with the experimental data; it is still an open question which paradigm best describes the processes that take place in the central nervous system during motion planning. Although both models are based on the idea of minimizing an integral cost functional, the implications of the models for the generation of human motion are quite different. The minimum-jerk model suggests that only the task space trajectory is planned. This trajectory is independent of the physical structure (dynamics) of the system that performs the motion. The equilibrium trajectory paradigm complements the minimum-jerk model and explains how the motion along the planned trajectory is generated. One of the main claims of this model is that the trajectory is planned separately at the kinematic (task) level and it is subsequently transformed into joint trajectories and actuator activations according to the equilibrium trajectory hypothesis (see also [55]). Unfortunately, the equilibrium trajectory hypothesis is very difficult to verify since the dynamic parameters of the human arm (damping, stiffness) during motion are difficult to measure reliably.

In contrast, the premise of the minimum torque-change model is that an explicit plan is computed in the joint space and the actuator space and that the planning process is performed in a single step thereby making the joint level and actuator level planning dependent. This is because the cost functional depends on the actuator forces and incorporates the dynamics of the human arm into the planning process. The minimum torque-change model provides an explicit trajectory for the actuator forces and is easier to verify with experiments. However, it is computationally more intensive than the minimum-jerk model since the dynamics of the system enters into the planning process and has to be modeled.

1.2 Trajectory generation in robotics

The division into explicit and implicit schemes for motion planning also applies to robotics. In most cases, explicit schemes are used for kinematic motion planning while implicit schemes are usually employed for dynamic motion planning.

An important concept that evolved through the study of kinematic motion planning in robotics is the notion of configuration space. Although well-known in mechanics [48], it was first defined in the robotics literature by Lozano-Pérez and Wesley [81]. The basic idea is to represent the robot by a point in an appropriate space. Each element of this space corresponds to a different configuration of the robot, from which the name *configuration space*. In this space, an obstacle Obs_i maps to a set of configurations $\mathcal{C}_{\mathrm{Obs}_i}$ in which the robot touches or penetrates the obstacle. Finding a collision-free path for a robot thus corresponds to finding a trajectory in the configuration space that does not intersect any of the sets $\mathcal{C}_{\mathrm{Obs}_i}$. Although the computation of sets $\mathcal{C}_{\mathrm{Obs}_i}$ is very time consuming, it is usually only performed once and a path between any arbitrary two configurations can be computed very efficiently afterwards.

1.2.1 Implicit schemes

Implicit schemes only use the information about the current state of the robot and the environment to compute how to move and can be interpreted as feedback mechanisms. They are very attractive from a computational point of view since no processing is required prior to the motion. The simplest scheme, corresponding to the final position control in biological systems, is to make the set-point for the joint controllers equal to the desired final position in the joint space and let the error between the current position and the set-point drive the robot. A modification of this scheme where the velocity during the motion is appropriately shaped is often provided on industrial robots as one of the possible modes of motion, but it is hardly useful for large amplitude motions. One of the reasons is that the shape of the trajectory in the task space depends on the location of the start and the end configuration within the joint space. If obstacles are present in the workspace of a robot, it is difficult to predict whether the robot will avoid them or not.

An approach analogous to Feldman's equilibrium point hypothesis [39] was developed by Khatib [67]. He defines a potential function with the equilibrium point at the goal configuration. The actuators of the robot are programmed to generate the force dictated by the potential field, driving the robot towards the goal configuration. This scheme is much more flexible than the final position control since the potential field that guides the motion can be chosen. It is also easy to implement obstacle avoidance by assigning a repulsive potential to each obstacle (this is how the method was first used). By making the range of repulsive potential limited, only the obstacles that are close to the robot will affect the motion. The robot thus only needs to know local information about the environment. If the potential is defined in the joint space, the problem of kinematic redundancy can be resolved as well [149]. The main drawback of the potential field approach is that there may exist local minima that can "trap" the robot. Koditschek [70] showed that if obstacles are present there is no potential function with a unique equilibrium point. Rimon and Koditschek [117] demonstrated that a potential function can be constructed (they call it *the navigation function*) which has a global minimum and for which all other equilibrium points are saddle-points (unstable equilibria) that lie in a set of measure zero. However, constructing such a navigation function requires complete knowledge of the space topology and many advantages of Khatib's approach are lost. Another deficiency of potential fields is that the generated trajectories are usually far from being of "minimal-length". Finally, it is difficult to take into account various constraints posed by the task such as velocity limits or nonholonomic constraints.

1.2.2 Explicit schemes

To compute a trajectory, explicit methods (we could also call them open-loop schemes) require knowledge of the global properties of the space. The advantage of such schemes is that task requirements can be taken into account during the planning process. The approach is also attractive from the control point of view: once the trajectory of the system is planned, the system can be linearized along this trajectory and methods from linear control theory can be used to control its motion [91, 120, 148].

An excellent overview of methods for kinematic motion planning can be found in Latombe's book [74]. Latombe divides planning algorithms into three classes: roadmap,

cell decomposition and potential field. We discussed the potential field methods above and argued that they are basically implicit (feedback) methods. Roadmap methods [21, 26, 100, 102] construct a set of curves, called roadmap, that "sufficiently connect" the space. A path between two arbitrary points is found by choosing a curve on the roadmap and connecting each of the two points to this curve with a simple arc. Instead, cell decomposition methods [3, 124, 125] divide the configuration space into non-overlapping cells and construct a connectivity graph expressing the neighborhood relations between the cells. The cells are chosen so that a path between any two points in the cell is easily found. To find a trajectory between two points in the configuration space, a "corridor" is first identified by finding a path between the cells containing the two points in the connectivity graph. Subsequently, a path in the configuration space is obtained by appropriately connecting the cells that form the corridor.

Kinematic motion planning methods described in this and in the previous section have been mainly developed for mobile robots or so-called free flying agents. Kinematic constraints typical for a serial manipulator make the analysis more complicated, although in principle the methods can be appropriately adapted. An interesting example of a more general method is the recursive algorithm for generating trajectories for nonholonomic vehicles presented in [75]. The most general versions of roadmap and cell decomposition methods work for cases in which the obstacles in the configuration space can be described as semi-algebraic sets [74]. However, most practical implementations assume that the obstacles and the robot can be described as polygons. At the price of considerably increased complexity, it is also possible to extend some of the approaches to cases in which the obstacles in the environment move and their position is provided by sensors.

A common feature of all the motion planning schemes described in this section is that they are based on discrete algorithms. In one way or another the configuration space is discretized and represented by a graph. Subsequently, trajectory planning is reduced to finding a path in this graph. These methods are also purely kinematic: they only generate a trajectory in the configuration space, while the dynamics of the robot and the possible constraints on the actuator forces are not taken into account. To obtain a trajectory in the actuator space a separate mechanism must be employed. From the point of view of hierarchical organization, they therefore assume separate planning at each of the three levels (task space, joint space, actuator space).

1.2.3 Continuous motion planning

We are interested in solving the motion planning problem without artificially discretizing the configuration space. Such motion planning methods will be called *continuous*. A good starting point is provided by studies of human arm motions (Section 1.1). Both, the minimum-jerk model (Equation 1.1) and the minimum torque-change model (Equation 1.2) are continuous models for motion planning. From the mathematical point of view, both schemes formulate motion planning as a variational problem. However, while the minimum-jerk cost functional only depends on the task variables, the minimum torque-change cost functional depends on the actuator torques. In turn, the dynamic equations represent additional constraints and finding a solution for the minimum torque-change model requires solving an optimal control problem [119]. We already noted that the minimum torque-change model is not a hierarchical planning method, it yields a solution at joint and

actuator levels simultaneously.

The problem of minimizing a functional on the trajectories of a dynamical system described with a set of differential equations is traditionally studied in optimal control. When the constraining equations are algebraic, it is customary to use variational calculus. It therefore seems that optimal control corresponds to dynamic motion planning while variational calculus is a kinematic motion planning method. We will show that such this distinction is artificial and that variational calculus and optimal control are two different names for the same set of techniques. For this reason, we will use the term variational methods to denote both, optimal control and traditional variational calculus methods.

The example of minimum-jerk and minimum torque-change models shows that variational methods can be used either for kinematic or for dynamic motion planning; the type of planning depends on the cost functional. Variational methods have been used in robotics in both roles long before similar ideas were used for modeling human arm motions. After all, the idea to move the robot along straight line paths in the task space, which is probably as old as the first robots (see [15] for a review of early work), is just a special case of variational approach where the cost functional is the Euclidean distance. Most of the work in robotics that uses a variational approach is based on optimal control and addresses the dynamic motion planning problem.

Optimal control has been extensively used in robotics for time-optimal control. In time-optimal control, the objective is to minimize the duration of motion between the initial and goal configurations so that constraints on the actuator forces are satisfied. The problem was first studied by Kahn and Roth [63]. The authors were able to show that for a three-link serial mechanism with constant limits on the torques at least one of the actuators operates at the limit during time-optimal motion. They also proposed an approximation scheme based on linearization of the robot dynamics to compute the optimal solution.

Time-optimal control is also an alternative for planning trajectories in the actuator space when the path in the joint space is obtained with other methods (see above). Bobrow et al. [13] and Shin and McKay [129] independently developed similar methods for computing the time-optimal control for a serial manipulator moving along a given path. They showed that this problem is equivalent to time-optimal control of a double integrator system and that the solution will thus be bang-bang (the input is always equal to one of the limits and it instantaneously switches between the limits finitely many times). Based on the definition of *a velocity limit curve* in the two-dimensional state space they developed an algorithm to compute the switching points. Pfeiffer and Johanni [109] and Slotine and Yang [132] further simplified this algorithm. Huang and McClamroch [59] adapted the method to contour following and found a solution for switching between free motion and constrained motion. Shiller and Lu [126] generalized the algorithm from [13] to handle singular trajectories. Dahl [32] extended the approaches from [13] and [129] to flexible manipulators.

Theoretical results for time-optimal control of mechanical linkages were derived in [1, 30, 135]. Shiller and Dubowsky [128] proposed an algorithm for computing time-optimal inputs and trajectories for a robot moving among obstacles. They approximated the trajectories with B-splines and used the algorithm from [13] in conjunction with graph-search techniques to find the time-optimal trajectory.

Time-optimal solutions require discontinuous jumps in the actuator forces. But in

practice, motors have their own dynamics and such discontinuous jumps can not be produced. Although minimum time solutions are attractive from a theoretical perspective, time-optimal control has limited practical applications. For this reason, some researchers tried to combine time-optimal control with approaches that yield smooth actuator trajectories. One possibility is to minimize the combination of time and the integral of the square norm of the vector of inputs. The last integral is usually taken as a measure of the energy consumed during the motion and the resulting problem is thus known as time-energy optimal control. Singh and Leu [131] used dynamic programming to solve time-energy optimal control when the robot path is specified. Bessonnet and Lallemand [8] computed time-energy optimal trajectories and inputs using Pontryagin's minimum principle. A similar problem was also solved by Shiller [127].

In applications where time and limits on the actuator forces are not critical, other cost functionals can be used to obtain smooth motion plans. Vukobratović and Kirćanski [146] proposed minimization of energy (approximated by the integral of the square norm of the vector of inputs) to compute optimal inputs for a 6 degree-of-freedom anthropomorphic manipulator. They assumed a known path and computed velocity distribution along the path using dynamic programming. Nakamura and Hanafusa [96] proposed optimal control to resolve kinematic redundancy. They studied several cost functionals (square norm of the velocity, square norm of the force vector, manipulability) and used Pontryagin's minimum principle in conjunction with the shooting method to find the joint trajectories for a 7 degree-of-freedom robotic arm from the task space trajectory. Similar is the work of Suh and Hollerbach [138] who studied the minimization of energy as a method for resolving kinematic redundancy. They also compared global (variational) methods to local optimization schemes for resolving actuator redundancy and showed that the former are usually superior.

An application of variational calculus to kinematic motion planning is the work of Buckley [23]. He proposed variational calculus as an alternative to discrete algorithms for the trajectory planning in the configuration space. The main focus of his work was the derivation of a distance function used to test the feasibility of the trajectories. Based on this definition he derived an algorithm for trajectory planning that favorably compared with discrete trajectory planning methods.

Variational calculus was applied to motion planning for nonholonomic systems in Fernandes *et al.* [40]. They studied motion of a falling cat, where the non-integrable (nonholonomic) constraint is the preservation of the angular momentum. The motion planning problem was posed as the minimization of the L^2 norm of the inputs and the Ritz method was used to compute the solution.

A variational method is also proposed in [134] for open-loop trajectory planning for nonlinear systems with no drift. The boundary-value problem that has to be solved to find a feasible trajectory is converted into a variational problem where the error at the boundaries is minimized. This technique has been extended in [33] for planning trajectories for nonholonomic systems moving amidst obstacles.

A rich body of literature exists on hierarchical approaches to motion planning. In this work, the emphasis is on the resolution of kinematic or actuator redundancy along a given task space trajectory. The general idea is to prescribe a scalar cost functional and find its minimum at each point along the task space trajectory. An example of such approach is the use of a pseudo-inverse in velocity control of redundant manipulators [151].

A similar approach for resolving actuator redundancy is proposed in [88]. The major difference between these methods and the variational approaches is that minimization in the former case is performed locally and in finite dimensions, while for the latter the search is performed in functional (infinite-dimensional) space and globally optimal solutions can be obtained. We therefore speak about *local* and *global* methods. The advantage of local methods is that the solution can be computed on-line during the motion, while for global methods the motion plan must be computed in advance. But since local methods assume a preceding stage in which the task space trajectory is planned off-line, their advantage over global methods seems questionable. A discussion of some further shortcomings of local optimization schemes can be found in [84] and [138].

At the end of this review we mention a class of continuous motion planning methods of a rather different character that are an alternative to variational methods. These techniques, usually known as *steering*, grew from research in nonlinear control [18, 139] and can also be viewed as constructive proofs of controllability. They have been very successfully applied to nonholonomic systems [77]. Murray and Sastry [93] showed that a large class of nonholonomic systems can be steered to a desired configuration using sinusoids. A more general theory was developed in [72] and [140] for drift-free systems. An extension of the work of Murray and Sastry can be found in [142]. An important link between trajectory planning and control for nonholonomic systems is provided by Walsh *et al.* [148], who showed how a planned trajectory can be stably tracked with a nonholonomic robot. Steering techniques were mainly developed for kinematic motion planning. Recent advances in modeling of systems with symmetry and modeling of nonholonomic mechanical systems [12, 103], make some of these ideas applicable to dynamic motion planning.

An overview of the different approaches to motion planning that appear in the literature is shown in Figure 1.2. Specific methods are in the leaves of the tree: the methods that were developed for robot motion planning are underlined while the rest of the methods can be found in studies of human motion. The frame around the word "variational" denotes that the focus in the dissertation will be on variational methods.

1.3 Approach in the dissertation

In most of the literature, motion planning is defined as finding a suitable path between two points in space. In this work, we take a much broader view so that the motion plan includes every aspect of the motion that is necessary to perform the task. This broader definition in a way combines what is traditionally understood as motion planning with task planning that is typically performed at a more abstract level. For example, if the task is to grasp an object with a multi-fingered hand, we are interested in how to form a grasp, move the object, control the force so that the object is firmly held but not crushed and to regrasp the object when necessary. Similarly, when planning motion for a multi-legged robot, we are concerned with the gait pattern, placement of the legs on the ground and timing of events during walking.

The methods developed in this dissertation fall into the class of variational or optimal control approaches. As indicated throughout the chapter, variational methods have several advantages:

- They can be used as dynamic motion planning methods. They can therefore take

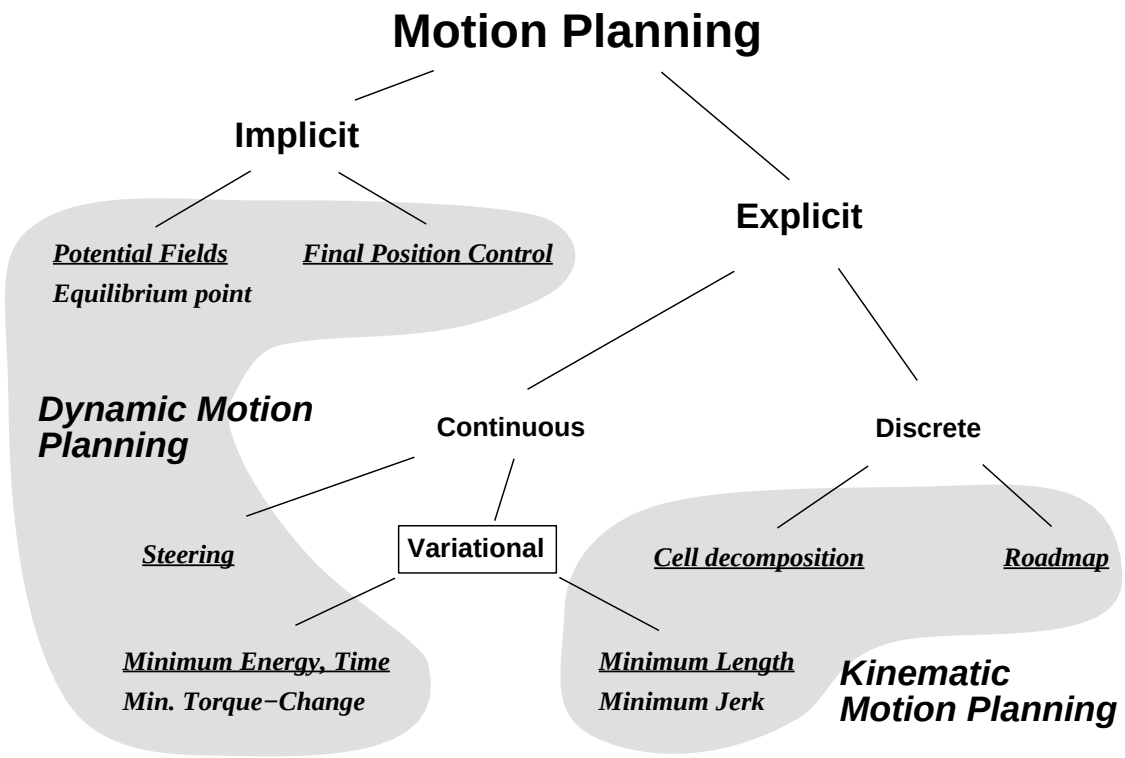


Figure 1.2: Division of the approaches to motion planning found in the literature.

into account the dynamics of the system and provide the trajectories at all three levels: task space, joint space and actuator space. Most of the other methods only provide kinematic trajectories.

- By minimizing a cost functional we find a trajectory and resolve kinematic and actuator redundancies within the same framework. In other approaches, different strategies are employed at different levels.

- Variational methods provide a unifying framework for dealing with equality and inequality constraints. In particular, since the dynamics of the system can be taken into account, nonholonomic constraints become simply equality constraints on the state space and can easily be treated.

- Variational methods can yield a feasible and globally optimal solution. In contrast, kinematic motion planning is not concerned with the limitations on the actuator capabilities and might produce solutions that are not suitable for implementation. This is also a disadvantage of local optimization schemes.

By choosing variational methods for motion planning we make some important assumptions. The critical assumption is that we have a good model of the system. In this work we also assume that the model is completely deterministic. These assumptions limit the generality of the proposed approach, but they are satisfied in many practical applications. By choosing an explicit scheme for motion planning we also assume that we can synthesize

a controller that can follow the computed trajectories. Such controllers can be readily synthesized if the system is controllable.

We will show that variational calculus and optimal control are equivalent and can be used for kinematic or dynamic motion planning, depending on the chosen cost functional. In the first part of this dissertation we will develop a general method for planning trajectories and actuator forces. Due to physical limitations of a robot and because of the nature of interactions of a robot with the environment, a robotic task typically involves equality and inequality constraints. The problem that we will address in the first part can be therefore described as follows:

Problem 1.1 (Dynamic motion planning) *Let the robot be described in the state space with the equations:*

$$\dot{x} = f(x, u, t) \tag{1.3}$$

where x is the vector of state variables and u is the vector of inputs. Suppose that during the motion, the robot must satisfy a set of equality and inequality constraints:

$$g_i(x, u, t) = 0 \quad i = 1, \ldots, k \tag{1.4}$$

and

$$h_i(x, u, t) \leq 0 \quad i = 1, \ldots, l, \tag{1.5}$$

and let the initial and final configurations be given by

$$\alpha(x, t)|_{t_0} = 0 \tag{1.6}$$

and

$$\beta(x, t)|_{t_1} = 0. \tag{1.7}$$

Choose a cost functional:

$$J = \Psi(x(t_1), t_1) + \int_{t_0}^{t_1} L\,(x(t), u(t), t)\; dt. \tag{1.8}$$

The functions f, g, h and L are assumed to be of class C^2 in all variables, while the functions α, β and Ψ are assumed to be of class C^1. ***The motion planning problem is to find (a piecewise smooth) state vector $x^*(t)$ and (a piecewise continuous, bounded) input vector $u^*(t)$ that satisfy equations (1.3)-(1.7) and minimize the cost functional (1.8).***

We are interested in motion planning for mechanical systems. Continuity requirements on the functions f, g and h are therefore not problematic since the dynamic equations for mechanical systems are sufficiently well behaved. Further, we get to choose the function L, so it can be made sufficiently smooth. However, if the task involves impacts, the impact forces will be impulsive and the state will be discontinuous. In this work we do not consider such cases, but the method for motion planning for systems with changing dynamic behavior that we describe in Chapter 6 provides some insight in how such problems can be addressed.

Kinematic motion planning is simpler and more efficient than dynamic motion planning. It is thus preferred to dynamic motion planning if there exists a suitable way to compute the actuator forces from the kinematic trajectory. Kinematic motion planning is also the

only feasible alternative when a dynamic model of the system is too complicated or is not available and is abstracted with a kinematic model.

Kinematic motion planning can be performed either in the task space or in the joint space. The task space has often more structure and the planning method must be carefully developed to take advantage of this structure. For this reason, we concentrate in the second part of the dissertation on variational methods for kinematic motion planning in the task space. The developed approach can be directly applied to the kinematic planning in the joint space. To develop a general method, we will pose kinematic motion planning as a variational problem on a Riemannian manifold:

Problem 1.2 (Trajectory planning) *Let the task space be described with a manifold Σ and take a Riemannian metric $< .,. >$ on Σ. Choose the desired initial and final configurations p_0 and p_1 for the robot in the task space ($p_0, p_1 \in \Sigma$) and let C_0 and C_1 be the additional conditions that have to be satisfied at these two points. Take the family $\mathcal{G}$ of all smooth curves $\gamma : [0,1] \to \Sigma$ that start at p_0, end at p_1, and satisfy C_0 and C_1: $\mathcal{G} = \{\gamma \mid \gamma(0) = p_0, \gamma(1) = p_1, C_0(\gamma), C_1(\gamma)\}$. Define a function $L : \mathcal{G} \to T\Sigma$ and a functional $J : \mathcal{G} \to \mathbb{R}$ of the form:*

$$J(\gamma) = \int_0^1 < L(\gamma, \frac{d\gamma}{dt}), L(\gamma, \frac{d\gamma}{dt}) > dt. \tag{1.9}$$

The trajectory planning problem on Σ is to find a curve $\gamma \in \mathcal{G}$ that minimizes the functional (1.9).

1.3.1 Contents of the dissertation

The dissertation will be roughly divided into two parts. Chapters 2 to 6 will address finding a solution to Problem 1.1 and present some applications of dynamic motion planning. In Chapter 2, we will briefly review the basic theory of optimal control and state the Pontryagin's minimum principle. We will show that a general optimal control problem with constraints can be formulated as a problem of Bolza in the calculus of variations if the constraints can be described with a set of equality and inequality constraints. We will formulate the necessary conditions for the solution of the variational problem and compare these conditions with those provided by the minimum principle for the optimal control problem to establish the exact mapping between the two problems. We will then show that the problem of Bolza can be transformed into an unconstrained problem in the calculus of variations that has the same set of extremals.

We will briefly review numerical methods for solving optimal control problems in Chapter 3. We will discuss a first-order method for solving Problem 1.1 that exploits the equivalence of extremals of the problem of Bolza and the corresponding unconstrained variational problem. A detailed analysis of the method for problems with equality and inequality constraints that only depend on the state variables will be presented. Since this method can fail to converge to a minimum, we will develop a direct minimization method by approximating the continuous problem with a sequence of discrete nonlinear programming problems. Methods for solving the resulting nonlinear programming problems will also be discussed. At the end of the chapter, we will address how an appropriate cost functional for motion planning can be chosen. We will explain why some cost functionals are only suitable for kinematic motion planning and discuss functionals suitable for planning smooth motions.

In Chapter 4 we will demonstrate the methods from Chapter 3 on the problem of planning trajectories and actuator forces for two planar arms cooperatively manipulating an object. We will start with a simple case when no constraints on the contact forces are present. A more complete treatment will follow in which frictional constraints will be taken into account.

Chapter 5 will provide additional motivation for using variational methods for motion planning. We will show that the method proposed for planning the motion for robotic systems can be applied to modeling human arm motions. Experimental investigation of human two-arm manipulation will be presented and a model based on the results from Chapter 4 proposed for modeling the measured data. This chapter also shows that the interaction between motion planning for biological and robotic systems goes in both directions: principles discovered for biological systems can be used for motion planning in robotics, while methods devised for generating motion plans in robotics are possible models for biological systems.

Chapter 6 represents a link between our framework for motion planning and algorithmic approaches to robot task planning. The method from Chapter 3 will be extended to generate motion plans for a robotic system whose dynamic equations change as it evolves in the state space. A framework for describing such systems in the state space will be developed and we will pose the optimal control problem. We discuss the complexity of this problem and present a partial solution that can be applied in several robotic applications. The method will be demonstrated on an example of a two-fingered hand manipulating a circular object.

Chapters 7 and 8 form the second part of the dissertation and will establish a general framework for kinematic motion planning in the task space. Some background material from Riemannian geometry will be briefly reviewed in Chapter 7. Since in most cases the task space manifold is the Euclidean group of rigid body motions, $SE(3)$, we will devote the rest of the chapter to the study of the geometric structure of this group. It will be shown that different applications in robotics demand different Riemannian metrics on $SE(3)$. In particular, we will look at the metrics suitable for the study of screw motions and discuss connections that are appropriate for kinematic analysis.

The analysis from Chapter 7 will be used in Chapter 8 to study trajectory planning on $SE(3)$. We will discuss the invariance of trajectories with respect to the choice of coordinate frames and introduce a framework for variational calculus on Riemannian manifolds. We will show how cost functionals such as minimum-jerk, that are used in studies of human arm motions, can be extended to an arbitrary manifold. We will also show that in some special cases the expressions for the trajectories minimizing these cost functionals are especially simple. At the end of the chapter we will show some numerically computed trajectories to illustrate their dependence on the choice of the metric, reference frames and boundary conditions.

Chapter 9 concludes the dissertation. There, we will summarize our results and point to the contributions of this work. We will discuss possible applications and identify directions for future research.

Chapter 2

Relation between optimal control and calculus of variations

The central problem of optimal control is to drive an (under-determined) dynamical system from an initial state to the desired final state so that a chosen cost functional is minimized. This problem has been already studied by Lagrange in the 18th century. Important contributions towards the solution of the problem were made by Mayer and Bolza at the turn of the century and later by Bliss and his students. However, optimal control is usually regarded as an outgrowth from the control theory as developed after the second world war. The two most important formal works that established the foundations for the subsequent research in optimal control were Bellman's principle of optimality of dynamic programming [5] and the minimum principle developed by Pontryagin and collaborators [113]. Applications in aviation and aerospace have fostered rich activity in the area in the following decade. Although it soon became apparent that almost all practical problems are intractable analytically, advances in computer technology and numerical methods have made numerical solutions to many optimal control problems feasible and there has been a steady interest in the practical applications of optimal control.

The formulation by Bolza is often more convenient for describing motion planning problems than optimal control. However, most results in the control literature are derived from the minimum principle and it is therefore not clear how they relate to the more classical results from the calculus of variations. In this chapter we establish some of these relations with the aim of using the Bolza formulation to develop a numerical method for solving the motion planning problem 1.1. We first state the Pontryagin's minimum principle for the general case. We show that the problem can be reformulated as the problem of Bolza in the calculus of variations if the set of admissible controls can be described by equality and inequality constraints. We state the first-order necessary conditions for a solution of the variational problem and demonstrate that these necessary conditions are a subset of the conditions given by the minimum principle. In the process, we also find relations between variables in the optimal control formulation and the variational formulation. Finally, we show that for a variational problem with equality and inequality constraints it is possible to formulate an unconstrained variational problem that has the same extremals (stationary points).

2.1 Optimal control problem and Pontryagin's minimum principle

In this section we define the basic setting for study of optimal control. We consider a dynamical system described by an underdetermined system of first order differential equations:

$$\dot{x} = f(x, u, t). \tag{2.1}$$

where the n-dimensional vector x is the state of the system and the m-dimensional vector u represents the controls (inputs) of the system. The system (2.1) is underdetermined since the controls u can be arbitrarily chosen. Once u is chosen, the system becomes determined. The state space will be assumed to be the Euclidean space $\mathbb{R}^n$ and the vector of controls u to be a function $u : \mathbb{R} \to U \subset \mathbb{R}^m$, where the set U can be arbitrary[1]. We define the *set of admissible controls*, $\mathcal{U}$ to consist of those functions u that are bounded and piecewise continuous. We also assume that the function $f : \mathbb{R}^n \times U \times \mathbb{R} \to \mathbb{R}^n$ is C^1 for all its arguments. Given a vector u, the existence and uniqueness of the solution to (2.1) is therefore guaranteed for arbitrary initial state x_0 [56].

A transition of the dynamical system from one state to another is called *a motion*. Let the initial and the desired final state for the dynamical system (2.1) be given by:

$$x(t_0) = x_0 \qquad \text{and} \qquad x(t_1) = x_1, \tag{2.2}$$

and define the cost of a motion to be the value of the functional:

$$J = \Psi\left(x(t_1), t_1\right) + \int_{t_0}^{t_1} L(x, u, t)dt. \tag{2.3}$$

Here t_0 and t_1 are the initial and final time, respectively. The function L is assumed to be of the same class as the function f. The optimal control problem is:

Problem 2.1 (Optimal control) *Consider the dynamical system described by Equation (2.1) and a functional J of the form (2.3).* ***The optimal control problem is to find among all the admissible controls $u \in \mathcal{U}$ the control which moves the system (2.1) from the initial state x_0 to the desired final state x_1 and minimizes the functional (2.3).***

In the rest of this work, we will assume that given a certain topology[2] on the set of piecewise smooth functions, there exists an open set of functions $u(t)$ that can bring the system (2.1) from the initial state x_0 to the final state x_1. If this were not the case, it would not make sense to talk about the minimization of the functional J. Problems where this assumption is satisfied are also called *normal* [119].

Without further assumptions, we can not say whether the optimal solution exists and whether it is unique. However, Pontryagin *et al.* [113] showed that if the optimal solution exists it must satisfy the necessary conditions known as the *Pontryagin's minimum principle*:

[1] See [139] for a more general setting of the problem.

[2] The choice is usually between so called strong and weak topologies [119].

Theorem 2.2 (Pontryagin's minimum principle) *Define the Hamiltonian:*

$$H(x, u, \lambda, t) = L(x, u, t) + \lambda^T f(x, u, t). \tag{2.4}$$

If a control $u^*(t)$ is optimal and it generates a trajectory $x^*(t)$, then there exists a nonzero solution $\lambda^*(t)$ of the (vector) ***adjoint equation****:*

$$\dot{\lambda} = -(\frac{\partial H}{\partial x})^T \tag{2.5}$$

such that for every $t \in [t_0, t_1]$ and for every admissible $u \in U$:

$$H(x^*, u^*, \lambda^*, t) \leq H(x^*, u, \lambda^*, t). \tag{2.6}$$

Pontryagin's minimum principle therefore identifies the set of inputs that could minimize the cost functional (2.3). The vector $\lambda(t)$ has dimension n and its components are called *the adjoint variables*. Note also that the system equations (2.1) can be rewritten as:

$$\dot{x} = \frac{\partial H}{\partial \lambda}. \tag{2.7}$$

The minimum principle is very general and places no restrictions on the set U [119]. However, Equation (2.6) is difficult to convert to an efficient numerical procedure. A more convenient expression can be obtained if the set U is open. In this case, Equation (2.6) implies that the optimal input u^* satisfies:

$$\frac{\partial H(x, u, \lambda, t)}{\partial u} = 0. \tag{2.8}$$

Remark 2.3 The optimality condition (2.8) is just the first-order necessary condition for (2.6) and is therefore weaker.

Minima of the cost functional (2.3) belong to a broader class of curves that render this cost functional stationary. Such curves are called *extremals* and play the central role in the theory of optimal control and the calculus of variations. When the set U is open, the stationary points of the functional (2.3) are exactly the curves that satisfy Equation (2.8). In this case, the extremals are therefore given by the following set of differential equations:

$$\begin{aligned} \dot{x} &= \frac{\partial H(x, u, \lambda, t)}{\partial \lambda} \\ \dot{\lambda} &= -\left(\frac{\partial H(x, u, \lambda, t)}{\partial x}\right)^T, \end{aligned} \tag{2.9}$$

where the input u is calculated from (2.8). If the initial state x_0 and the final state x_1 are given, they provide all the necessary boundary conditions for the system (2.9). If the value $x^k(0)$ is not specified, the missing condition must be replaced by (see [22]):

$$\lambda^k(0) = 0. \tag{2.10}$$

Similarly, if a component $x^k(1)$ is missing, the boundary condition becomes:

$$\lambda^k(1) = \left.\frac{\partial \Psi}{\partial x^k}\right|_{t_1}. \tag{2.11}$$

Equations (2.10)-(2.11) are also called *the natural boundary conditions* and are a special case of so called *transversality conditions*.

In general, it is hard to obtain results about the existence and uniqueness of the solutions of the boundary value problem (2.9) and this is the main reason that such results are also difficult to state for optimal control problems. Another question is how realistic is the assumption that the set U is open. We will show that when the boundaries of the set U can be expressed analytically, the machinery of Lagrange multipliers can be used to convert the problem with constrained set U into one where U is the whole space and therefore open. Since this is the case in most applications, the minimum principle has found widespread use.

Suppose therefore that the set of admissible controls consists of all bounded, piecewise continuous functions $u : \mathbb{R} \to \mathbb{R}^m$ that satisfy a set of equality and inequality constraints:

$$g_i(x, u, t) = 0 \quad i = 1, \ldots, k \tag{2.12}$$

and

$$h_i(x, u, t) \leq 0 \quad i = 1, \ldots, l. \tag{2.13}$$

For the problem to be meaningful, we require that $k \leq m$. The minimum principle in this case can be restated in the unconstrained form [22]:

Proposition 2.4 (Unconstrained formulation) *Define the Hamiltonian:*

$$H = L + \lambda^T f + \mu^T g + \nu^T h. \tag{2.14}$$

where the vectors μ and ν have dimensions k and l, respectively. If the optimal control problem has a solution then there exists a nonzero solution $\lambda^*(t)$ of the adjoint equation:

$$\dot{\lambda} = -(\frac{\partial H}{\partial x})^T \tag{2.15}$$

and the optimal input $u^*(t)$ satisfies the equation:

$$\frac{\partial H(x, u, \lambda, \mu, \nu, t)}{\partial u} = \frac{\partial L}{\partial u} + \lambda^T \frac{\partial f}{\partial u} + \mu^T \frac{\partial g}{\partial u} + \nu^T \frac{\partial h}{\partial u} = 0. \tag{2.16}$$

The optimal solution $x^*(t)$ and the optimal input $u^*(t)$ must satisfy Equations (2.12) and (2.13), and, in addition, the following ***complementarity conditions*** *must hold:*

$$\nu_i \begin{cases} = 0 & \text{if } h_i < 0 \\ \geq 0 & \text{if } h_i = 0. \end{cases} \tag{2.17}$$

We note that when the set of admissible controls is given by the Equations (2.12)-(2.13), the optimal control problem 2.1 is exactly the same as the motion planning problem 1.1.

2.2 Variational reformulation of the optimal control problem

An alternative approach to solve Problem 2.1 is to observe that it can be converted into a problem in the calculus of variations known as *the problem of Bolza* [6]. The problem of Bolza was defined at the beginning of the century by Bolza and is the closest in formulation to the optimal control problem. However, it can be shown to be equivalent to problems of Lagrange and Mayer which were defined much earlier [11]. These problems were extensively studied by Bliss and his students at the University of Chicago in the '30s and '40s and the results are described in [11]. Formally, the problem of Bolza is defined as:

Problem 2.5 (Bolza) ***Find a piecewise smooth (r-dimensional vector) function $X(t)$ that minimizes the functional:***

$$J = \hat{\Psi}\left(X(t_1), t_1\right) + \int_{t_0}^{t_1} \hat{L}(X, u, t)dt \tag{2.18}$$

subject to the side constraints:

$$\varphi_i(X, \dot{X}, t) = 0 \quad i = 1, \ldots, p \quad (p \leq r), \tag{2.19}$$

the initial condition

$$X(t_0) = X_0, \tag{2.20}$$

and the final condition

$$X(t_1) = X_1. \tag{2.21}$$

We again assume that the problem is normal, that is, that there exists an open set (in an appropriate topology) of functions $X(t)$ that solve the boundary value problem (2.19)-(2.21). If this were not the case, this boundary value problem would only have isolated solutions and it would not make sense to talk about a variation of the functional J.

To establish the relation between the optimal control problem and the problem of Bolza we first prove the following:

Lemma 2.6 ***If the set of admissible controls for the optimal control problem 2.1 can be described with a set of equality and inequality constraints, the optimal control problem can be formulated as a problem of Bolza 2.5.***

Proof: Conversion of Problem 1.1 into a problem of Bolza involves 4 steps:

1. Introduce a vector of *slack variables* ξ of dimension l and define:

$$\Xi_i = h_i(x, u, t) + \xi_i^2 \quad i = 1, \ldots, l, \tag{2.22}$$

where the functions h_i are those from (2.13). It is easy to see that the inequality constraints (2.13) are equivalent to the equality constraints:

$$\Xi_i(x, u, t, \xi) = 0 \quad i = 1, \ldots, l. \tag{2.23}$$

This conversion of inequality constraints into equality constraints was proposed by Valentine [144]. The slack variables can be interpreted as additional inputs to the system.

2. Let $r = n + m + l$ and let

$$\begin{aligned} x_i &= X_i \quad i = 1, \ldots, n \\ u_i &= \dot{X}_{n+i} \quad i = 1, \ldots, m \\ \xi_i &= \dot{X}_{n+m+i} \quad i = 1, \ldots, l. \end{aligned} \tag{2.24}$$

Note that the inputs and the slack variables are integrated when converted to the variables of the problem of Bolza. The unknowns for the variational problem are therefore smoother.

3. Define

$$\begin{aligned} \varphi_i &= f_i - \dot{X}_i \quad i = 1, \ldots, n \\ \varphi_{n+i} &= g_i \quad i = 1, \ldots, k \\ \varphi_{n+k+i} &= \Xi_i \quad i = 1, \ldots, l, \end{aligned} \tag{2.25}$$

where the functions f_i and g_i are those from Equations (2.1) and (2.12), while the functions Ξ_i were defined in (2.22). For the problem to be meaningful we assume $k \leq m$.

4. The last step consists of renaming the parameters of the functions Ψ and L in the functional (2.3) to reflect definitions (2.24):

$$\begin{aligned} \hat{\Psi}(X, \dot{X}, t) &= \Psi(x, t) \\ \hat{L}(X, \dot{X}, t) &= L(x, u, t). \end{aligned} \tag{2.26}$$

□

The above reduction was first proposed by Hestenes [52] and was later used in [6] and [49] to study optimal control problems through the theory of the calculus of variations.

Remark 2.7 The reduction of the optimal control problem to a well understood problem in the calculus of variations (that was known before the minimum principle was published [52]) suggests that optimal control could be studied within the classical theory of the calculus of variations. When Pontryagin and his coworkers derived the minimum principle, the classical theory was derived for *(piecewise) smooth* functions defined over some *open set* in the appropriate space. Instead, the minimum principle was derived for admissible controls being *measurable* and for *arbitrary choice* of the range space U of controls [113]. At the time, the theory of optimal control was therefore more advanced than the classical theory of the calculus of variations (at least as far as necessary conditions are concerned).

The necessary conditions for the solution of Problem 2.5 can be obtained using classical results from the calculus of variations [11]. A study of the first variation of the functional (2.18) yields the following (first-order) necessary condition:

Proposition 2.8 *Define an admissible variation, $Z(t)$, to be a piecewise smooth function such that $Z(t_0) = Z(t_1) = 0$. If a piecewise smooth function $X(t)$ is a solution of Problem*

2.5, then there exists a nonzero vector of Lagrange multipliers $\Lambda = (\Lambda_1, \ldots, \Lambda_{n+k+l})$ *and a function* $\hat{H}$ *defined by:*

$$\hat{H} = \hat{L} + \sum_{i=1}^{n+k+l} \Lambda_i \varphi_i \tag{2.27}$$

such that:

$$\int_{t_0}^{t_1} \left[\frac{\partial \hat{H}}{\partial X}^T Z + \frac{\partial \hat{H}}{\partial \dot{X}}^T \dot{Z} \right] dt = 0, \tag{2.28}$$

for any admissible variation Z*. In addition, the side constraints*

$$\varphi = 0 \tag{2.29}$$

must hold.

With some further manipulation, we can eliminate the need for explicitly considering admissible variations in the above theorem. The necessary conditions for the solution of the problem of Bolza thus become:

Theorem 2.9 (Euler-Lagrange equations) *If a piecewise smooth function* $X(t)$ *is a solution of Problem 2.5 then there exists a nonzero vector of Lagrange multipliers* $\Lambda = (\Lambda_1, \ldots, \Lambda_{n+k+l})$ *and a function* $\hat{H}$ *defined by:*

$$\hat{H} = \hat{L} + \sum_{i=1}^{n+k+l} \Lambda_i \varphi_i \tag{2.30}$$

such that at every point where the vector $X(t)$ *is smooth, the following* ***Euler-Lagrange*** *equations hold:*

$$\frac{d}{dt}\frac{\partial \hat{H}}{\partial \dot{X}} - \frac{\partial \hat{H}}{\partial X} = 0. \tag{2.31}$$

At a point t *where* $\dot{X}(t)$ *has jump discontinuity, the so called* ***Weierstrass-Erdmann corner conditions*** *must hold:*

$$\begin{aligned} \left.\frac{\partial \hat{H}}{\partial \dot{X}}\right|_{t^-} &= \left.\frac{\partial \hat{H}}{\partial \dot{X}}\right|_{t^+} \\ \left.\left(\dot{X}\frac{\partial \hat{H}}{\partial \dot{X}} - \hat{H}\right)\right|_{t^-} &= \left.\left(\dot{X}\frac{\partial \hat{H}}{\partial \dot{X}} - \hat{H}\right)\right|_{t^+}. \end{aligned} \tag{2.32}$$

In addition, the $n + k + l$ *side constraints*

$$\varphi = 0 \tag{2.33}$$

must hold everywhere.

Since the first order necessary conditions are exactly the conditions that the functional (2.18) is made stationary, the functions that satisfy the Euler-Lagrange equations are precisely the extremals.

2.3 Equivalence of the extremals

In this section we find the relation between the extremals of the optimal control problem and the extremals of the corresponding variational problem. The main result is:

Proposition 2.10 *Suppose that the set U in the statement of Problem 2.1 can be described with a set of equalities and inequalities. Then the following statements hold:*

(a) If a piecewise smooth function $X(t)$ satisfies the first order necessary conditions for the optimal control problem stated in Proposition 2.4, it also satisfies the first order necessary conditions of the equivalent problem of Bolza given by Theorem 2.9.

(b) If $X(t)$ satisfies the first order necessary conditions for the Bolza problem given in Theorem 2.9 and if the Lagrange multipliers corresponding to the inequality constraints are everywhere non-negative, then $X(t)$ also satisfies the first order necessary conditions in Proposition 2.4 for the optimal control problem.

Proof: The first-order necessary conditions for the optimal control problem with equality and inequality constraints were stated in Proposition 2.4. Following the proposition, we define:

$$H = L + \lambda^T f + \mu^T g + \nu^T h. \tag{2.34}$$

The equations for the extremals are therefore the system equations:

$$\dot{x} = f, \tag{2.35}$$

the adjoint equations:

$$\dot{\lambda}^T = -\left(\frac{\partial L}{\partial x} + \lambda^T \frac{\partial f}{\partial x} + \mu^T \frac{\partial g}{\partial x} + \nu^T \frac{\partial h}{\partial x}\right), \tag{2.36}$$

the equality constraints:

$$g = 0, \tag{2.37}$$

the optimality conditions:

$$\frac{\partial L}{\partial u} + \lambda^T \frac{\partial f}{\partial u} + \mu^T \frac{\partial g}{\partial u} + \nu^T \frac{\partial h}{\partial u} = 0 \tag{2.38}$$

and the complementarity conditions:

$$\nu_i \begin{cases} = 0 & \text{if } h_i < 0 \\ \geq 0 & \text{if } h_i = 0. \end{cases} \tag{2.39}$$

Note that the complementarity conditions can be rewritten as:

$$\nu^T h = 0, \quad \nu \geq 0. \tag{2.40}$$

We now derive the equations for the extremals of the equivalent problem of Bolza. Following the four steps from the proof of Lemma 2.6, we define $X = [X_1^T X_2^T X_3^T]^T$, where:

$$\begin{aligned} X_{1_i} &= x_i \qquad && i = 1, \ldots, n \\ \dot{X}_{2_i} &= u_i \qquad && i = 1, \ldots, m \\ \dot{X}_{3_i} &= \xi_i \qquad && i = 1, \ldots, l. \end{aligned} \tag{2.41}$$

We partition the vector of Lagrange multipliers from Theorem 2.9 into $\Lambda = [\Lambda_1^T \Lambda_2^T \Lambda_3^T]^T$, where the dimension of Λ_1 is n, the dimension of Λ_2 is k and the dimension of Λ_3 is l. Analogously, we define the vector of side constraints $\varphi = [\varphi_1^T \varphi_2^T \varphi_3^T]^T$ by:

$$\begin{aligned} \varphi_{1_i} &= f_i - \dot{X}_{1_i} = 0 \qquad && i = 1, \ldots, n \\ \varphi_{2_i} &= g_i = 0 && i = 1, \ldots, k \\ \varphi_{3_i} &= h_i + \xi_i^2 = 0 && i = 1, \ldots, l. \end{aligned} \tag{2.42}$$

Following Theorem 2.9, we define:

$$\hat{H} = \hat{L} + \Lambda_1^T \varphi_1 + \Lambda_2^T \varphi_2 + \Lambda_3^T \varphi_3. \tag{2.43}$$

Note that the function $\hat{H}$ does not depend on X_2 and X_3 but only on their time derivatives.

Since the side constraints must hold, we first obtain:

$$\begin{aligned} \varphi_1 &= f - \dot{X}_1 = 0 \\ \varphi_2 &= g = 0, \end{aligned} \tag{2.44}$$

which are Equations (2.35) and (2.37). Further, according to the definition (2.42) the equation:

$$\frac{d}{dt}\frac{\partial \hat{H}}{\partial \dot{X}_1} - \frac{\partial \hat{H}}{\partial X_1} = 0, \tag{2.45}$$

leads to:

$$-\frac{d}{dt}\Lambda_1^T - \left(\frac{\partial \hat{L}}{\partial X_1} + \Lambda_1^T \frac{\partial \varphi_1}{\partial X_1} + \Lambda_2^T \frac{\partial \varphi_2}{\partial X_1} + \Lambda_3^T \frac{\partial \varphi_3}{\partial X_1} \right) = 0. \tag{2.46}$$

It is not difficult to see that this is exactly Equation (2.36), if we put $\Lambda_1 = \lambda$, $\Lambda_2 = \mu$ and $\Lambda_3 = \nu$. Next, take the equation:

$$\frac{d}{dt}\frac{\partial \hat{H}}{\partial \dot{X}_2} - \frac{\partial \hat{H}}{\partial X_2} = 0. \tag{2.47}$$

Since $\hat{H}$ does not depend on X_2, the equation becomes:

$$\frac{d}{dt}\frac{\partial \hat{H}}{\partial \dot{X}_2} = 0, \tag{2.48}$$

and after integration:

$$\frac{\partial \hat{H}}{\partial \dot{X}_2} = \text{const}. \tag{2.49}$$

There are no boundary conditions specified for X_2. We are only interested in $u = \dot{X}_2$, which means that an arbitrary constant can be added to X_2 without changing the value of u, so we can set:

$$X_2|_{t_0} = 0. \tag{2.50}$$

On the other boundary, we must enforce the so called *natural boundary condition* [119]:

$$\left. \frac{\partial \hat{H}}{\partial \dot{X}_2} \right|_{t_1} = 0. \tag{2.51}$$

From Proposition 2.9, it can also be seen that the function $\frac{\partial H}{\partial \dot{X}}$ is continuous along the extremal. We therefore conclude:

$$\frac{\partial \hat{H}}{\partial \dot{X}_2} = 0, \tag{2.52}$$

which becomes:

$$\frac{\partial \hat{L}}{\partial \dot{X}_2} + \Lambda_1^T \frac{\partial \varphi_1}{\partial \dot{X}_2} + \Lambda_2^T \frac{\partial \varphi_2}{\partial \dot{X}_2} + \Lambda_3^T \frac{\partial \varphi_3}{\partial \dot{X}_2} = 0. \tag{2.53}$$

This equation is the same as the optimality condition (2.38). Finally, we consider the equation:

$$\frac{d}{dt}\frac{\partial \hat{H}}{\partial \dot{X}_3} - \frac{\partial \hat{H}}{\partial X_3} = 0. \tag{2.54}$$

Following similar reasoning as above, this equation yields:

$$\frac{\partial \hat{H}}{\partial \dot{X}_3} = 0. \tag{2.55}$$

Using the definitions for $\hat{H}$ and φ_3, the equation becomes:

$$2\Lambda_{3_i}\xi_i = 0 \quad i = 1, \ldots, l. \tag{2.56}$$

This equation can be rewritten as:

$$\Lambda_{3_i}\xi_i^2 = 0 \quad i = 1, \ldots, l. \tag{2.57}$$

and finally by expressing ξ from (2.42):

$$\Lambda_{3_i} h_i = 0 \quad i = 1, \ldots, l. \tag{2.58}$$

Since $\Lambda_{3_i} = \nu_i$, the last equation will be satisfied if the complementarity condition (2.40) is satisfied, which proves (a). Further, if $\Lambda_3 \geq 0$, this equation is exactly the complementarity condition (2.40) which also proves (b). □

Remark 2.11 We could prove a stronger claim that the minimizing extremals of both problems are the same by using second-order necessary conditions. In this case, the condition $\Lambda_3 \geq 0$ would follow from the Legendre condition $H_{\dot{X}\dot{X}} \geq 0$.

Corollary 2.12 *If the optimal control problem is reformulated as the problem of Bolza, the following relations hold between the Lagrange multipliers of the two problems:*

$$\begin{aligned} \lambda &= \Lambda_1 \\ \mu &= \Lambda_2 \\ \nu &= \Lambda_3. \end{aligned} \tag{2.59}$$

This relations will be important when we discuss the case in which the functions h and g do not depend explicitly on the inputs u.

The next observation is that the Lagrange multipliers Λ in the variational formulation can be replaced with new functions Θ, where:

$$\Lambda = \dot{\Theta}. \tag{2.60}$$

With this substitution, it is not difficult to see from Equation (2.43) that:

$$\varphi = \frac{\partial \hat{H}}{\partial \dot{\Theta}}. \tag{2.61}$$

Further, if a natural boundary conditions are enforced on $\dot{\Theta}$ at t_1, the side constraint is equivalent to the equation:

$$\frac{d}{dt}\frac{\partial \hat{H}}{\partial \dot{\Theta}} = 0. \tag{2.62}$$

Since the function $\hat{H}$ does not depend on Θ, the last equation has the form of the Euler-Lagrange equation for the variable Θ. This leads to:

Proposition 2.13 *The extremal $X(t)$ and the set of Lagrange multipliers $\Lambda(t)$ for the problem of minimizing the cost function:*

$$J = \hat{\Psi}\left(X(t_1), t_1\right) + \int_{t_0}^{t_1} \hat{L}(X, \dot{X}, t)dt \tag{2.63}$$

subject to the side constraints

$$\varphi(X, \dot{X}, t) = 0, \tag{2.64}$$

correspond to the extremal $Y(t) = \left(X^T(t), \Theta^T(t)\right)^T$ of the problem of minimizing the cost function:

$$\tilde{J} = \hat{\Psi}\left(X(t_1), t_1\right) + \int_{t_0}^{t_1} \left(\hat{L}(X, \dot{X}, t) + \dot{\Theta}^T \varphi(X, \dot{X}, t)\right) dt \tag{2.65}$$

where $\Lambda = \dot{\Theta}$.

This proposition shows that to find the extremals (not necessarily minimizing), the variational calculus problem with side constraints can be converted into an equivalent problem without side constraints if the Lagrange multipliers Θ_i are treated in the same way as the unknown functions. In other words:

Formulation 1
Critical point X of:

$$J = \hat{\Psi}\left(X(t_1), t_1\right) + \int_{t_0}^{t_1} \hat{L}(X, \dot{X})dt$$

subject to

$$\varphi(X, \dot{X}) = 0.$$

$\Longleftrightarrow$

Formulation 2
Critical point $Y = \left(X^T, \Theta^T\right)^T$ of:

$$\tilde{J} = \hat{\Psi}\left(X(t_1), t_1\right) + \int_{t_0}^{t_1} H(Y, \dot{Y})dt$$

where

$$H(Y, \dot{Y}) = \hat{L}(X, \dot{X}) + \dot{\Theta}^T \varphi(X, \dot{X}).$$

It is important to note that the first order necessary conditions do not distinguish between maxima, minima and inflection points. This implies that it would be wrong to conclude on the basis of Proposition 2.13 that a critical point that is a minimum for Formulation 1 must also be a minimum for Formulation 2. To check if an extremal is a minimum, we must employ second order necessary conditions. The second order necessary condition (also known as the Legendre-Clebsch necessary condition) for X to minimize the cost functional J in Formulation 1 is that the matrix $G_{\dot{X}\dot{X}}$, where

$$G = \hat{L} + \lambda^T \varphi,$$

is positive definite [119]. For Formulation 2, the necessary condition that (X, Θ) minimizes the cost functional $\tilde{J}$ is that the matrix $H_{\dot{Y}\dot{Y}}$ is positive definite [53], where Y is the vector:

$$Y = \begin{bmatrix} X \\ \Theta \end{bmatrix}.$$

Now assume that the only constraints are the system equations. The function φ is thus given by:

$$\varphi_i(X, \dot{X}) = f_i(X, \dot{X}) \qquad\qquad i = 1, \ldots, n.$$

In this case, the Legendre-Clebsch necessary condition for Formulation 1 becomes $H_{uu} \geq 0$. We also have:

$$Y = \begin{bmatrix} x \\ Y_u \\ Y_\lambda \end{bmatrix},$$

where

$$\begin{aligned} u &= \dot{Y}_u \\ \lambda &= \dot{Y}_\lambda. \end{aligned}$$

A short calculation shows that the matrix $H_{\dot{Y}\dot{Y}}$ has the following form:

$$H_{\dot{Y}\dot{Y}} = \begin{bmatrix} 0_{n\times n} & 0_{n\times m} & -I_{n\times n} \\ 0_{m\times n} & \frac{\partial^2 H}{\partial u^2} & 0_{m\times n} \\ -I_{n\times n} & 0_{n\times m} & 0_{n\times n} \end{bmatrix}. \tag{2.66}$$

This matrix is indefinite regardless of the value of the matrix H_{uu} (which depends on x, u and λ). This implies that for Formulation 2 no critical point is a minimum for $\tilde{J}$! Lagrange multipliers can therefore not be used to convert a constrained **minimization** problem to an unconstrained **minimization** problem.

Remark 2.14 If the variational problem is convex, the cost functional in the Formulation 2 must be actually **maximized** over the Lagrange multipliers Θ [7].

2.3.1 Regular and singular extremals

Take the Bolza problem 2.5 with r unknown functions (the dimension of the vector $X(t)$) and s side constraints (the dimension of the vector φ). An extremal $X(t)$ for the problem of Bolza is called *regular*, if:

$$\det \left. \frac{\partial(\hat{H}_{\dot{X}_1}, \ldots, \hat{H}_{\dot{X}_r}, \varphi_1, \ldots, \varphi_s)}{\partial(\dot{X}_1, \ldots, \dot{X}_r, \Lambda_1, \ldots, \Lambda_s)} \right|_{\substack{X = X(t) \\ \Lambda = \Lambda(t)}} \neq 0 \quad \text{for all } t \in [t_0, t_1], \tag{2.67}$$

where Λ is the vector of Lagrange multipliers corresponding to the extremal $X(t)$. An extremal which is not regular is called *singular* [119]. An analogous definition exists for the optimal control problems [53].

If an extremal is regular, it is possible to reduce the Euler-Lagrange equations and the side constraints to a system of $2\,r$ first order differential equations (see [119, pp. 336]). In other words, the field of extremals (all the solutions of the Euler-Lagrange differential equation with no boundary conditions imposed) can be embedded into a $2\,r$ parameter family of curves. This is a good indication that we can satisfy $2\,r$ boundary conditions (but it is not sufficient). On the other hand, even if the extremal is singular, we might be able to satisfy the boundary conditions in some special cases! A singular extremal can be compared to an over-constrained system of linear equations: In general, such a system does not admit a solution, but if the rank of the system matrix is the same as the rank of the extended system matrix, a solution exists.

Numerical methods often produce oscillatory solutions along singular extremals. This can be explained by the fact that the first non-zero term in the Taylor series expansion of the cost functional along a singular arc is of the fourth order (at a minimum, the first non-zero term in the Taylor series must be of the even order, otherwise the variation would change sign). The influence of the solution on the value of the cost functional is therefore very "weak" and as a result the numerical methods are ill-conditioned. The common practice in such cases is to add a term to the cost functional that makes the problem non-singular and then gradually decrease this term to zero.

Chapter 3

Numerical method

In this chapter, we present two methods for finding extremals of the motion planning problem 1.1 numerically. We start with a short overview of numerical methods for optimal control. We next present a numerical method for computing the extremals of the problem of Bolza that was proposed by Gregory and Lin [49]. The method finds a solution that satisfies the (first order) necessary conditions for the extremal, but can not distinguish between a local minimum or a local maximum. We point to some shortcomings of the method if it is applied to problems with equality and inequality constraints and propose appropriate modifications. Next, we describe a novel method for direct minimization of a cost functional. The continuous problem is discretized by using ideas from finite-element analysis; techniques from nonlinear programming are used to solve the resulting finite-dimensional optimization problem. We review the method of augmented Lagrangian for solving constraint problems and Newton's method for minimization. We show that the expressions for the gradients that are used in minimization can be easily computed if the proposed scheme is used to discretize the continuous problem. Finally, we discuss the choice of cost functionals that can be used to formulate the motion planning problem and demonstrate that the cost functional determines whether a motion planning or a trajectory planning problem is solved.

3.1 Numerical methods for solving optimal control problems

A number of numerical methods have been developed over the years for solving optimal control problems. Most methods were developed for specific applications and there is no consensus on a good general method. The main classes of methods are:

- Indirect methods: The extremals are computed from the minimum principle (first-order necessary conditions) by solving a two-point boundary value problem. The most popular technique for solving boundary value problems is to guess the missing initial values and improve the guess until the boundary values are matched. If the initial guess is close to the exact solution, this method can be shown to converge. For obvious reason, the technique is also called *the method of neighboring extremals* [22]. The main drawback of the method is that it is very sensitive to the initial guess.

- Direct methods: The cost functional (2.3) is minimized with respect to u directly by computing the gradient $\frac{\partial J}{\partial u}$. In analogy with the minimization of functions in finite dimensions, first order and second order gradient methods can be formulated, as well as conjugate gradient algorithms. In the earlier methods [60, 73], the gradient of (2.3) with respect to u was computed by a procedure known as backward sweep and the methods were conceptually very similar to the method of neighboring extremals. More recent algorithms [83, 86, 89] approximate the gradient directions of the original problem by approximating it with a simpler optimal control problem. These methods are robust and globally convergent. They usually deal with constraints by adjoining them to the cost functional in the form of *penalty functions*.

- Discretization methods: The continuous problem is discretized and transformed into a finite-dimensional problem. In this way, the integral in (2.3) becomes a finite sum and methods of finite-dimensional mathematical programming can be used to find a solution. Some methods only discretize the vector of controls [141], while the others discretize both, the state and controls [116]. Another possibility is to interpolate the unknown functions with a set of basis functions (collocation methods) [99]. As with the direct methods, the constraints are often adjoined to the cost functional with the penalty functions.

- Dynamic programming: The idea is to discretize the state space and compute the optimal value of the cost functional for each grid point taken as the initial state for the optimization. This value is called *the optimal return function* [22] and is computed by a recursive algorithm [5]. Given the values of the optimal return function, it is easy to construct an extremal starting at an arbitrary initial point. Dynamic programming always produces a global minimum, but the computation becomes prohibitively expensive as the dimension of the state space increases.

- An indirect method alternative to the neighboring extremals method is to solve the two point boundary value problem resulting from the first-order necessary conditions with a finite difference methods [154]. By its nature, it is a combination of the indirect methods and discretization methods: the unknowns and the first-order necessary conditions are discretized, so that a two-point boundary value problem is converted into a system of algebraic equations.

In deciding which numerical method to use for our work, we were guided by a number of criteria. Any realistic motion planning problem contains inequality constraints. The method should therefore be able to easily handle such constraints, preferably without any extra computations. Further, the method must be applicable to a wide variety of dynamical systems that appear in robotic applications (redundant and over-actuated manipulators, multiple cooperating manipulators, walking machines, etc.). It is also desirable that equality constraints (apart from the dynamic equations) can be handled without elimination of variables. This is especially important when the system contains closed kinematic loops resulting in kinematic closure equations that can not be solved analytically. Similarly, it is advantageous if the dynamic equations can be implicit. Implicit equations are typically obtained after elimination of Lagrange multipliers, which are used to derive dynamic equations for systems with constraints. Since most of these requirements can be met if the

motion planning problem is formulated as the problem of Bolza, we concentrate on the methods that are appropriate for solving problems in this form.

3.2 First-order method for finding the extremals

The aim of first-order methods is to find an extremal of the variational problem – that is, a curve that satisfies the first-order optimality conditions. Since an extremal is not necessary a minimum for the variational problem, these methods are usually used when we can get a good initial guess for a minimizing extremal or in combination with a direct method after this has approached the minimum.

Traditionally, the extremals are obtained by solving the Euler-Lagrange equations (2.31). However, the Euler-Lagrange equations only hold at points where the extremal, $X(t)$, is smooth. At points where $\dot{X}(t)$ has jumps (these points are called *corners*), the Weierstrass-Erdmann corner conditions (2.32) must be met. The location of the corners, their number and the amplitudes of the jumps in $\dot{X}$ are not known in advance so it is difficult to obtain a numerical method for a general problem using Euler-Lagrange equations.

An alternative way of computing the extremals was suggested by Gregory and Lin [49]. They based their numerical method on the necessary conditions in the integral form, Theorem 2.8, which says that if $X(t)$ is an extremal, the first variation of the functional (2.18) must vanish for every piecewise smooth admissible variation $Z(t)$:

$$\int_{t_0}^{t_1} \left[\left(\frac{\partial \hat{H}}{\partial X} \right)^T Z + \left(\frac{\partial \hat{H}}{\partial \dot{X}} \right)^T \dot{Z} \right] dt = 0, \tag{3.1}$$

Since the Euler-Lagrange equations and the Weierstrass-Erdmann corner conditions are derived from Equation (3.1), no information is lost by proceeding this way.

In Proposition 2.13 we showed that for any problem with side constraints we can construct a problem with no side constraints that has the same extremals. In deriving numerical method we therefore assume that we have to find an extremal for the variational problem:

$$J = \hat{\Psi}\left(X(t_1), t_1\right) + \int_{t_0}^{t_1} \hat{L}(X, u, t) dt, \tag{3.2}$$

and there are no side constraints. Let the number of the unknown functions (dimension of X) be M. To compute the solution numerically, we discretize the interval $[t_0, t_1]$ with a sequence of points $t_0 = a_0 < a_1 < \cdots < a_N = t_1$. For simplicity, we assume that $a_i - a_{i-1} = h$ for $i = 1, \ldots, N$. Next, for $k = 1, \ldots, N-1$ we introduce a set of piecewise linear shape functions

$$\alpha_k(t) = \begin{cases} \frac{t - a_{k-1}}{h} & \text{if } \; a_{k-1} < t \leq a_k, \\ \frac{a_{k+1} - t}{h} & \text{if } \; a_k < t \leq a_{k+1}, \\ 0 & \text{otherwise.} \end{cases} \tag{3.3}$$

A graph of the shape function $\alpha_k(t)$ is shown in Figure 3.1.

Since Equation (3.1) must hold for any admissible variation $Z(t)$, it must also hold if the i^{th} component of $Z(t)$ equals α_k and all the other components equal 0. In this way, we obtain a total of $M\,(N-1)$ equations of the form:

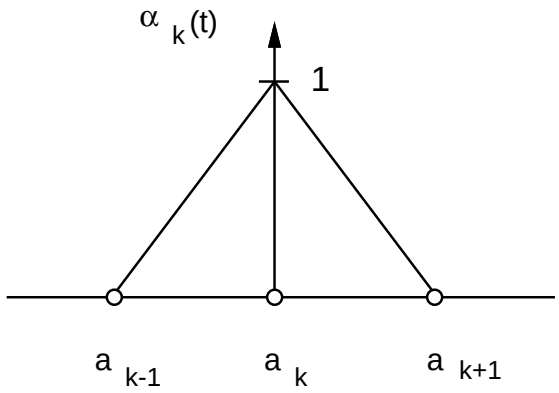


Figure 3.1: Shape functions used in the numerical method.

$$\int_{a_{k-1}}^{a_{k+1}} \left[\frac{\partial \hat{H}_i}{\partial X} \alpha_k + \frac{\partial \hat{H}_i}{\partial \dot{X}} \dot{\alpha}_k \right] dt = 0. \tag{3.4}$$

The integral can be split into two integrals:

$$\int_{a_{k-1}}^{a_k} \left[\frac{\partial \hat{H}_i}{\partial X} \alpha_k + \frac{\partial \hat{H}_i}{\partial \dot{X}} \dot{\alpha}_k \right] dt + \int_{a_k}^{a_{k+1}} \left[\frac{\partial \hat{H}_i}{\partial X} \alpha_k + \frac{\partial \hat{H}_i}{\partial \dot{X}} \dot{\alpha}_k \right] dt = 0. \tag{3.5}$$

These expressions can be approximated using only the values of the unknown function $X(t)$ at the mesh points. We approximate the integrals using the mean-value theorem [43] and replace the derivatives $\dot{X}$ with finite difference approximations:

$$\dot{X} \left(\frac{a_{k-1} + a_k}{2} \right) = \frac{X(a_k) - X(a_{k-1})}{a_k - a_{k-1}}. \tag{3.6}$$

If the equations for all the components of X at point a_k are assembled into a vector, we obtain the following (vector) algebraic equation:

$$\begin{aligned} 0 = {} & \frac{h}{2} \hat{H}_X \left(\frac{a_k + a_{k-1}}{2}, \frac{X_k + X_{k-1}}{2}, \frac{X_k - X_{k-1}}{h} \right) + \\ & \hat{H}_{\dot{X}} \left(\frac{a_k + a_{k-1}}{2}, \frac{X_k + X_{k-1}}{2}, \frac{X_k - X_{k-1}}{h} \right) + \\ & \frac{h}{2} \hat{H}_X \left(\frac{a_{k+1} + a_k}{2}, \frac{X_{k+1} + X_k}{2}, \frac{X_{k+1} - X_k}{h} \right) - \\ & \hat{H}_{\dot{X}} \left(\frac{a_{k+1} + a_k}{2}, \frac{X_{k+1} + X_k}{2}, \frac{X_{k+1} - X_k}{h} \right), \end{aligned} \tag{3.7}$$

where $\hat{H}_X$ and $\hat{H}_{\dot{X}}$ stand for $\frac{\partial \hat{H}}{\partial X}$ and $\frac{\partial \hat{H}}{\partial \dot{X}}$, respectively. But to use the mean value theorem, functions $\hat{H}_X$ and $\hat{H}_{\dot{X}}$ must be continuous. While this is true for $\hat{H}_{\dot{X}}$, function $\hat{H}_X$ could exhibit a jump discontinuity at the corners (points where $\dot{X}$ is discontinuous). However, we can still argue that the approximation in (3.7) is sufficient: There are only finitely many corner points and if the mesh is dense enough, we can show that the error we make in the approximation is small.

If the values $X(a_k)$ are the unknowns, the above procedure yields $M\,(N-1)$ equations to compute them. For the variational problem, we also have $2M$ boundary conditions, either given as part of the problem or obtained from the transversality conditions [119]. At least formally, we thus have enough equations to compute all the unknowns. The following (plausible) argument shows why these equations should also be sufficient to compute all the unknowns: The variation $Z(t)$ is piecewise smooth and each of its components can be approximated with a sum $\hat{Z}_i = \sum_{j=0}^{N} z_{ij}\alpha_j(t)$. As N increases, this approximation can be arbitrarily accurate. Equation (3.1) is linear in every component of $Z(t)$, so any equation that can be obtained from this equation through the above discretization will be just a linear combination of the equations of the form (3.7). This indicates that if the original integral equation completely specifies the extremal, the algebraic equations (3.7) specify its values at the chosen discrete points.

The final outcome of the above process will be a set of $M\,N$ nonlinear algebraic equations. We chose the Newton-Raphson method [115] to solve this system of equations, but any suitable method could be used. An important property of the system of equations (3.7) is that any equation only depends on the values of the unknown variables at three consecutive mesh points. The linearized system that has to be solved with the Newton-Raphson method thus has a block-tridiagonal structure and the computational cost of finding the solution is significantly reduced (see Section 4.3).

3.2.1 State dependent equality constraints

Consider an optimal control problem with equality constraints. Without loss of generality, we assume that inequality constraints are not present and that none of the equality constraints explicitly depends on the input. We also assume that the equality constraints are independent. In other words:

$$\operatorname{rank}\frac{\partial(g_1,\ldots,g_k)}{\partial(x_1,\ldots,x_n)} = k, \tag{3.8}$$

but

$$\frac{\partial(g_1,\ldots,g_k)}{\partial(u_1,\ldots,u_m)} = 0. \tag{3.9}$$

As in the proof of Lemma 2.6, the optimal control problem is converted into the equivalent problem of Bolza by defining:

$$\begin{array}{ll} x = X_1 & \varphi_1 = f - \dot{X}_1 \\ u = \dot{X}_2 & \varphi_2 = g. \end{array} \tag{3.10}$$

The Hamiltonian is defined by:

$$\hat{H} = \hat{L} + \Lambda_1^T\varphi_1 + \Lambda_2^T\varphi_2. \tag{3.11}$$

The matrix from Equation (2.67) can be therefore partitioned in the following way:

$$R = \frac{\partial(\hat{H}_{\dot{X}_1}, \hat{H}_{\dot{X}_2}, \varphi_1, \varphi_2)}{\partial(\dot{X}_1, \dot{X}_2, \Lambda_1, \Lambda_2)} = \begin{bmatrix} \frac{\partial(\hat{H}_{\dot{X}_1}, \hat{H}_{\dot{X}_2}, \varphi_1)}{\partial(\dot{X}_1, \dot{X}_2, \Lambda_1, \Lambda_2)} \\ \frac{\partial\varphi_2}{\partial(\dot{X}_1, \dot{X}_2, \Lambda_1, \Lambda_2)} \end{bmatrix}, \tag{3.12}$$

with the obvious understanding of the notation $\frac{\partial(F,G)}{\partial(X,Y)}$ when F, G, X and Y are vectors. Clearly, the lower part of the matrix R is identically 0, which means that the problem is singular.

We conclude that equality constraints that only depend on the state can not be adjoined to the Hamiltonian $\hat{H}$ directly. One way to overcome this problem is to replace the constraint:

$$g_i(x,t) = 0, \tag{3.13}$$

with the equivalent constraint:

$$\dot{g}_i(x,\dot{x},t) = \sum_{j=1}^{n} \frac{\partial g_i}{\partial x_j}\dot{x}_j = 0 \quad \text{and} \quad g_i(x,t)|_{t_0} = 0. \tag{3.14}$$

If the matrix R in (3.12) is still singular, we replace $\dot{x}$ with the system function $f(x,u,t)$ (Equation 2.1) and repeat the same procedure until we arrive at an equation that explicitly depends on the input. For the motion planning problems considered in this work, the procedure always led to a regular problem.

Another complication that arises with the equality constraints that do not explicitly depend on the inputs concerns the number of boundary conditions that can be specified for the problem. If k constraints do not depend on the inputs, these constraints completely determine k states as a function of the remaining $n-k$ states, assuming that (3.8) holds. Consequently, we only have $n-k$ independent variables and we can specify at most $2\,(n-k)$ boundary conditions. On the other hand, each time a constraint is differentiated, we obtain an additional boundary condition. Clearly, this new boundary conditions must be consistent with the existing ones, otherwise the problem will not have a solution.

3.2.2 State dependent inequality constraints

In Section 3.3.6 we mentioned that extremals for the problems with inequality constraints typically contain corner points. At these points the extremals are not smooth and the Lagrange multipliers can be discontinuous. According to Corollary 2.12 and Proposition 2.13, the relation between the Lagrange multipliers corresponding to inequality constraints for the optimal control problem and the variational problem is given by:

$$\nu = \dot{\Theta}, \tag{3.15}$$

where ν is the multiplier for the optimal control problem and Θ the equivalent multiplier for the problem of Bolza. If ν has a jump discontinuity, $\dot{\Theta}$ will also have a jump discontinuity, which means that Θ is continuous. Jump discontinuities in the multipliers ν thus get eliminated through integration when the optimal control problem is transformed into the equivalent variational problem.

But if the inequality constraints do not depend on inputs, the problem gets more complicated. Assume that there is a single inequality constraint:

$$h(x,t) \leq 0. \tag{3.16}$$

(It is straight forward to generalize our discussion to more than one constraint.) Jacobson *et al.* [62] showed that when such inequality constraint is directly adjoined to the Hamiltonian

of the optimal control problem with a multiplier ν (Equation 2.14), the integral of ν can have jump discontinuity where the constraint becomes active or ceases to be active. Since the integral of ν is equal to Θ, the later will be discontinuous. This violates the assumption that the solutions of the variational problem are piecewise smooth which was used in derivation of the numerical method in Section 3.2.

It appears that this problem can be avoided using an idea suggested by Jacobson and Lele [61]. We assume that $h \in C^{p+1}$ and that:

$$\frac{\partial \frac{d^p h}{dt^p}}{\partial u} \neq 0 \quad \text{and} \quad \frac{\partial \frac{d^i h}{du^i}}{\partial u} = 0 \ \text{ for } i < p. \tag{3.17}$$

The number p, indicating how many times the constraint has to be differentiated before the inputs explicitly appear, is called *the order* of the inequality constraint [61]. The inequality constraint (3.16) is first transformed into an equality constraint:

$$h(x,t) + \xi^2 = 0, \tag{3.18}$$

with a slack variable ξ. Jacobson and Lele then took the sequence:

$$\begin{aligned} h + \xi^2 &= 0 \\ \frac{dh}{dt} + \frac{d\,\xi^2}{dt} &= 0 \\ &\vdots \\ \frac{d^p h}{dt^p} + \frac{d^p \xi^2}{dt^p} &= 0 \end{aligned} \tag{3.19}$$

and defined new state variables:

$$\begin{aligned} y_1 &= \xi \\ y_2 &= \dot{\xi} \\ &\vdots \\ y_{p-1} &= \frac{d^{p-1}\xi}{dt^{p-1}} \end{aligned} \tag{3.20}$$

and a new input:

$$v = \frac{d^p \xi}{dt^p}. \tag{3.21}$$

The initial values for the state variables $y_1, \ldots, y_{p-1}$ can be computed from the sequence (3.19). We will denote these initial values by $y_1^0, \ldots, y_{p-1}^0$. We can therefore write the state equations:

$$\begin{aligned} \dot{y}_1 &= y_2 & y_1(t_0) &= y_1^0 \\ \dot{y}_2 &= y_3 & y_2(t_0) &= y_2^0 \\ &\vdots & &\vdots \\ \dot{y}_{p-2} &= y_{p-1} & y_{p-2}(t_0) &= y_{p-2}^0 \\ \dot{y}_{p-1} &= v & y_{p-1}(t_0) &= y_{p-1}^0 \end{aligned} \tag{3.22}$$

If the state variables y_i and the input v are substituted into the last equation in (3.19), the equation can be rewritten as:

$$\frac{d^p h}{dt^p} + \frac{d^p \xi^2}{dt^p} = G(x, u, y, v, t) = 0. \tag{3.23}$$

It is not difficult to see that Equation (3.23), together with the state equations and initial conditions (3.22), is equivalent to the original inequality constraint (3.16). Further, the equality constraint (3.23) explicitly depends on the new input v. It therefore appears that we were able to avoid the problem of discontinuities of the unknown functions.

Further analysis of the resulting system shows that the problem is still there. We can verify that y_1 is given by:

$$y_1 = \sqrt{-h}. \tag{3.24}$$

Since $y_2 = \dot{y}_1$, it follows that:

$$y_2 = -\frac{\dot{h}}{2\sqrt{-h}}. \tag{3.25}$$

But on the constrained arc, y_1 is identically 0, which implies that at point t_c where the system switches between the unconstrained arc $h < 0$ and the constrained arc $h = 0$, the values of y_2 are:

$$y_2|_{t_c^-} = \lim_{t \to t_c^-} -\frac{\dot{h}}{2\sqrt{-h}} \qquad y_2|_{t_c^+} = 0 \tag{3.26}$$

Clearly, in general the two values will not be the same and y_2 will have a discontinuity.

The method by Jacobson and Lele can be modified to avoid the discontinuity of y_2. We replace the equation defining y_2 in (3.20) by:

$$y_2 = \frac{d(\xi^2)}{dt} = \frac{d(y_1^2)}{dt}, \tag{3.27}$$

With this definition, the sequence of equations (3.19) becomes:

$$\begin{aligned} h + y_1^2 &= 0 \\ \frac{dh}{dt} + y_2 &= 0 \\ &\vdots \\ \frac{d^p h}{dt^p} + v &= 0, \end{aligned} \tag{3.28}$$

so that:

$$y_2 = -\frac{dh}{dt}. \tag{3.29}$$

Since we assumed that $h \in C^p(\mathbb{R})$, y_2 is continuous. It is also clear from Equation (3.28) that all the other states y_i are continuous. We can proceed in the same way as above to formulate the optimal control problem.

Unfortunately, this derivation hides yet another problem. If we expand Equation (3.27), we obtain:

$$y_2 = 2y_1\dot{y}_1. \tag{3.30}$$

This is clearly a singular differential equation at $y_1 = 0$ hence the optimal control problem derived with the above method is singular. The singularity is not a product of our method and Jacobson and Lele point to it in [61]. Intuitively, the singularity is the result of the nature of the state inequality constraint. At the time the constraint becomes active, all higher derivatives of the slack variable instantaneously become 0. The method of slack variables therefore invariably leads to singular problems. In numerical computations, the singularity results in oscillations of the slack variable ξ at the junction of constrained and unconstrained arcs. Because of the oscillations, the convergence of the Newton-Raphson method becomes quite slow.

3.3 A direct method for solving the Bolza problem

The method from the previous section can deal with a general type of optimization problem and is relatively easy to implement. However, as all indirect methods, it can not distinguish between minima, maxima and inflection points. In other words, even when the method converges, the critical point might not be a local minimum. Furthermore, if we determine that the resulting critical point is not a minimum, the only way to find a different extremal is to change the initial guess. This is clearly not a good way of solving optimization problems. In this section, we develop a direct method for solving the Bolza problem that always converges to a (local) minimum.

As we saw at the end of Section 2.3, unconstrained formulation of the variational problem defined in Proposition 2.13 can not be a basis for an algorithm that directly minimizes a cost functional. We have to deal with the constraints in some other way. There exists a number of techniques in finite-dimensional nonlinear programming for dealing with constraints. Examples include penalty function methods, methods of feasible directions and gradient projection methods [111]. The basic idea is to build a sequence of approximate solutions that converges to the exact solution. Most of the methods rely on the fact that the space over which the function is minimized is closed and bounded. A closed and bounded subset of a finite-dimensional space is compact which implies the existence of an accumulation point for the sequence. These methods can be usually also formulated in an abstract setting and subsequently applied to the infinite-dimensional case. However, in passing to an infinite-dimensional domain, even if the domain is closed and bounded it is not necessarily compact. The existence of an accumulation point and convergence of the method is thus not assured.

We follow an alternative approach to deal with the infinite-dimensional domain: we approximate the infinite-dimensional problem with a sequence of finite-dimensional nonlinear programming problems. Formally, one must show that the sequence of the solutions of the approximate problems converges to the solution of the original problem (sufficient conditions for this to be true are derived in [112]).

We defer the discussion on how to approximate an infinite-dimensional problem with a finite-dimensional nonlinear programming problem until Section 3.3.3 and first concentrate on methods for solving finite-dimensional nonlinear programming problems.

3.3.1 Penalty functions and augmented Lagrangian

Suppose that we have a nonlinear programming problem:

$$\min_x L(x) \qquad \text{subject to} \qquad g(x) = 0, \tag{3.31}$$

where x belongs to a subset of a finite-dimensional vector space. To convert this nonlinear programming problem into an unconstrained problem we use the method of augmented Lagrangian. The method was independently proposed by Hestenes [54] and Powell [114] and combines the features of the penalty function methods and Lagrangian methods [7]. Its main advantage is that it is relatively simple to implement and that it can handle nonlinear constraints. The main idea is to adjoin the constraint with a penalty function to the Lagrangian to obtain *the augmented Lagrangian*:

$$H_c(x, \lambda) = L(x) + \lambda g(x) + \frac{c}{2} g(x)^2. \tag{3.32}$$

It is not difficult to see that:

$$\inf_x \lim_{c \to \infty} H_c(x, \lambda) = \inf_x L(x). \tag{3.33}$$

An iterative scheme is obtained by exchanging inf and lim to compute:

$$\lim_{c \to \infty} \inf_x H_c(x, \lambda). \tag{3.34}$$

It can be shown that under weak assumptions we can indeed exchange the minimization and the limit process [7]. We can then perform a sequence of minimizations:

$$\min_x H_{c_k}(x, \lambda_k), \tag{3.35}$$

by using a non-decreasing sequence of positive penalty parameters $\{c_k\}$. Let x_k be a solution to this minimization problem for the value c_k. If the Lagrange multiplier λ is fixed, the penalty parameter c has to be increased to infinity for the sequence x_k to converge to the solution of (3.31). Further, we can show that the minimization (3.35) becomes increasingly ill-conditioned as c increases. However, if after each step λ_k is augmented by:

$$\lambda_{k+1} = \lambda_k + c_k g(x_k), \tag{3.36}$$

where x_k is the solution of (3.35), we can show that the method converges to the solution of the original constrained problem for a finite value of the penalty parameter c_k. An iteration consisting of a minimization step (3.35) followed by augmentation of the Lagrange multiplier (3.36) thus represents a feasible numerical algorithm and can be shown to have good convergence properties [7].

If we have to satisfy an inequality constraint:

$$h(x) \leq 0, \tag{3.37}$$

we can use exactly the same procedure after converting the inequality constraint to an equality constraint with a slack variable:

$$h(x) \leq 0 \qquad \Leftrightarrow \qquad \hat{h}(x, \zeta) = h + \zeta^2 = 0. \tag{3.38}$$

At each step we then have to minimize:

$$\min_{x,\zeta} H_c(x,\zeta) = \min_{x,\zeta} L(x) + \lambda \hat{h}(x,\zeta) + \frac{c}{2}\hat{h}(x,\zeta)^2. \tag{3.39}$$

But for any given x, the above minimization can be performed explicitly with respect to ζ. By writing $\zeta^2 = z$, we have:

$$\min_{z\geq 0} \lambda(h+z) + \frac{c}{2}(h+z)^2. \tag{3.40}$$

The above function is quadratic in z with positive leading coefficient. Its minimum, z^*, on the positive half-plane is attained at:

$$z^* = \max\{0, -[\frac{\lambda}{c} + h(x)]\}. \tag{3.41}$$

By writing:

$$p(x,\lambda,c) \stackrel{\text{def}}{=} h(x) + z^* = \max\{h(x), -\frac{\lambda}{c}\}, \tag{3.42}$$

the cost function becomes:

$$\hat{H}_c(x) = L(x) + \lambda\, p(x,\lambda,c) + \frac{c}{2}\, p(x,\lambda,c)^2. \tag{3.43}$$

The update rule for the Lagrange multiplier is therefore:

$$\lambda_{k+1} = \lambda_k + c_k\, p(x,\lambda,c) = \max\{0, \lambda_k + c_k h(x_k)\}. \tag{3.44}$$

An advantage of this derivation is that by using $\hat{H}_c(x)$ the slack variable ζ is completely eliminated[1].

In the case of two-sided constraints of the form:

$$\alpha \leq h(x) \leq \beta, \tag{3.45}$$

a similar procedure (see [7, pp. 164-167] for derivation) leads to the following expression for the augmented Lagrangian:

$$\hat{H}_c = L(x) + p(h(x),\lambda,c), \tag{3.46}$$

where the penalty function $p(h(x),\lambda,c)$ is given by:

$$p(h(x),\lambda,c) = \begin{cases} (h(x)-\beta)\,(\lambda + \frac{c}{2}(h(x)-\beta)) & \text{if} \quad \lambda + c(h(x)-\beta) \geq 0, \\ (h(x)-\alpha)\,(\lambda + \frac{c}{2}(h(x)-\alpha)) & \text{if} \quad \lambda + c(h(x)-\alpha) \leq 0, \\ -\frac{\lambda^2}{2c} & \text{otherwise.} \end{cases} \tag{3.47}$$

The corresponding multiplier iteration is given by:

$$\lambda_{k+1} = \begin{cases} \lambda_k + c_k(h(x_k)-\beta) & \text{if} \quad \lambda_k + c_k(h(x_k)-\beta) \geq 0, \\ \lambda_k + c_k(h(x_k)-\alpha) & \text{if} \quad \lambda_k + c_k(h(x_k)-\alpha) \leq 0, \\ 0 & \text{otherwise.} \end{cases} \tag{3.48}$$

[1] The derivation is taken from [7].

3.3.2 Minimization by Newton's method

Each iteration of the method of augmented Lagrangian requires minimization of a scalar function. Consider the following minimization problem:

$$\min_x f(x). \tag{3.49}$$

The general scheme for any iterative minimization method is to find a direction of descent, d_k, and update the current solution by:

$$x_{k+1} = x_k + \alpha_k d_k. \tag{3.50}$$

The most popular methods include gradient method, conjugate-gradient method, Newton's method and quasi-Newton's method. The last three methods can achieve quadratic convergence near the solution. All these methods are globally convergent, that is, they will converge to a local minimum if one exists.

Different methods differ in the way how the descent direction d_k is computed. For the Newton's method, the descent direction is computed by scaling the gradient with the inverse of the Hessian:

$$d_k = -[\nabla^2 f(x_k)]^{-1} \nabla f(x_k). \tag{3.51}$$

The motivation for this choice of d_k is that it minimizes the second-order Taylor expansion of the function $f(x)$ around $x = x_k$. Obviously, the Hessian must be invertible in order to compute d_k. Further, d_k will be a descent direction only if the Hessian is positive-definite. A practical scheme that avoids these problems is [7]:

Modified Newton's method: Choose:

$$d_k = -[\nabla^2 f(x_k)]^{-1} \nabla f(x_k). \tag{3.52}$$

if $[\nabla^2 f(x_k)]^{-1}$ exists and the following tests are satisfied:

$$\begin{aligned} \nabla f(x_k)^T [\nabla^2 f(x_k)]^{-1} \nabla f(x_k) \geq c_1 |\nabla f(x_k)|^{p_1}, \qquad & c_1 > 0, p_1 > 2 \\ c_2 |\nabla f(x_k)| \geq |[\nabla^2 f(x_k)]^{-1} \nabla f(x_k)|^{p_2} \qquad & c_2 > 0, p_2 > 1. \end{aligned} \tag{3.53}$$

Otherwise, set:

$$d_k = -D \nabla f(x_k), \tag{3.54}$$

where D is a positive definite scaling matrix and c_1, c_2, p_1 and p_2 are chosen constants (in [7], the suggested choice is $c_1 = 10^{-5}$, $c_2 = 10^5$, $p_1 = 3$, $p_2 = 2$).

The coefficient α_k in Equation (3.50) can be chosen in a number of ways. One possibility is:

$$\alpha_k = \arg\min_t f(x_k + t d_k). \tag{3.55}$$

Another method which is often used in practice is the Armijo rule [7]:

Armijo rule: Choose scalars $s > 0$, $\beta \in (0, 1)$ and $\sigma \in (0, \frac{1}{2})$ and set $a_k = \beta^i s$, where i is the smallest non-negative integer that satisfies:

$$f(x_k) - f(x_k + \beta^i s d_k) \geq -\sigma \beta^i s \nabla f(x_k)^T d_k. \tag{3.56}$$

It is shown in [7] that the described scheme for choosing the descent direction d_k combined with the Armijo rule for choosing the coefficient α_k results in a globally convergent method.

3.3.3 Discretization

To approximate the infinite-dimensional, continuous optimal control problem with a sequence of finite-dimensional, nonlinear programming problems, we employ collocation techniques complemented by the ideas from the finite-element analysis. Collocation methods treat the state variables and the inputs as unknowns and therefore produce larger systems of equations than iterative integration schemes [123], which eliminate the state variables by substituting the function of the input obtained through the numerical integration of the system equations. However, in collocation, the constraints can be easily expressed and we will demonstrate that it is easy to compute the gradient of the cost functional with respect to the unknowns. In robotic tasks, state dependent constraints are practically always present in the form of kinematic closure equations or as bounds on the range of motion of individual joints. Collocation methods are thus well suited for discretizing motion planning problems.

Take the Bolza form of the motion planning problem[2]:

$$\min_x \int_{t_0}^{t_1} L\,(x, \dot{x})\; dt.$$

subject to

$$\begin{aligned} \varphi(x, \dot{x}) &= 0, \\ g(x) &= 0. \end{aligned} \tag{3.57}$$

If the inequality constraints are present, they can be dealt with using methods from Section 3.3.1.The problem is discretized by assuming that the unknown (vector) function can be approximated with a set of basis functions:

$$x_i(t) \approx \sum_{j=0}^{N} p_i^j \phi_j(t), \tag{3.58}$$

$$\dot{x}_i(t) \approx \sum_{j=0}^{N} p_i^j \dot{\phi}_j(t), \tag{3.59}$$

The approximation of the function is required to be exact on the chosen set of grid points $t_0 = a_0 < a_1 < \ldots < a_{N-1} < a_N = t_1$:

$$x_i(a_j) = \sum_{j=0}^{N} p_i^j \phi_j(a_j). \tag{3.60}$$

For simplicity, we assume that $a_j - a_{j-1} = h$ for all $j = 1, \ldots, N$.

The approximating functions $\phi_j(t)$ form a subset of the basis for the space $C^1[t_0, t_1]$ and are assumed to be piecewise smooth on the interval $[t_0, t_1]$ and smooth on each subinterval (a_{j-1}, a_j). Let the values of the unknown x_i at the grid points be $\{x_i^j\}_{j=0}^N$ and let $\bar{x}$ be the collection of these values. Further, let $\hat{x}$ denote the approximation of the function x

[2]If the time t explicitly occurs it can be taken to be an additional state variable so that the system becomes autonomous.

in Equation (3.60). If the set $\{\phi_j\}$ is properly chosen, there will be a 1-1 correspondence between the coefficients p_i^j and the values x_i^j and therefore between $\hat{x}$ and $\bar{x}$. In this case, the discretization of the unknown functions leads to the following approximation of the optimization problem (3.57):

$$\min_{\bar{x}} \int_{t_0}^{t_1} L\left(\hat{x}, \dot{\hat{x}}, t\right)\, dt.$$

subject to

$$\begin{aligned} \int_{a_{j-1}}^{a_j} \varphi(\hat{x}, \dot{\hat{x}})\, dt = 0, \qquad & j = 1, \ldots, N \\ g(\hat{x}(a_j)) = 0, \qquad & j = 1, \ldots, N. \end{aligned} \tag{3.61}$$

This is now a finite-dimensional problem so we can use the augmented Lagrangian method to deal with the constraints. As a result, we have to solve a sequence of optimization problems:

$$\min_{\bar{x}} H_c(\bar{x}, \lambda, \mu) \tag{3.62}$$

where

$$\begin{aligned} H_c = & \int_{t_0}^{t_1} L\left(\hat{x}, \dot{\hat{x}}\right)\, dt \\ + & \sum_{j=1}^{N} \left\{ \lambda^{jT} \int_{a_{j-1}}^{a_j} \varphi(\hat{x}, \dot{\hat{x}})\, dt + \frac{c}{2} \| \int_{a_{j-1}}^{a_j} \varphi(\hat{x}, \dot{\hat{x}})\, dt \|^2 \right\} \\ + & \sum_{j=1}^{N} \left\{ \mu^{jT} g(\hat{x}(a_j)) + \frac{c}{2} \|g(\hat{x}(a_j))\|^2 \right\}, \end{aligned} \tag{3.63}$$

and $\|.\|$ is the usual Euclidean norm.

Choice of functions ϕ_j: A necessary condition for solutions of the discretized problems to converge to a solution of the continuous problem is that:

$$\lim_{h \to 0} \hat{x} \to x, \tag{3.64}$$

where the convergence is assumed to be in the norm:

$$\|x\|_S = \max_{[t_0,t_1]} \|x(t)\| + \sup_{[t_0,t_1]} \|\dot{x}(t)\|. \tag{3.65}$$

In other words, as the discretization gets finer, functions ϕ_j must approximate x arbitrarily well. There are many possible choices for the appropriate set of functions. As we shall see below, it is advantageous for the computation of the gradient if the shape functions have localized support[3]. For our computations, we choose the triangular shape functions described by Equation (3.3) and shown in Figure 3.1 that were also used in the derivation of the method in Section 3.2. In this way, the function x is approximated by a piecewise linear function, while $\dot{x}$ is approximated by a piecewise constant function (Figure 3.2).

One of the properties of the triangular shape functions is that $\phi_j(a_k) = \delta_j^k$, where δ_j^k is the Kronecker symbol. As a result, the coefficients p_i^j in the approximation (3.59) are exactly the functional values at the grid points, $p_i^j = x_i^j$.

[3]Support of a function is the set $\text{supp} f = \text{Cl}\{x | f(x) \neq 0\}$, where Cl stands for closure.

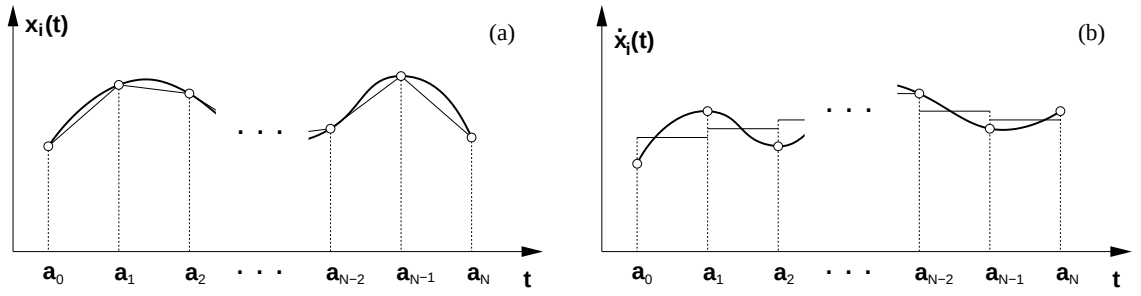


Figure 3.2: Approximation of the state (a) and its derivative (b).

3.3.4 Computation of the gradient ∇H_c

To simplify the computations, we will assume that the support of the approximating functions ϕ_j is a subset of the interval $[a_{j-1}, a_{j+1}]$. This is obviously the case for the triangular shape functions (3.3). We can rewrite H_c as:

$$H_c(\bar{x}) = \sum_{j=1}^{N} \left[I^j(\bar{x}) + F^j(\bar{x})^T \left\{ \lambda^j + \frac{c}{2} F^j(\bar{x}) \right\} + G^j(\bar{x})^T \left\{ \mu^j + \frac{c}{2} G^j(\bar{x}) \right\} \right], \tag{3.66}$$

where we have denoted:

$$\begin{aligned} I^j(\bar{x}) &= \int_{a_{j-1}}^{a_j} L\left(\hat{x}, \dot{\hat{x}}\right) dt, \\ F^j(\bar{x}) &= \int_{a_{j-1}}^{a_j} \varphi(\hat{x}, \dot{\hat{x}})\, dt, \\ G^j(\bar{x}) &= g\left(\hat{x}(a_j)\right). \end{aligned} \tag{3.67}$$

Further, let $\Theta^j(\bar{x})$ denote the function under the summation sign in (3.66):

$$\Theta^j(\bar{x}) = I^j(\bar{x}) + F^j(\bar{x})^T \left\{ \lambda^j + \frac{c}{2} F^j(\bar{x}) \right\} + G^j(\bar{x})^T \left\{ \mu^j + \frac{c}{2} G^j(\bar{x}) \right\}. \tag{3.68}$$

The assumption that the support of the function ϕ_j is a subset of $[a_{j-1}, a_{j+1}]$ implies that Θ^j only depends on the values of $\hat{x}$ at the grid points a_{j-1} and a_j. Therefore:

$$\frac{\partial H_c}{\partial x_i^j} = \frac{\partial \Theta^j}{\partial x_i^j} + \frac{\partial \Theta^{j+1}}{\partial x_i^j}. \tag{3.69}$$

We also have:

$$\frac{\partial \Theta^j}{\partial x_i^j} = \frac{\partial I^j}{\partial x_i^j} + \left\{ \lambda^j + cF^j \right\}^T \frac{\partial F^j}{\partial x_i^j} + \left\{ \mu^j + \frac{c}{2} G^j \right\}^T \frac{\partial G^j}{\partial x_i^j}. \tag{3.70}$$

Expressions $\frac{\partial I^j}{\partial x_i^j}$, $\frac{\partial F^j}{\partial x_i^j}$ and $\frac{\partial G^j}{\partial x_i^j}$ are matrices of partial derivatives. These matrices can be computed directly except for the derivatives of the integrals I^j and F^j. We therefore focus on differentiation of a term of the form:

$$\Theta^j(\bar{x}) = \int_{a_{j-1}}^{a_j} \theta\left(\hat{x}, \dot{\hat{x}}\right) dt. \tag{3.71}$$

Changing the order of differentiation and integration, we have:

$$\frac{\partial \Theta^j}{\partial x_i^j} = \int_{a_{k-1}}^{a_k} \left(\frac{\partial \theta}{\partial x_i} \phi_j + \frac{\partial \theta}{\partial \dot{x}_i} \dot{\phi}_j \right) dt. \tag{3.72}$$

Since θ is in general nonlinear, this integral can not be evaluated analytically. The first-order approximation is obtained by using the midpoint rule for the integration:

$$\frac{\partial \Theta^j}{\partial x_i^j} \approx \frac{h}{2} \frac{\partial \theta}{\partial x_i} \left(\frac{x^{j-1} + x^j}{2}, \frac{x^j - x^{j-1}}{h} \right) + \frac{\partial \theta}{\partial \dot{x}_i} \left(\frac{x^{j-1} + x^j}{2}, \frac{x^j - x^{j-1}}{h} \right). \tag{3.73}$$

The first-order approximation for $\frac{\partial \Theta^{j+1}}{\partial x_i^j}$ is obtained in an analogous way:

$$\frac{\partial \Theta^{j+1}}{\partial x_i^j} \approx \frac{h}{2} \frac{\partial \theta}{\partial x_i} \left(\frac{x^j + x^{j+1}}{2}, \frac{x^{j+1} - x^j}{h} \right) + \frac{\partial \theta}{\partial \dot{x}_i} \left(\frac{x^j + x^{j+1}}{2}, \frac{x^{j+1} - x^j}{h} \right). \tag{3.74}$$

Note the similarity of these expressions to the equation (3.7). Derivation in this section can thus be used as an alternative to the derivation in Section 3.2.

3.3.5 Computation of the Hessian $\nabla^2 H_c$

We have the expressions for the gradient ∇H_c. For the Newton's method, we also need to compute the Hessian $\nabla^2 H_c$. Because of the large number of unknowns, it is not practical to derive the analytical expressions for the entries of the Hessian. We therefore use a simple forward differentiation scheme to compute these entries:

$$\frac{\partial^2 H_c}{\partial x_i^j \partial x_k^l} \approx \frac{\frac{\partial H_c(x + \Delta e_i^j)}{\partial x_k^l} - \frac{\partial H_c(x_i^j)}{\partial x_k^l}}{\Delta}. \tag{3.75}$$

In the equation, e_i^j represents a matrix that has zeroes everywhere except in the component (i, j).

3.3.6 Discussion

The main difficulty in finding solutions of the optimal control problems is due to the presence of corners, that is, points where the derivatives of the unknown function $x(t)$ exhibit jumps. It can be shown that at the corners the Lagrange multipliers can also become discontinuous [119]. Proposition 2.10 established that the minimum principle is closely related to the first-order necessary conditions in the form of Euler-Lagrange equations. Consequently, corners will represent the same problem for the optimal control formulation as for the Bolza formulation.

Corners typically occur for problems with inequality constraints because the dimension of the manifold on which the system evolves changes as the system passes between the unconstrained motion and the constrained motion. If multiple constraints are present, the dimension of the manifold will change each time one of the constraints becomes active or ceases to be active. As the system passes between manifolds of different dimensions, the derivatives of the state typically change discontinuously.

Indirect methods implicitly assume smoothness of the extremals and solve the equations obtained from the first-order necessary conditions. These methods are therefore not directly applicable to problems with inequality constraints. Similarly, direct methods integrate the system equations and the adjoint equations to compute the gradient directions of the cost functional and are therefore sensitive to corners, too. In contrast, discretization methods solve a finite-dimensional problem and problems with corners do not arise.

We argued that the Bolza formulation is convenient for describing motion planning problems. Consequently, we developed a numerical method that can solve the problem in this form. The conversion of the optimal control problem into a variational problem is fairly formal and can be easily automated. In this conversion, integrals of the inputs from the optimal control problem become the unknown functions. These unknowns are thus continuous, enhancing the numerical stability of the method.

The main disadvantage of the presented method is that it requires solving a sequence of nonlinear programming problems. We described one possible technique for solving the nonlinear programming problem to illustrate how a complete numerical algorithm for solving the motion planning problem can be implemented. By choosing a more sophisticated method for dealing with constraints (for example, exact penalty functions) it is possible to solve the nonlinear programming problem in one step. The presented framework for discretization of the continuous problem is thus independent of the method chosen for solving the resulting nonlinear programming problem.

3.4 Choice of the cost functional

Choice of the cost functional depends on the nature of the task for which the motion is planned. Quite often, requirements of the task are contradictory and the cost functional represents a tradeoff between different types of criteria. An example is time-optimal control in robotics: while we wish to minimize the time for the robot motion, we also look for smooth actuator forces so that the stress on the actuators and the mechanism is reduced. These two requirements are contradictory since the time-optimal solution is typically bang-bang. In this example, we have to trade between minimizing the time and maximizing the smoothness of the actuator forces (see [8, 127, 131]).

In the introduction we argued that methods based on minimizing a cost functional are suitable for motion planning because they provide a unified framework for resolving kinematic and actuator redundancies. But we also saw that in certain cases the solution of the variational problem was purely kinematic (for the cost functional (1.1)) while in other instances the solution depended on the dynamic equations (for cost functional (1.2)). We would therefore like to establish what properties the cost functional must possess to yield an unambiguous solution for the actuator forces.

In this dissertation, we are mainly concerned with motion planning for rigid mechanical linkages. The Lagrange equations for such systems can be written in the form:

$$I(q)\ddot{q} + C(q,\dot{q}) + G(q) = B(q)\tau + \Gamma(q)^T\lambda, \tag{3.76}$$

where q is the vector of generalized coordinates of the system, τ is the vector of actuator forces, λ the vector of constraint forces, $I(q)$ the inertia matrix, $C(q,\dot{q})$ the vector of Coriolis and centripetal forces, $G(q)$ the vector of gravity terms, $B(q)$ is the matrix that describes

how the actuators act on the system, and $\Gamma(q)$ is the matrix describing the constraints:

$$\Gamma(q)\dot{q} = 0. \tag{3.77}$$

The inertia matrix $I(q)$ is positive definite, so we can rewrite the dynamic equations for a rigid linkage in the state space form. We define:

$$\begin{array}{ll} x_1 = q & u_1 = \tau \\ x_2 = \dot{q} & u_2 = \lambda, \end{array} \tag{3.78}$$

and rewrite Equation(3.76) as:

$$\dot{x}_1 = x_2 \tag{3.79}$$

$$\dot{x}_2 = I(x_1)^{-1}[-C(x_1, x_2) - G(x_1) + B(x_1)u_1 + \Gamma(x_1)^T u_2]. \tag{3.80}$$

We must also satisfy the constraint equations:

$$\Gamma(x_1)x_2 = 0. \tag{3.81}$$

In Chapter 2 we saw that when the optimal control problem is reformulated as a variational problem, the actuator inputs become additional unknowns that have to be computed as part of the optimization. We mentioned in Section 2.3.1 that the Euler-Lagrange equations for regular problems can be rewritten as a system of first order differential equations in the unknown functions and so called canonical variables [119], and in general will yield an extremal. For regular problems, the actuator forces can be therefore computed from the Euler-Lagrange equations.

In many cases, the system function f depends on the inputs linearly. When there are no additional equality or inequality constraints and the cost functional is defined as:

$$J = \int_{t_0}^{t_1} u^T u \, dt, \tag{3.82}$$

then $H_{uu} = 2I$, where I is the $m \times m$ identity matrix and the problem is regular [53]. This is one of the reasons that the cost functional (3.82) is popular in the optimal control community. The inputs in robotic systems are actuator forces and the cost functional (3.82) is often taken for a measure of energy consumed during the motion. The intuition behind such interpretation is that torques produced with an electro-mechanical actuator are approximately proportional to the current flowing through the motor and that the rate of energy consumption is approximately proportional to the square of the current.

The minimum torque-change criterion:

$$J = \frac{1}{2}\int_{t_0}^{t_1} \dot{\tau}^T \dot{\tau} \, dt. \tag{3.83}$$

used by Uno *et al.* [143] to model human arm motion, also has the form of (3.82) if we define:

$$u = \dot{\tau} \tag{3.84}$$

and extend the state of the system with the actuator torques τ. On the other hand, for the minimum-jerk criterion:

$$J = \frac{1}{2}\int_{t_0}^{t_1} \dddot{x}^T \dddot{x} \; dt. \tag{3.85}$$

where x is the position vector, the state equations can be defined as:

$$\begin{array}{ll} x_1 = x & \dot{x}_1 = x_2 \\ x_2 = \dot{x} & \dot{x}_2 = x_3 \\ x_3 = \ddot{x} & \dot{x}_3 = u. \end{array} \tag{3.86}$$

This is a purely kinematic problem and is singular if the inputs are the actuator forces. Therefore, it can not be used to compute the actuator trajectory.

It appears that also time-optimal control problems are singular since the cost functional does not depend on the inputs:

$$J = \int_{t_0}^{t_1} dt. \tag{3.87}$$

However, for the time-optimal control we have to satisfy the constraints:

$$U_{\min} \leq u \leq U_{\max}, \tag{3.88}$$

which become part of the side constraints φ. It is not difficult to check that because of the constraints this problem is regular.

In many motion planning problems it is desirable to obtain motion plans that are smooth. Smooth trajectories prevent jerky motions that can excite structural resonances of the system and damage the mechanical structure or the actuators. It is customary to take the L^2 norm of the derivative of a curve for a measure of its smoothness. At first it seems that minimizing the L^2 norm of higher derivatives should produce a curve that is even smoother. The following simple example shows that such conclusion would be wrong.

Suppose we are interested in smooth curves in $\mathbb{R}^2$ between points $(0,0)$ and $(1,1)$. It is not difficult to see that the curve $x(t)$ that minimizes:

$$J_1 = \int_0^1 \dot{x}^2 dt, \tag{3.89}$$

is the straight line $x_1(t) = t$. We would like to see which curve minimizes the integral:

$$J_2 = \int_0^1 \ddot{x}^2 dt. \tag{3.90}$$

On the straight line $x_1(t) = t$, the value of this cost functional is 0 and this is the global minimum! Now suppose that the initial and final velocities for the curve are specified to be 0. In this case, the curve that minimizes J_2 is the cubic $x_2(t) = -2\,t^3 + 3\,t^2$. Because we imposed additional boundary conditions on the curve, the value of the cost functional J_2 increased! Of course, also the value of J_1 for the curve $x_2(t)$ is higher than the value for the curve $x_1(t)$. This example illustrates that it would be wrong to take L^2 norm of higher derivatives of a curve as a measure of smoothness. In deciding what order of derivatives to use as a measure of the smoothness of a curve, we must be guided by the boundary conditions that the curve has to match (in general, we must also consider other constraints) and use the derivatives of the smallest possible order.

Minimum energy (3.82), minimum torque-change (1.2) and minimum-jerk (1.1) functionals can all be interpreted as measures of smoothness: the first two of the actuator trajectories and the last of the kinematic trajectories. As demonstrated above, the first

two criteria are suitable for dynamic motion planning while the last can only be used for kinematic motion planning. According to the above example, the criterion for measuring the smoothness should be inferred from the boundary conditions that need to be satisfied. For robotic systems, we often want to assure that the actuator torques do not exhibit jumps as we switch from one motion to another. This is the same as saying that the accelerations at the junction points of different arcs are the same. In general, if a boundary condition for a variable needs to be specified, the cost functional should depend on the derivatives of the variable. If we need to specify the accelerations, the cost functional should either depend on the derivative of the acceleration or if we wish to use the dynamics, the derivative of the torques. Minimum-jerk and minimum torque-change functionals depend on these derivatives, hence they can be very useful in robotics. Which of the two we pick depends on whether we are interested in kinematic or dynamic motion planning.

Chapter 4

Motion planning for two cooperating robots

In this chapter, we study two manipulators cooperatively manipulating an object. We wish to find smooth trajectories for the actuator torques hence the minimum torque-change cost functional is used to formulate the motion planning problem. Motion plans are computed for two cases. When the object is rigidly grasped, the manipulators can exert arbitrary grasp forces and the motion planning problem is unconstrained. But when the two manipulators hold the object with a friction assisted grasp, frictional inequality constraints that only depend on the state must be satisfied.

4.1 Dynamic model for two manipulators holding an object

We start by deriving the dynamic model for two manipulators holding a rigid object. Both manipulators are assumed to have n links and n actuated degrees of freedom, but we could easily extend the derivations to more general case. The interaction between each manipulator and the object is assumed to be rigid: no relative motion between the object and the manipulator is allowed. All degrees of freedom in the system thus correspond to the degrees of freedom of the two manipulators. The motion of the object is described in a m-dimensional configuration space, where $m = 3$ for the planar case and $m = 6$ in the spatial case. (See Figure 4.2 for a case when $n = 3$ and $m = 3$.)

Dynamics of the two manipulators is described by the following equations:

$$\begin{aligned} I_1(\theta_1)\ddot{\theta}_1 + C_1(\theta_1, \dot{\theta}_1) + G_1(\theta_1) &= \tau_1 - \Gamma_1(\theta_1)^T F_1 \\ I_2(\theta_2)\ddot{\theta}_2 + C_2(\theta_2, \dot{\theta}_2) + G_2(\theta_2) &= \tau_2 - \Gamma_2(\theta_2)^T F_2 \end{aligned} \tag{4.1}$$

where θ_i is the $n \times 1$ vector of the joint coordinates of the ith ($i = 1, 2$) manipulator, $I_i(\theta_i)$ is the $n \times n$ inertia matrix, $C_i(\theta_i, \dot{\theta}_i)$ is the $n \times 1$ vector of nonlinear terms (Coriolis and centrifugal forces), $G_i(\theta_i)$ is the $n \times 1$ vector of gravity terms, τ_i is the $n \times 1$ vector of the joint torques, $\Gamma_i(\theta_i)$ is the $n \times m$ Jacobian matrix relating the velocity of the center of mass of the object in the configuration space to the joint velocities of the manipulator i and F_i is the $m \times 1$ generalized force vector in the configuration space, consisting of the forces and the moments about the center of mass exerted on the object by the manipulator i.

The dynamics of the object in the configuration space is given by:

$$M\ddot{p} = F_1 + F_2 \tag{4.2}$$

where M is the $m \times m$ inertia matrix of the object and p is the $m \times 1$ vector representing the position/orientation of the object in the configuration space.

The two manipulators completely restrain the motion of the object, forming a closed kinematic chain. The kinematic closure equations can be obtained by expressing the position of the center of mass of the object (which is fixed with respect to the robot end-effector) as a function of the joint coordinates of each robot. If $p_1(\theta_1)$ is the position vector of the center of mass of the object expressed as a function of the joint coordinates of the first robot and $p_2(\theta_2)$ the position vector of the center of mass of the object expressed as a function of the joint coordinates of the second robot, then the kinematic closure equations can be written as:

$$p_1(\theta_1) - p_2(\theta_2) = 0. \tag{4.3}$$

By defining:

$$g(\theta_1, \theta_2) = p_1(\theta_1) - p_2(\theta_2), \tag{4.4}$$

it is easy to see that:

$$\begin{aligned} \Gamma_1 &= \frac{\partial g}{\partial \theta_1} \\ \Gamma_2 &= -\frac{\partial g}{\partial \theta_2}. \end{aligned} \tag{4.5}$$

The closed kinematic chain has $2n - m$ degrees of freedom. If r degrees of freedom are required to perform the task, the task can be performed as long as $r \leq (2n - m)$. If strict inequality holds, the linkage is kinematically redundant for the given task. There are also $2n$ actuators at our disposal for moving the joints. The system is therefore over-constrained and m actuators are redundant.

In dynamics, the interaction forces F_1 and F_2 due to the holonomic constraint (4.3) are also called the Lagrange multipliers. The fact that the closed chain is over-constrained and possesses m surplus actuators implies that the interaction forces that do not affect motion of the linkage can be exerted on the object. Such forces are called the *internal* or *preload* forces and are typical in tasks such as walking, grasping and multiple-arm coordination during which closed kinematic chains are formed. Internal forces can be understood through an example where two force sources act on a rigid object (Figure 4.1). For the case of manipulation with two arms, the two sources are the two arms. Assume that the object is a point mass and that the two sources exert forces F_1 and F_2 on the object. The object moves according to the equation:

$$m\ddot{p} = F_1 + F_2 \tag{4.6}$$

where m and p denote the mass and the position of the object, respectively. Now assume that the exerted forces are changed to $F_1' = F_1 + \Lambda$ and $F_2' = F_2 - \Lambda$. Obviously, the sum of the forces does not change:

$$F_1' + F_2' = F_1 + F_2,$$

so the motion of the object will not be affected by the force Λ. The force Λ is thus an internal force. Assuming that each source contributes the same share to the motion of the object, we can write:

$$\begin{aligned} F_1 &= \frac{F}{2} + \Lambda \\ F_2 &= \frac{F}{2} - \Lambda \end{aligned} \tag{4.7}$$

and the internal force is given by:

$$\Lambda = \frac{1}{2}(F_1 - F_2). \tag{4.8}$$

When we have n actuators operating in a linkage with r degrees of freedom, $r \leq n$, the set of internal forces is in general a vector space of dimension $n - r$.

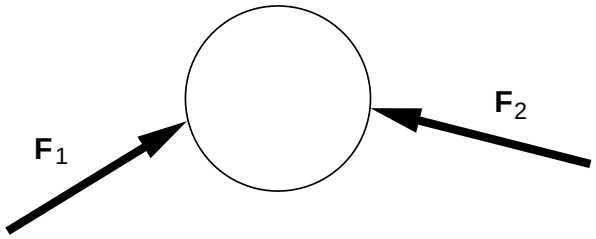


Figure 4.1: Internal force in an over-actuated system.

In robotics, kinematic and actuator redundancies are typically resolved with some sort of local optimization [69, 88]. The advantage of such schemes is that they can be used on-line as the motion is executed. However, they require a separate trajectory planning phase. Instead, by choosing a suitable cost functional and solving the resulting variational problem, trajectory planning and redundancy resolution are performed at the same time. The resulting motion plan is globally optimal and can be subsequently used to simplify the control [91].

4.2 Motion planning for two planar cooperating arms

We turn our attention to the case of two 3-R manipulators moving an object in the horizontal plane as shown in Figure 4.2. Each manipulator has three actuators ($n = 3$). The object moves in the plane and $m = 3$, hence the system has $2n - m = 3$ degrees of freedom. The task is to arbitrary position and orient the object in the plane. Three degrees of freedom are required to perform the task, so $r = 3$ and we have no kinematic redundancy. All joints of the robots are actuated, yielding a total of 6 actuators. In general, only 3 are required to move the linkage, which leaves 3 surplus actuators and gives rise to a three-dimensional space of internal forces.

Since the closed kinematic chain is not kinematically redundant, the matrices Γ_1 and Γ_2 in equations (4.1) are invertible and the forces F_1 and F_2 can be expressed from Equation (4.1). In the planar case, these forces have the form:

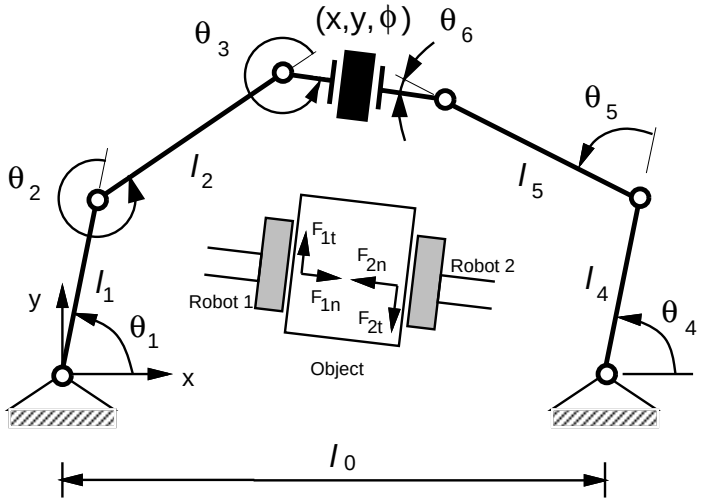


Figure 4.2: Two planar 3-R arms holding an object.

$$F_i = \begin{bmatrix} F_{i_x} \\ F_{i_y} \\ T \end{bmatrix}, \tag{4.9}$$

where F_{i_x} and F_{i_y} are the x and y components of the Cartesian force and T is the moment that the contact force produces around the center of mass of the object. Similarly, the object inertia matrix becomes:

$$M = \begin{bmatrix} m & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & I_c \end{bmatrix} \tag{4.10}$$

where m is the mass of the object and I_c the moment of inertia of the body around the center of mass. By replacing the forces F_1 and F_2 in Equation (4.2) with the expressions obtained from Equation (4.1) and expressing the position of the center of mass of the object as a function of the joint variables of the first robot, $p = p_1(\theta_1)$, a single equation describing the dynamics of the closed chain is obtained:

$$M(\Gamma_1\ddot{\theta}_1 + \dot{\Gamma}_1\dot{\theta}_1) = \Gamma_1^{-T}(\tau_1 - I_1(\theta_1)\ddot{\theta}_1 - C_1(\theta_1, \dot{\theta}_1)) + \Gamma_2^{-T}(\tau_2 - I_2(\theta_2)\ddot{\theta}_2 - C_2(\theta_2, \dot{\theta}_2)). \tag{4.11}$$

We neglected the gravity terms given that the system moves in the horizontal plane.

Next, we define the cost functional. We consider the task of moving the object from an initial position to a final position along smooth trajectories so that the initial and final accelerations are 0. As discussed in Section 3.4, to satisfy boundary conditions on the acceleration, the cost functional must depend on the derivative of the position variables of the order 3 or more. One possible choice is the minimum-jerk functional:

$$J = \int_{t_0}^{t_1} (\dddot{p}^T \dddot{p}) dt. \tag{4.12}$$

The vector p denotes the position of the object in the task-space. Analytic expressions for the kinematic trajectories that minimize the minimum-jerk cost functional can be readily calculated and are fifth-order polynomials [42].

The system of the two manipulators holding an object is over-actuated. Therefore, finding the kinematic trajectories is not enough to uniquely determine the actuator forces. To compute the actuator forces through the optimization we have to use, loosely speaking, a dynamical equivalent of the minimum-jerk cost functional. In this way, we obtain the minimum torque-change functional:

$$J = \frac{1}{2} \int_{t_0}^{t_1} \dot{\tau}^T \dot{\tau} \, dt, \tag{4.13}$$

where τ is the vector of the actuator forces (see Equation 4.11). Since the equation for the acceleration depends on τ, the minimum torque-change functional can be intuitively understood as replacing the jerk with $\dot{\tau}$ to make the cost functional depend on the dynamics. We therefore require smoothness of the actuator forces and not directly smoothness of the trajectory of the object.

4.2.1 No constraints on the interaction forces

When no constraints are imposed on the interaction forces F_i, the optimal control problem is completely defined by the dynamic equations (4.11), constraints (4.3) and the cost functional (4.13). However, the dynamic equations (4.11) have to be rewritten as a system of first order differential equations. We introduce a vector of unknowns:

$$\begin{bmatrix} X_1 \\ X_2 \end{bmatrix} = \left[\begin{array}{c} \theta_1 \\ \theta_2 \\ \hline \dot{\theta}_1 \\ \dot{\theta}_2 \end{array}\right] \tag{4.14}$$

which implies:

$$\dot{X}_1 = X_2. \tag{4.15}$$

We further define:

$$X_3 = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix}. \tag{4.16}$$

With this definitions, Equation (4.11) can be rewritten as:

$$I(X_1)\dot{X}_2 + C(X_1, X_2) + B(X_1)X_3 = 0, \tag{4.17}$$

where

$$\begin{aligned} I(X_1) &= \left[M\Gamma_1 + \Gamma_1^{-T} I_1, \Gamma_2^{-T} I_2\right] \\ C(X_1, X_2) &= M\dot{\Gamma}_1\dot{\theta}_1 + \Gamma_1^{-T} C_1 + \Gamma_2^{-T} C_2 \\ B(X_1) &= \left[-\Gamma_1^{-T}, -\Gamma_2^{-T}\right]. \end{aligned} \tag{4.18}$$

With the definition (4.16), the cost functional (4.13) becomes:

$$J = \frac{1}{2} \int_{t_0}^{t_1} \dot{X}_3^T \dot{X}_3 \, dt. \tag{4.19}$$

Note that the matrix I in Equation (4.17) is not a square matrix and is therefore not invertible. However, for the Bolza formulation of the optimal control problem the dynamic equations need not be explicit so we can directly apply the methods from Chapter 3.

Finally, as discussed in Section 3.2.1, the holonomic constraints (4.3):

$$g(\theta_1, \theta_2) = 0,$$

must be differentiated to obtain a regular problem. After differentiation and using Equation (4.5), the constraint becomes:

$$\Gamma_1 \dot{\theta}_1 + \Gamma_2 \dot{\theta}_2 = 0. \tag{4.20}$$

In addition, we have to enforce the initial condition:

$$g(\theta_1, \theta_2)|_{t_0} = 0. \tag{4.21}$$

The optimal control problem is not completely specified without the boundary conditions. The initial and the final positions of the object are specified. Furthermore, we require that the object is at rest at the beginning and end of the motion. Therefore,

$$\begin{array}{ll} X_1(t_0) = X_{1_0} & X_2(t_0) = 0, \\ X_1(t_1) = X_1 & X_2(t_1) = 0. \end{array} \tag{4.22}$$

For the manipulation task we require that the initial and final accelerations of the object are 0. Hence, there are three ways to specify the boundary conditions for the torques:

1. The actuator forces at the initial and the final time are arbitrarily specified subject to the requirement that the initial and the final accelerations are 0.

2. The internal force at the initial time is specified while the object's initial acceleration is 0. The object's acceleration at the final time must be 0 but the internal force is not specified. Alternatively, the acceleration at the initial time could be set to zero and the forces and the accelerations specified at the final time.

3. The acceleration of the object at the initial and at the final time must be 0. No internal forces are specified.

The first case is the easiest to implement, the other two cases require some additional work to determine the transversality conditions (see [119, pp. 364]). When the boundary values for the actuator forces are not directly specified, they are computed from these transversality conditions as part of the optimization.

We have not specified whether the final time t_1 is given in advance or has to be computed. For simplicity, take $t_0 = 0$ and assume that we computed the extremal for the final time t_1. Suppose we want to compute the extremal for the final time $\hat{t}_1$, where $\nu = t_1/\hat{t}_1$. It can be verified that if $\theta(t)$, $\omega(t) = \dot{\theta}(t)$ and $\tau(t)$ satisfy the side constraints and the boundary conditions on the interval $[0, t_1]$, then $\hat{\theta}(t) = \theta(\nu t)$, $\hat{\omega}(t) = \nu\, \omega(\nu t)$ and $\hat{\tau}(t) = \nu^2\, \tau(\nu t)$ is a solution on the interval $[0, \hat{t}_1]$. Further, if:

$$J = \frac{1}{2} \int_0^{t_1} \dot{\tau}^T \dot{\tau}\, dt, \tag{4.23}$$

where τ is the input computed for the extremal, the value of the cost functional for the new time $\hat{t}_1 = t_1/\nu$ is:

$$\begin{aligned} \hat{J} &= \frac{1}{2}\int_0^{\hat{t}_1} \dot{\hat{\tau}}^T \dot{\hat{\tau}}\, d\hat{t} = \\ &= \frac{1}{2\nu}\int_0^{t_1} (\nu^3\dot{\tau})^T(\nu^3\dot{\tau})\, dt = \nu^5 J. \end{aligned} \tag{4.24}$$

Therefore, as the time t_1 increases, the cost decreases with factor ν^5 ($\nu \leq 1$). It follows that:

$$\lim_{t_1\to\infty} J = 0. \tag{4.25}$$

In summary, the final time t_1 has to be fixed for the problem to be well-posed. For any other final time $\hat{t}_1$, the solution can be obtained by appropriate scaling of the solution for the final time t_1.

Another interesting observation that follows from the above is that if $X^*(t)$ is an extremal for the boundary values $X(t_0) = X_0$ and $X(t_1) = X_1$, then the extremal for the boundary values $X(t_0) = X_1$ and $X(t_1) = X_0$ is given by $X^*(t_1 - t)$. This can be useful in verifying the correctness of the numerical solutions.

4.2.2 Frictional constraints on the interaction forces

The situation where the object is completely restrained by the two robots so that arbitrary forces can be exerted on the object seldom occurs in the cooperating task. If the object is to be grasped rigidly, special grippers have to used and such grippers can not grasp objects of arbitrary shape. Instead, it is more common to restrain the motion of the object with frictional forces. With such grasp, the normal forces must be positive and they must produce sufficiently large frictional forces to completely immobilize the object.

We perform the analysis of the frictional contact in the contact frame. The origin of this frame corresponds to the point of contact of the robot with the object (or the centroid of the contact area if this area is not a point), the x axis corresponds to the inward normal to the surface of the object and the y axis to the tangent. The contact force F_i is transformed to the force ${}^cF_i$ in the contact frame according to:

$${}^cF_i = R(\theta_i)F_i, \tag{4.26}$$

where the 3×3 screw transformation $R(\theta_i)$ transforms the forces from the coordinate frame at the center of mass of the object (with the axes parallel to the inertial frame) to the contact frame at the contact point i. Contact forces are computed from Equation (4.1):

$$F_i = \Gamma_i^{-T}\left(\tau_i - I_i(\theta_i)\ddot{\theta}_i - C_i(\theta_i, \dot{\theta}_i)\right). \tag{4.27}$$

We denote the normal and tangential components of the contact force ${}^cF_i$ by ${}^cF_{i_n}$ and ${}^cF_{i_t}$, respectively:

$${}^cF_i = \begin{bmatrix} {}^cF_{i_n} \\ {}^cF_{i_t} \\ {}^cT \end{bmatrix}. \tag{4.28}$$

Normal force must be positive in order to produce the frictional force. The first constraint is therefore:

$$ {}^{c}F_{i_n} \geq 0. \tag{4.29} $$

Tangential forces are frictional forces. If μ is the coefficient of friction, the tangential forces hence satisfy:

$$ |{}^{c}F_{i_t}| \leq \mu F_{1_n}. \tag{4.30} $$

The last inequality can be rewritten as a pair of constraints[1]:

$$ {}^{c}F_{i_t} \leq \mu F_{i_n}, \qquad -\mu F_{i_n} \leq {}^{c}F_{i_t}. \tag{4.31} $$

We can rewrite the frictional contact constraints in a matrix form:

$$ D\ {}^{c}F_i = 0, \tag{4.32} $$

where D is a constant 3×3 matrix:

$$ D = \begin{bmatrix} -1 & 0 & 0 \\ -\mu & 1 & 0 \\ -\mu & -1 & 0 \end{bmatrix}. \tag{4.33} $$

Therefore, in their final form, the inequality constraints describing a frictional contact, are:

$$ DR(\theta_i)\Gamma_i^{-T}\left(\tau_i - I_i(\theta_i)\ddot{\theta}_i - C_i(\theta_i, \dot{\theta}_i)\right) \leq 0, \quad i = 1, 2. \tag{4.34} $$

Note that because of the choice of the cost functional, the torques τ are part of the state and the inequality constraints do not depend explicitly on the inputs!

4.3 Numerical results

To solve the optimal control problem, we follow the method from Section 3.2. In this example, the method produced a local minimum (we established this by checking the second order necessary conditions) so we did not use the more sophisticated method from Section 3.3. We define the Hamiltonian $\hat{H}$ according to Equation (2.43) and Proposition 2.13. Subsequently, a system of algebraic equations of the form (3.7) is obtained. The vector of unknown functions X has dimension 18. The dynamic equations (4.15), (4.17) and the constraints (4.20) define 12 side constraints which are adjoined to $\hat{H}$ with 12 multipliers. A total of 30 unknown functions must be therefore computed to obtain an extremal.

When the contact forces are constrained, we proceed as before except that the constraints (4.34) are transformed into equality constraints with the slack variables and adjoined to the Hamiltonian with the Lagrange multipliers. The number of unknown functions when the inequality frictional constraints have to be taken into account thus increases to 42.

Any suitable numerical method can be used to find a solution of the system of algebraic equations (3.7). We decided to use the Newton-Raphson method. The idea of the method

[1] In the spatial case we can write ${}^{c}F_{i_t}^2 \leq \mu^2 F_{i_n}^2$.

is quite simple. We want to find a zero of the (vector) function $F(x)$. If x_0 is the current approximation of the solution, we are looking for a correction Δx such that:

$$F(x_0 + \Delta x) = 0. \tag{4.35}$$

The first-order Taylor expansion yields:

$$F(x_0 + \Delta x) \approx F(x_0) + \left.\frac{\partial F}{\partial x}\right|_{x_0} \Delta x, \tag{4.36}$$

which implies that a good approximation to the desired vector of corrections is:

$$\Delta x \approx - \left.\frac{\partial F}{\partial x}^{-1}\right|_{x_0} F(x_0). \tag{4.37}$$

In each iteration we have to compute and invert the matrix of gradients $E = \frac{\partial F}{\partial x}$. Since the number of unknowns (dimension of the system of algebraic equations) is MN, where M is the number of unknown functions and N the number of mesh points, inversion of the matrix E is a time consuming process. However, Equation (3.7) only depends on the values of the unknown functions at three adjacent mesh points. Matrix E is therefore block-tridiagonal and can be inverted in time $\mathcal{O}(NM^3)$ instead of $\mathcal{O}(N^3M^3)$.

The second simplification in our numerical scheme involves computation of the entries of E. This entries should be computed analytically. For any problem of practical significance, Equations (3.7) are already quite complicated and computation of the partial derivatives would be a laborious process. The entries in E are therefore computed numerically using forward difference scheme [111].

For the simulation, the two manipulators were assumed identical and they were modeled by the human arm. The dimensions of the manipulators were:

Link	Length m	Mass kg	Inertia 10^{-3} kg m^2
1	0.29	2.0	18.0
2	0.27	1.2	7.6
3	0.07	0.4	0.3

The object was a rectangle of length 0.2m, width 0.1m, mass 1.3kg and moment of inertia 5.4 10^{-3} kg m^2. The coefficient of friction was $\mu = 0.5$ when frictional constraints were taken into account. Duration of motion was 1s.

As discussed in Section 4.2.1, boundary conditions for the actuator forces can be determined in different ways. For numerical simulations, we specified the initial preload (compressive) force and enforced zero acceleration at the beginning and end of the motion (case 2 on page 53).

Figure 4.3 compares the trajectories of the unconstrained motion (dotted) with two constrained motions: one with the normal preload force of 0.1N (solid) and the other 2.0N (dashed). The figure shows that the trajectory is curved for the unconstrained case and it is almost a straight line for the two constrained cases. Varying the preload force only slightly changes the trajectory.

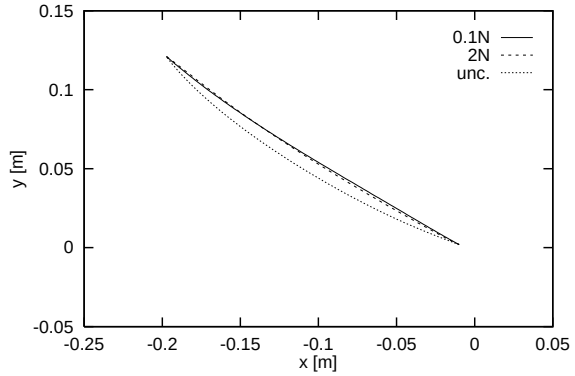


Figure 4.3: Trajectories of the center of mass of the object.

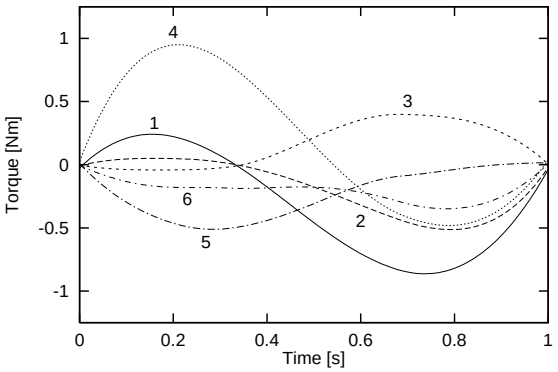


Figure 4.4: Optimal distribution of the torques for the preload force of 0.1N.

The calculated optimal distribution of the torques for the constrained case with the preload force of 0.1N is shown in Figure 4.4. The numbers denote the joints of the robots, the first three corresponding to the first robot and the last three to the second robot. The figure indicates that because of the constraints on the contact forces, the torques do not change smoothly at the junctions of constrained and unconstrained arcs (compare with Figure 4.5). The inputs (torque derivatives) are continuous but not smooth. Analysis of the normal forces (not presented on the figure) shows that the constraint (4.29) never becomes active, the normal force is always strictly greater than 0.

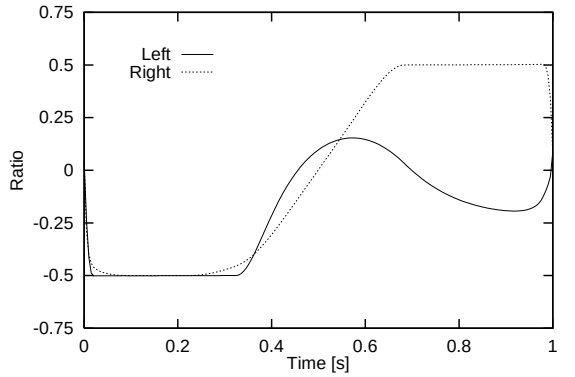


Figure 4.5: Ratio of the tangential vs. normal force for the preload force of 0.1N.

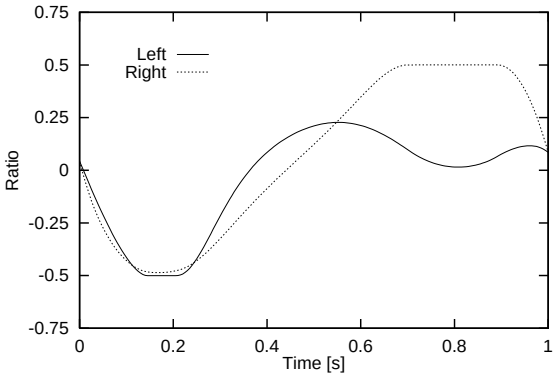


Figure 4.6: Ratio of the tangential vs. normal force for the preload force of 2.0N.

Figures 4.5 and 4.6 show the ratios of the frictional and normal forces for the two constrained motions. The ratios clearly show when the constraints are active. On Figure 4.5 we can see that the tangential force at the right contact has two constrained portions, with unconstrained parts at both endpoints and in the middle. There are four points at which the unconstrained and the constrained arcs are joined together implying that there are 4 corner points. Similar analysis shows that 2 additional corner points occur because of the frictional constraint for the left arm. We can also see that both arms simultaneously move along constrained arcs for a period of time in the first half of motion. The system as a whole thus exhibits quite complicated behavior as the number of active constraints changes

throughout the motion. Figure 4.6 shows that the behavior of the system changes when the preload force increases. Increased preload force results in a considerable normal force, giving "more room" to the frictional component. Consequently, there are fewer regions where the constraints are active and each tangential contact force only saturates once. The comparison of the two constrained motions also shows that it is difficult to predict the structure of the solution and the number of corner points.

4.4 Discussion

We studied the example of two arms manipulating an object through frictional point contacts. The task was to find a dynamically smooth trajectory for the system while optimally distributing forces between the actuators. To satisfy boundary conditions on the accelerations, inputs to the system had to be the derivatives of the actuator forces. As a result, inequalities representing the frictional constraints only depended on the state.

This example is representative of tasks such as multifingered grasping, multiarm manipulation, walking and running. In the past, such problems have been typically solved using local optimization schemes in which the task space trajectory is determined independently, often in an *ad hoc* fashion. In contrast, our method provides globally optimal solutions for the force distribution (actuator forces) as well as for the joint trajectories.

Several advantages of solving the optimal control problem by converting it into a variational calculus problem are apparent from the example. The variational formulation allowed us to use Equation (4.17) directly, without rewriting it in the state-space form. Such implicit differential equations are obtained whenever the Lagrange multipliers are eliminated from the dynamic equations. To transform the system into the state-space form, we would also have to eliminate the extra state variables using the kinematic closure equations. Such elimination is not feasible unless the robot kinematics is analytically invertible.

In the variational calculus problems, Hamiltonian is allowed to depend on the derivatives of the unknowns. For this reason we did not eliminate the accelerations $\ddot{\theta}_1$ and $\ddot{\theta}_2$ from the constraint equations (4.34). This again simplifies the formulation of the problem.

We conclude this chapter with a brief note about the computational efficiency of the numerical method. For quite arbitrary initial guess (straight lines between the boundary values where these are known, zero functions otherwise) about 20 steps of Newton-Raphson method were performed to obtain the solution with the accuracy 10^{-6}. The number of iterations did not vary with the number of discretization points. For 200 points, each Newton-Raphson iteration involved calculating the coefficients and solving a system of $8400 = 200 \times 42$ linear equations. This computation takes approximately 45s on Silicon Graphics Indigo 2.

Chapter 5

Modeling of human two-arm motion planning

Motion planning for robotic systems and modeling of principles that govern motion planning in humans are complementary problems: in the first case we try to generate the trajectories for the system, while in the second case we measure the trajectories and try to discover how they were generated. A principle that governs human trajectory generation can be used for robot trajectory planning. In this chapter we proceed in the other direction: we try to model human trajectory generation using principles that were developed for robot motion planning.

Based on studies of single arm reaching in humans, Flash and Hogan [42] suggested that the central nervous system uses an optimality criterion to plan the trajectory. They observed that medium speed, large amplitude, unconstrained planar motions for a single arm can be well described with trajectories minimizing the integral of the jerk. These *minimum-jerk* trajectories [42] depend only on the kinematics of the task. Kawato et al. [143, 65] proposed an alternative cost functional to model human trajectory generation: the integral of the norm of the vector of derivatives of the actuator torques. This cost functional, called *the minimum torque-change* criterion, takes into account the dynamics of the system. According to the studies of coordinated manipulation with two arms by humans [45], the success of the minimum-jerk model in accounting for two-arm trajectories is limited. In this chapter we investigate if the minimum torque-change criterion better describes trajectories for human two-arm manipulation. We present experimental observations for the task of positioning and orienting an object firmly grasped by two hands. Three types of motions are analyzed and it is demonstrated that during these motions the force distribution between the two arms is asymmetric and depends on the configuration. The model developed in the previous chapter for planning the motion for two cooperating manipulators is proposed for modeling of the data. Because of the asymmetry of the force distribution, the model is only applied when the manipulation task is symmetric for the two arms. For such motions, kinematic features of the trajectories and force distribution between the two arms observed in humans are well predicted by the model.

5.1 Experimental results

Two-arm planar motions of human subjects were recorded using the three degree of freedom planar passive manipulandum shown in Figure 5.1. The apparatus consists of three links, connected by revolute joints. The third link is a handlebar that can rotate around its center. Three optical encoders mounted at each joint are used to record the manipulandum joint angles. Six axis force sensors are mounted underneath each handle, allowing measurement of the forces and torques exerted on the handles by the subject. The encoders and the force sensors are sampled at 150 Hz.

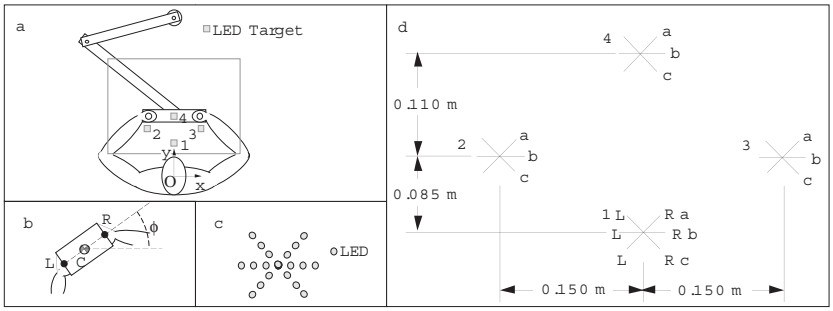


Figure 5.1: Experimental setup.

The subject is seated in front of a transparent Plexiglas plate and firmly grasps the two handles of the handlebar (Figure 5.1a). The plate lies horizontally at the level of the subject's chin. Four target sets are mounted on the plate as shown in Figure 5.1.d. Target 1 is 30cm in front of the subject's chest, Target 2 (Target 3) is displaced by 8.5cm forward and 15cm to the left (right) of Target 1, and Target 4 is displaced 19.5cm forward with respect to Target 1. Each target set consists of arrays of light emitting diodes (LED's) that can specify target positions and orientations (Figure 5.1c). For detailed description of the experimental setup we refer the reader to [44].

During the experiment, the room was darkened and a computer generated sequence of target configurations was displayed. Subjects were instructed to move the handlebar at their preferred velocity to the position and orientation specified by the lit targets. They were instructed to prevent slippage between the hands and the handles and to keep the elbows in the shoulder plane (so that a 1-1 correspondence between the manipulandum joint angles and the shoulder, elbow and wrist angles of the human arms was established). In all the experiments reported here, the subjects were instructed to keep the handlebar parallel to the frontal plane. Four subjects participated in the study: two right-handed, one left-handed and one ambidextrous. Three types of motions were studied: a) motions parallel to the frontal plane ($2 \rightarrow 3$ and $3 \rightarrow 2$ in Figure 5.1a), b) motions in the sagittal plane ($1 \rightarrow 4$ and $4 \rightarrow 1$), and c) oblique motions ($1 \rightarrow 2$ and $1 \rightarrow 3$). Velocities and forces exerted on the handles by the left and the right arm were studied for each motion.

All quantities are expressed in a coordinate frame located at the first target with the x-axis normal to the sagittal plane and pointing to the left and y-axis normal to the frontal plane and pointing forward. Only those components for which the prescribed amplitudes were non-zero are considered (e.g., the y components of motions parallel to the frontal plane are disregarded). These components will be referred to as "significant components".

Because of the large number of measured trajectories, we only present some representative plots to give an idea of the experiments and the data that was collected.

5.1.1 General observations

Velocity profiles: Some examples of normalized velocity profiles are shown in Figure 5.2 for different subjects. Only the significant components are shown. The velocity profiles are bell-shaped but the shape is not symmetric and the peak occurs on average at around 43% of the duration of motion. The rise to the peak is steeper than the fall to zero. Since the subjects did not have visual (end-point) feedback of the handle, it is not clear whether this effect can be attributed to a slowing down in order to achieve accurate positioning.

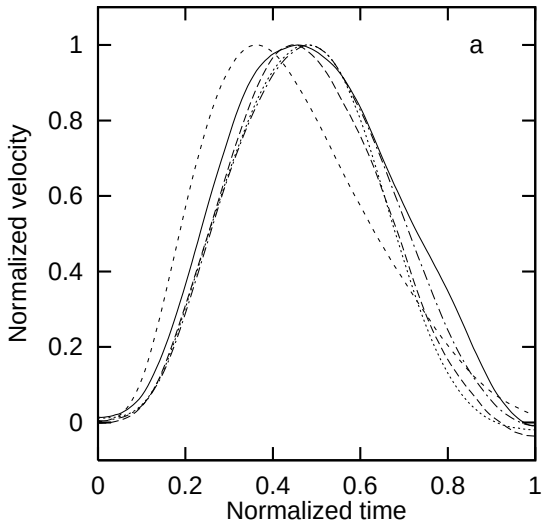


Figure 5.2: Sample plots of normalized velocity profiles measured during the experiments.

Force profiles: Some representative plots of the normalized significant components of the total force acting on the handle are shown in Figure 5.3a. The forces exerted on the object are roughly sinusoidal. The force profile for a particular subject is in general invariant with respect to the direction of motion, but it varies from person to person. In part, this difference can be attributed to different speeds at which different subjects performed the motions. At higher speeds (solid line in Figure 5.3a), the profile is roughly symmetric with the peak and the following valley having approximately the same duration and amplitude. At lower speeds, the peak was considerably sharper and higher than the valley.

Force distribution: Representative trajectories showing the axial component of the internal forces (F_a) are shown in Figure 5.3b. During motions with components in the x direction, the internal force component was nonzero. In other words, the distribution of forces between the two arms in the x direction is asymmetric. In contrast, the arm forces in the y direction were roughly equal.

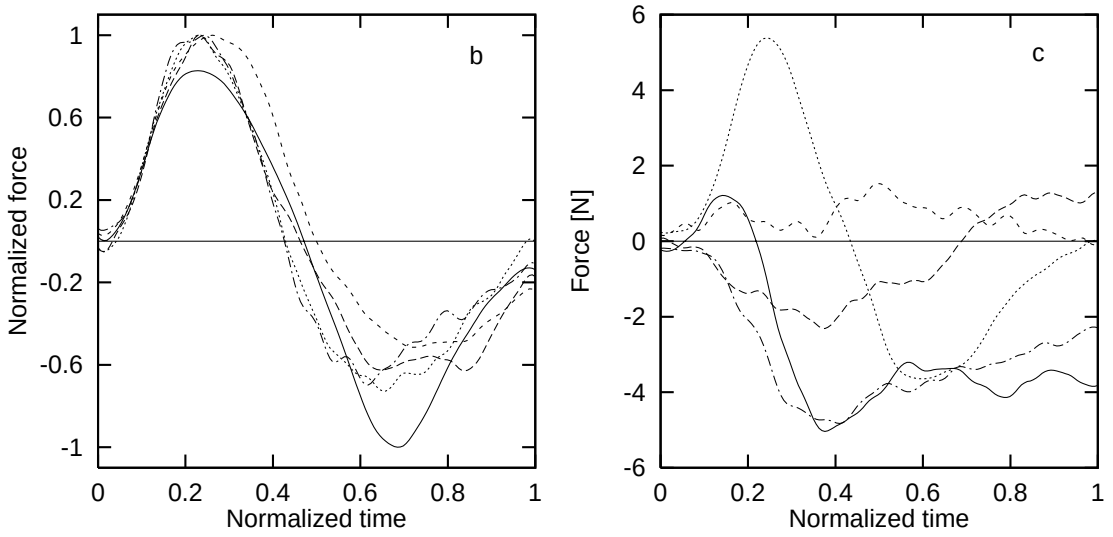


Figure 5.3: Sample plots of the measured forces: a) Total force acting on the object. b) Internal forces in the axial direction.

Remark 5.1 Some of these observations are a natural consequence of the physics of the task and are not surprising. If the object is at rest at the beginning and at the end of motion, and the manipulation task is otherwise unconstrained, one can expect to see a bell-shaped velocity profile. The shape of the force profile is linked to the shape of the velocity profile because the total force is proportional to the acceleration. If the velocity is bell-shaped, the acceleration will be sinusoidal. The symmetry of the force distribution in the y direction can also be explained. The subjects did not exert large torsional moments at the handles which implies that it is not possible to exert internal forces with significant components in the y direction.

5.1.2 Motions in the sagittal plane

The Cartesian trajectories of the object for motions that are nominally in the sagittal plane are shown in Figure 5.4a. Inward trajectories (solid) are shown shifted to the left so that they can be compared to the outward trajectories (dotted). An interesting observation is that the trajectories are always curved. Further, outward trajectories ($1 \rightarrow 4$) curve in the opposite direction as inward trajectories ($4 \rightarrow 1$). For both, left and right-handed subjects, outward trajectories curve to the right while inward trajectories curve to the left. The corresponding internal forces (axial component only) are shown in Figure 5.4b. The axial component of the internal forces affects the forces in the x direction. When the subjects moved inwards, they exerted compressive forces (positive internal forces) on the handle, while on the way outwards the handle was in tension (negative internal forces). This appears to be due to the natural tendency of the arms to move closer together on the inward path, while on the way outward they tend to move apart. Analysis of the individual arm forces (not shown) indicates that in all cases the force exerted by the right arm had slightly higher amplitude (this was also true for the left-handed subject). Thus, for the inward motion, when the arms compress the handle, we can expect the trajectory to curve

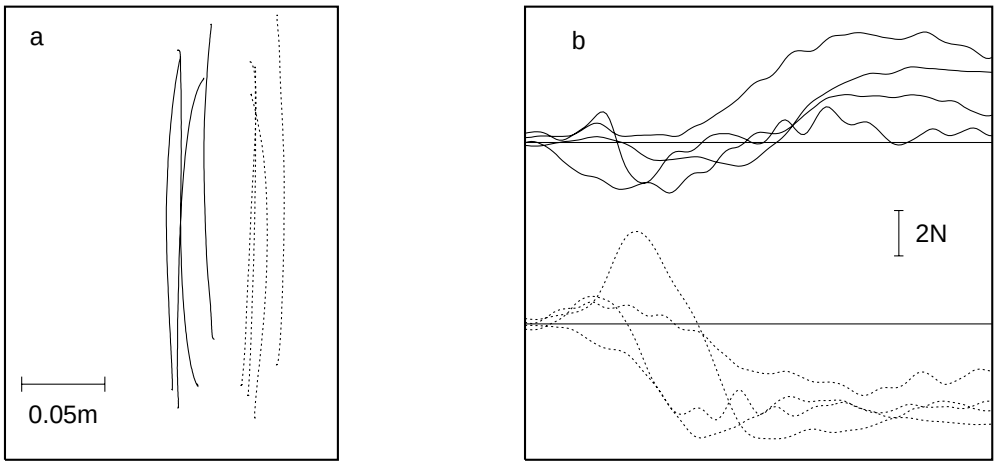


Figure 5.4: Motions in the sagittal plane. a) Trajectories for inward (solid) and outward (dotted) motions. b) Internal forces in the axial direction.

toward the left as corroborated by our observations. A similar argument can be made for the outward motion.

5.1.3 Other motions

For motions parallel to the frontal plane and oblique motions, the amplitude of motion in the x direction is large. Common to all motions with large amplitude in the x direction is the asymmetry of the forces in the x direction and presence of relatively large internal forces (the axial component). Figure 5.3b shows that the internal forces tend to start at zero, become compressive (positive) for a short period and are afterwards tensile (negative). This behavior can be observed for left and right-handed subjects. Hence there is no ground to believe that the dominance of one arm over the other plays an important role in the distribution of forces. It is also interesting that the total force exerted on the object does not vary much from subject to subject, but the distribution of the forces between the two arms varies greatly.

5.2 Optimal forces and trajectories predicted by the model

Human two-arm manipulation task is modeled by two 3-R planar manipulators holding a rigid object. Motion planning for this system was studied in Chapter 4. During the experiments, the subjects firmly grasped the handle of the manipulandum. No constraints on the contact forces are thus assumed in the model (this case was described in Section 4.2.1). We have seen in the previous section that the force distribution in the x direction varies from subject to subject. Only for the motions in the sagittal plane the variability is small so we limit the comparison of the model with the experimental data only to this case. We chose a typical data set for a sagittal plane motion, $4 \rightarrow 1$. From this data set we obtained

the boundary values for the optimization and then calculated the minimum torque-change trajectories. The physical dimensions of the arms for the optimization were taken from the measurements of the subject. Dynamic parameters (mass and moments of inertia) of the human arm were calculated from the normalized anthropometric measurements reported by Winter [153].

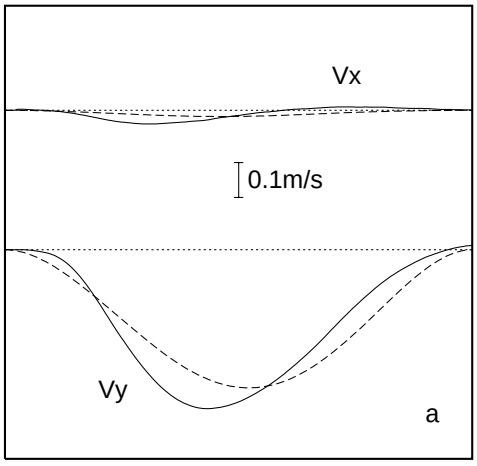


Figure 5.5: Comparison of the measured velocity profiles (solid) with the velocity profile predicted by the model (dashed).

Figure 5.5 compares the velocity profiles predicted by the simulation (dashed) with the measured ones (solid). As expected, the simulation predicts that the velocity in the x direction will be close to 0. It is slightly curved because the measured initial and final points did not have the same x coordinate. The measured x velocity profile, on the other hand, considerably deviates from 0 because the measured trajectories are curved (Figure 5.4a). The predicted and observed y-velocity profiles have similar shapes, but there is also some discrepancy between the two: the measured trajectory reaches the valley at 43% of movement duration, while the computed trajectory attains its minimum at 53%. This disparity is also reflected in the amplitudes.

The comparison of the measured forces with those predicted by the simulation is shown in Figs. 5.6a and 5.6b. The agreement between the simulated and the measured forces is reasonable except for the force exerted by the right arm in the x direction where it is poor (Figure 5.6b). This discrepancy is due to an apparent dominance of the right arm over the left arm which was observed for both, left and right-handed subjects. The optimality criterion, on the other hand, does not incorporate such dominance. The discrepancy in the y force component is consistent with the differences of the velocity profiles in the y direction. The measured force profile exhibits a valley that is sharper than the ensuing peak. The two extrema (the valley and the peak) are also not symmetric about the midpoint. In contrast, the optimization results are fairly symmetric.

Comparison of the computed and measured trajectories shows that the minimum torque-change model correctly predicts some features of the motion, at least when the effects causing asymmetric distribution of the load between the two arms are small. These

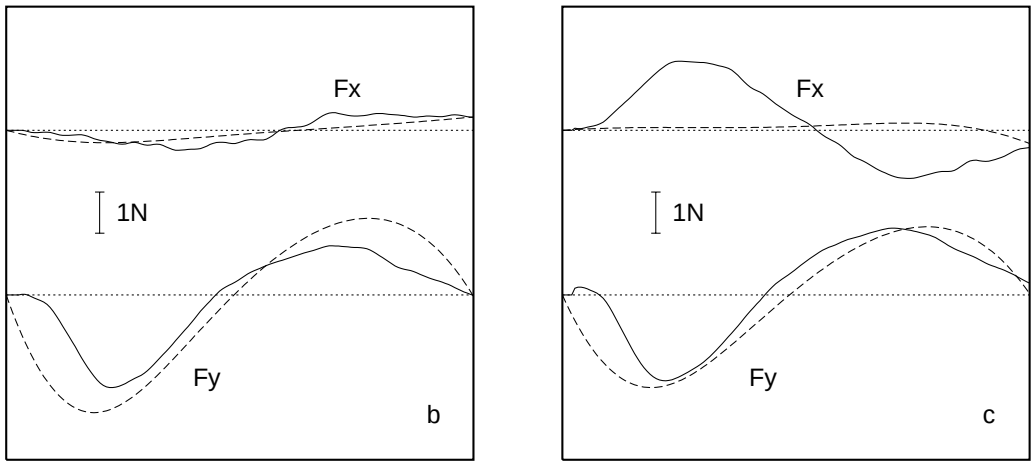


Figure 5.6: Comparison of the measured forces (solid) with the predictions of the model (dashed): a) Left-arm forces. b) Right-arm forces.

features include the general shape of the trajectories, velocities and force distributions. While we have only presented results for one instance of motion, this set of data is typical of what we have observed with other subjects as well. On the other hand, there are important differences between the model predictions and the data. Most notably, the calculated velocity profile is quite symmetric while all the measured velocity profiles are consistently asymmetric. At the force level, the major discrepancy is in the x component. There is an apparent dominance of the right arm over the left arm, and neither our dynamic model nor the minimum torque-change criterion incorporates this effect. It is worth noting, however, that this dominance appears to be independent of whether the subject is left-handed, right-handed or ambidextrous. Finally, we have not attempted to compare the predicted results with the measurements for the other motions (when the amplitude of motion in the x direction is significant), because of the variability of the force data across subjects.

5.3 Discussion

In this chapter, the objective was to study and model human two-arm motions. We reported on experiments of human subjects performing a planar manipulation task with two arms and attempted to explain the observed behavior with a model based on the minimum torque-change criterion.

The observed trajectories were approximately straight lines with bell-shaped velocity profiles (along the x and y directions) and they were quite repeatable. The measured force distributions among the two arms were markedly asymmetric in the x direction and symmetric in the y direction. However, no consistent pattern could be observed among different subjects except in sagittal plane movements, where the right arm appeared to be dominant (regardless of whether the subjects were left or right-handed).

In the minimum torque-change model, the objective function is the integral of the

vector of derivatives of the actuator forces. Unlike other models, the minimum torque-change criterion predicts the internal forces in addition to the trajectories of the system, and thus resolves indeterminacies in the task space, joint space and actuator space. This is especially important in the two-arm manipulation task where the internal forces can be seen as a measure of the interaction between the two hands. The minimum torque-change model seemed to reasonably predict the kinematic characteristics of the experimentally observed motions and the total force histories. However, it was unable to account for the observed dominance of the right arm over the left arm in sagittal plane motions.

While single joint movements and single arm movements have been studied extensively, this work (see also [45]) is the first quantitative study of human manipulation using two arms. The initial motivation for this study was to understand human motion planning with the aim of improving control and planning for multiple-arm robotic systems. The experimental data was subsequently used to investigate if the method developed for motion planning in robotics can be used to model human trajectory generation.

Chapter 6

Motion planning for systems with discrete and continuous state

When we stated the motion planning problem we implicitly assumed that the dynamic description of the system does not change during the task. In many robotic applications this assumption does not hold and the dynamic equations change as the system moves. Take for example a multi-fingered hand holding an object. If the set of fingers which are in contact with the object does not change the motion of the object relative to the wrist is described by a set of ordinary differential equations [92] and a motion plan for such a system can be generated by "traditional" techniques. Now consider regrasping of the object where another finger comes in contact or one of the fingers is withdrawn from the object. The occurrence of any such events causes a change in the system of ordinary differential equations describing the state of the system, and a change in the algebraic constraints [78]. For systems of this type, the state space can be therefore partitioned into regions so that the motion of the system in each of the regions is governed by a different set of ordinary differential equations. These regions can be viewed as separate discrete states with the differential equations within each discrete state describing the evolution of the continuous state. Successful completion of a task necessitates transitions between the discrete states and a motion plan therefore consists of a sequence of discrete states as well as a continuous trajectory of the system within each discrete state. In the example of a multi-fingered hand holding an object, the sequence of discrete states corresponds to the grasp gait [29, 76]; for a walking machine, different sequences represent different gaits.

The systems described above fall into the general class of hybrid systems. These are control systems that involve continuous dynamics, discrete phenomena that may change the behavior of the system, as well as continuous or discrete control laws. There are different approaches to modeling of hybrid systems [50]. The most general model was proposed by Branicky *et al.* [16]. Suitable for modeling the systems described above are also models of Brockett [20], and of Kohn and Nerode [98]. Motion planning for a hybrid systems is difficult and is subject of much research. The hybrid systems framework of Branicky *et al.* [16] is used to prove the existence of optimal motion plans. Computational techniques for finding optimal hybrid control based on dynamic programming are proposed in [17], but they have prohibitive complexity and are thus not practical for solving complex motion planning problems.

This chapter describes a method for generating continuous motion plans for a restricted class of hybrid systems that are encountered in robotics. In robotic tasks, we require that the state variables, such as positions and velocities (and in some applications, accelerations and forces) must vary "smoothly" (precise definition of smoothness depends on the application and the model chosen to describe a task). We are therefore interested in generating smooth motion plans that allow transitions between discrete states. Part of the motion plan should be a sequence of discrete states (for example, the walking or grasp gait). According to the underlying idea of this dissertation, the motion planning problem is formulated as an optimal control problem and we achieve smoothness by specifying an appropriate cost function. We can also prescribe constraints at the points where the system switches between discrete states. The resulting optimal control problem is very complex, so we simplify it by assuming that the sequence of discrete states is known. We argue that in many robotic applications this sequence can be obtained separately. We develop a technique that uses the given sequence of discrete states to convert a problem with unknown switching points to a problem in the standard form [22].

We illustrate the method by computing smooth motion plans for fine manipulation of a circular object with a two-fingered hand. In this case, the discrete states correspond to different grasp configurations and the motion plan consists of a sequence of grasp configurations (discrete states) as well as a trajectory (evolution of the continuous state) within each grasp configuration. Given a sequence of grasp configurations (grasp gait), we generate a smooth motion plan for moving the system through these configurations. The motion plan also specifies when to switch between different configurations.

6.1 Problem formulation

Let the continuous state space $\mathcal{X}$ of the dynamical system with n states be given by:

$$\mathcal{X} = \bigcup_{j=1}^{p} D_j, \tag{6.1}$$

where D_j are pairwise disjoint, connected subsets of $\mathbb{R}^n$. On each subset D_j, the system is described with system equations:

$$\dot{x} = F_j(x, u, t), \tag{6.2}$$

end algebraic constraints:

$$\begin{aligned} G_j(x, u, t) &= 0, \\ H_j(x, u, t) &\leq 0. \end{aligned}$$

The vector $x \in \mathcal{X} \subset \mathbb{R}^n$ is the (continuous) state of the system, $u \in \mathbb{R}^m$, F_j is a (smooth) vector field and G_j and H_j are smooth (vector) functions. The sets D_j are called discrete states and they partition the state space into regions so that on each region the evolution of the system is governed by a (different) vector field F_j and is subject to a different set of algebraic constraints. We require that the continuous state changes continuously (but not smoothly) between the regions.

Since the sets D_j are disjoint, given a continuous state x, there is a unique set $D_{j(x)}$ such that $x \in D_{j(x)}$. We can therefore rewrite (6.2) as:

$$\dot{x} = f(x, u, t), \tag{6.3}$$

subject to:

$$\begin{aligned} g(x, u, t) &= 0, \\ h(x, u, t) &\leq 0, \end{aligned}$$

where $f(x, u, t) = F_{j(x)}(x, u, t)$, $g(x, u, t) = G_{j(x)}(x, u, t)$, and $h(x, u, t) = H_{j(x)}(x, u, t)$.

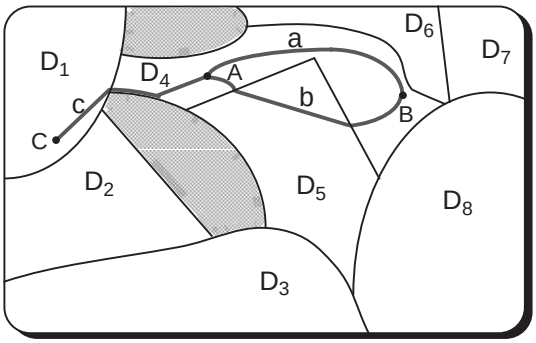


Figure 6.1: A schematic of a system with changing dynamic behavior.

Figure 6.1 schematically illustrates such a system. The continuous state space is partitioned into eight discrete states (regions). The shaded areas in the state space are regions which are not accessible (for example, where the robot would penetrate an obstacle). Three sample trajectories, denoted by a, b, and c, are shown. Given a starting point A, and an end point B, it may be possible to go from A to B without changing the discrete state (system equations) following the trajectory a. Alternatively, the trajectory b exhibits a change from D_4 to D_5 and then a transition back to D_4. The optimal path between A and C may follow a straight line in D_4 until it hits a boundary, travel along the boundary until a state transition to D_1 occurs and then follow a straight line in D_1. Since along a the continuous state evolves within the same discrete state, the trajectory is smooth. On the other hand, b and c are only piecewise smooth.

We now define the motion planning problem for the system (6.3). Our premise is that the task provides a way to evaluate the performance of different motion plans, where the performance can be formally described with a cost functional. We can therefore regard the motion planning problem as an optimal control problem:

Problem 6.1 *Given a system described by (6.3), an initial state x_0 and a final state x_1, find a set of inputs, u, that minimizes the cost functional:*

$$J = \int_{T_0}^{T_1} L(x, u, t)dt. \tag{6.4}$$

To each trajectory $\gamma(x, u, t)$ of (6.3) that connects state x_0 with x_1, there corresponds a sequence of points $T_0 = t_0 < t_1 < \ldots < t_{N+1} = T_1$ and a sequence of indices $j_0, \ldots, j_N$ such that on the interval $[t_i, t_{i+1}]$, the trajectory belongs to the set D_{j_i} and at the time t_{i+1} it switches from the region D_{j_i} to $D_{j_{i+1}}$. (Times t_i thus correspond to switches in the discrete state of the system.) This implies that a solution of the optimal control problem consists of four components:

(a) The number of switches, N, between discrete states.

(b) The sequence, $\{D_{j_i}\}_{i=0}^{N}$, of discrete states (regions) that the system traverses as it moves along the optimal trajectory.

(c) The sequence, $\{t_i\}_{i=1}^{N}$, of times when the switches occur.

(d) The trajectories for the continuous state and the inputs on each interval $[t_i, t_{i+1}]$.

It is difficult to find all four components of the optimal solution (see also [17, 71]). One of the reasons is that in general the optimal solution depends on the sequence of discrete states in very complicated way: even if two sequences are almost identical, trajectories of the continuous state can be very different. For this reason, we can not use minimization methods to find the best sequence and we must perform an exhaustive search of the space of all sequences (assuming that we can limit the length of the sequence). To find a suboptimal solution, we can solve the problem hierarchically: we first find a sequence of discrete states and then compute the trajectory for the continuous state which is optimal for the chosen sequence. The sequence of discrete states can be, for example, the shortest path in a weighted graph that approximates the optimization problem. We develop a method for computing the optimal continuous trajectory for a given sequence of discrete states in the next section.

6.2 Method for computing switching times

In tasks such as grasping with a multi-fingered hand or moving a legged mechanism on the ground, the sequence of discrete states is usually known *a priori*. For the grasping task it can be obtained by investigating feasible grasp gaits [76]. In walking, the gait is usually computed in advance to avoid regions that are unsuitable for foot placement [14, 28]. Another instance where the sequence of discrete states is *a priori* known is planning open-loop trajectories for transition between a free motion and position control to a constrained motion and force control [59, 104].

When the sequence of discrete states is given, the optimal control problem 6.1 reduces to finding the optimal switching times and the optimal trajectories for the continuous state. This problem is still difficult to solve, but we show that it can be reduced to a simpler problem. The main idea is to make the unknown switching times part of the state and introduce a new independent variable with respect to which the switching times are fixed. The resulting optimal control problem can then be solved using available methods.

Assume that the number N and the sequence of discrete states $\{D_{j_i}\}_{i=0}^{N}$ through which the system evolves are known. Without loss of generality, we can assume that $T_0 = t_0 = 0$ and $T_1 = t_{N+1} = 1$. We can proceed in a similar way as in solving boundary value

problems with unknown terminal time [115]. The first step is to introduce new state variables $x_{n+1}, \ldots, x_{n+N}$ corresponding to the switching times t_i with

$$\begin{aligned} x_{n+i} &= t_i \\ \dot{x}_{n+i} &= 0. \end{aligned} \tag{6.5}$$

We next introduce a new independent variable, s. The relation between s and t is linear, but the slope of the curve changes on each interval $[t_i, t_{i+1}]$. We establish piecewise linear correspondence between time t and the new independent variable s so that at every chosen fixed point s_i, where $i = 0, \ldots, N+1$, t equals t_i. For convenience we set $s_i = i/(N+1)$, but any monotonically increasing sequence of N numbers on interval $[0, 1]$ could be used. Figure 6.2 illustrates this idea. As a result we obtain the following expressions:

$$t = \begin{cases} (N+1)x_{n+1}s, & 0 \le s \le \frac{1}{N+1} \\ \ldots & \\ (N+1)(x_{n+i+1} - x_{n+i})s & \\ +(i+1)x_{n+i} - ix_{n+i+1}, & \frac{i}{N+1} < s \le \frac{i+1}{N+1} \\ \ldots & \\ (N+1)(1 - x_{n+N})s & \\ +(N+1)x_{n+N} - N, & \frac{N}{N+1} < s \le 1. \end{cases}$$

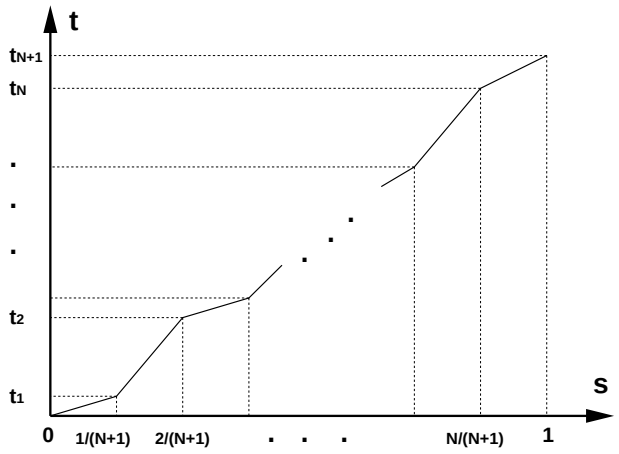


Figure 6.2: A new independent variable has fixed values at the switching times t_i.

With the new independent variable, the evolution equation on the interval $[t_i, t_{i+1}]$ that was given by:

$$\dot{x} = F_{j_i}(x, u, t), \tag{6.6}$$

becomes:

$$x' = (N+1)(x_{n+i+1} - x_{n+i})\hat{F}_{j_i}(x, u, s), \tag{6.7}$$

where $(.)'$ denotes the derivative of $(.)$ with respect to the new independent variable s and

$$\hat{F}(x, u, s) = F(x, u, t(s)).$$

If we denote by $\hat{x}$ the extended state vector:

$$\hat{x} = [x_1, \ldots, x_n, x_{n+1}, \ldots, x_{n+N}]^T,$$

we can define on each interval $\frac{i}{N+1} < s \leq \frac{i+1}{N+1}$:

$$\hat{L}(\hat{x}, u, s) = (N+1)(x_{n+i+1} - x_{n+i})L(x, u, t(s)). \tag{6.8}$$

Finally, we can rewrite the functional (6.4) as:

$$J = \int_0^1 \hat{L}(\hat{x}, u, s)ds. \tag{6.9}$$

and the task is to minimize J in the extended state space. Points s_i at which the system described by the function $\hat{F}$ switches between the discrete states are known. In the optimal solution, $\hat{x}^*$, the last N components will be the optimal switching times $\{t_i\}_{i=1}^N$ for the original problem.

6.3 Illustrative example

We study an example of two fingers with limited workspace rotating a circular object in a horizontal plane. In [76], a similar example is used to study grasp gaits. Our approach is also in line with the conceptual framework proposed in [78] for motion planning for dextrous manipulation. For our task we require that exactly one finger is on the object during any finite time interval (an example of such task would be a driver turning a steering wheel). This effectively partitions the continuous state space into two discrete states, each representing the case when one finger is in contact with the object and the other freely moves in the plane. There is also an intermediate third state when both fingers are in contact with the object, through which the system passes instantaneously at each switch. We assume that the object can rotate around a fixed axis passing through the center of the object and that the fingers can exert arbitrary forces on the object. The motion of each finger is restricted to a cone in the plane of the object. We can easily generalize the example to the case of a multi-fingered hand manipulating a planar object or include frictional constraints.

6.3.1 Mathematical formulation

The position of the object is given by its turning angle φ (Figure 6.3). The center of the object (and thus the pivot point) is at the origin of the global coordinate system and the radius of the object is equal to R. Positions of the two fingers in the plane are expressed in polar coordinates and are (r_1, θ_1) and (r_2, θ_2). The dynamics of the object is given by:

$$\begin{aligned} I\ddot{\varphi} &= -F_{1x}R\sin\theta_1 + F_{1y}R\cos\theta_1 \\ &\quad -F_{2x}R\sin\theta_2 + F_{2y}R\cos\theta_2, \end{aligned} \tag{6.10}$$

where I is the moment of inertia of the object around the axis of rotation while F_1 and F_2 are the forces, expressed in the global coordinate frame, that the fingers 1 and 2 exert on the object.

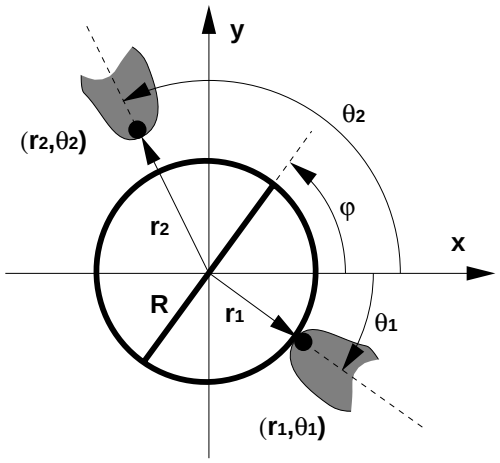


Figure 6.3: Two fingers rotating a circular object in a plane.

The dynamic equations of the two fingers can be expressed in Cartesian coordinates. For simplicity, we assume that the fingers behave like point masses (located at the finger tip). The dynamics of the finger i is thus given by:

$$\begin{aligned} m_i(\ddot{r}_i \cos\theta_i - 2\dot{r}_i\dot{\theta}_i \sin\theta_i - r_i\dot{\theta}_i^{\,2}\cos\theta_i - r_i\ddot{\theta}_i \sin\theta_i) &= -F_{ix} + u_{ix} \\ m_i(\ddot{r}_i \sin\theta_i + 2\dot{r}_i\dot{\theta}_i \cos\theta_i - r_i\dot{\theta}_i^{\,2}\sin\theta_i + r_i\ddot{\theta}_i \cos\theta_i) &= -F_{iy} + u_{iy}, \end{aligned} \tag{6.11}$$

where m_i is the effective mass of the finger i, and u_i is the 2×1 vector of driving forces for finger i. By defining a state vector:

$$x = [\varphi, \dot{\varphi}, r_1, \theta_1, \dot{r}_1, \dot{\theta}_1, r_2, \theta_2, \dot{r}_2, \dot{\theta}_2]^T \tag{6.12}$$

the dynamic equations of the object and the two fingers can be transformed into the state-space form. The vectors u_1 and u_2 are the inputs to the system.

We assume that the workspace of the two fingers is limited. The workspace of each finger is a cone of angle $2\,\alpha$ centered at the origin. The axis of the cone for the first finger corresponds to the half-line $\theta = 0$ while the axis of the cone for the second finger is the half-line $\theta = \pi$. The cones $\mathcal{W}_1$ and $\mathcal{W}_2$ representing the workspace of the fingers 1 and 2, respectively, are thus given by:

$$\begin{aligned} \mathcal{W}_1 &= \{\, \theta \mid -\alpha \le \theta \le \alpha \} \\ \mathcal{W}_2 &= \{\, \theta \mid -\alpha + \pi \le \theta \le \alpha + \pi \}. \end{aligned} \tag{6.13}$$

For our task of rotating the object we require that exactly one finger is on the object during any finite time interval. It is easy to see that this requires partitioning of the state space into three regions:

$$\begin{aligned} D_1 &= \{\, x \mid \theta_1 \in \mathcal{W}_1, \theta_2 \in \mathcal{W}_2, r_1 > R, r_2 = R \} \\ D_2 &= \{\, x \mid \theta_1 \in \mathcal{W}_1, \theta_2 \in \mathcal{W}_2, r_1 = R, r_2 > R \} \\ D_3 &= \{\, x \mid \theta_1 \in \mathcal{W}_1, \theta_2 \in \mathcal{W}_2, r_1 = R, r_2 = R \}. \end{aligned}$$

Region D_1 corresponds to the case when the second finger is in contact with the object while the first finger does not touch the object. Region D_2 describes the opposite situation. In region D_3 both fingers are on the object. Because of the requirement that exactly one finger is on the object during any finite time interval, the system can not stay in D_3. This basically reduces the state space to $D_1 \cup D_2$, with the switch between the two corresponding to D_3.

The dynamic equations are the same in all three regions. Therefore, $f = f_1 = f_2$, where f is the system function obtained when the dynamic equations are rewritten in the state space. However, in the region D_1, the constraint $r_1 = 1$ forces $\dot{r}_1 = 0$. Similarly, the requirement $r_2 > 0$ implies $F_2 = 0$. Analogous equations hold for D_2.

Finally, we have to choose the cost functional for the optimal control problem. The choice depends on the desired properties of the optimal trajectories. For example, continuous force profiles can be obtained by minimizing the rate of change of forces. In this work we chose to minimize a measure of the energy necessary to move the two fingers:

$$J = \int_0^1 L dt = \frac{1}{2} \int_0^1 (u_{1x}^2 + u_{1y}^2 + u_{2x}^2 + u_{2y}^2) dt. \tag{6.14}$$

At $t = 0$, we have the initial conditions:

$$\varphi = 0, \quad \theta_1 = 0, \quad \theta_2 = \pi, \quad r_1 = R, \quad r_2 = R,$$
$$\dot{\varphi} = 0, \quad \dot{\theta}_1 = 0, \quad \dot{\theta}_2 = 0, \quad \dot{r}_1 = 0, \quad \dot{r}_2 = 0.$$

and we assume that the system immediately passes into the region D_1. We require the object to be rotated through $\pi/3$. Both fingers are prescribed to end their motion on the object, but we are not interested where. The final conditions are therefore:

$$\varphi = \tfrac{\pi}{3}, \quad r_1 = R, \quad r_2 = R,$$
$$\dot{\varphi} = 0, \quad \dot{r}_1 = 0, \quad \dot{r}_2 = 0, \quad \dot{\theta}_1 = 0, \quad \dot{\theta}_2 = 0.$$

In order to use the method from Section 6.2, we have to fix the number of transitions between the two regions in the state space. Assume that the system switches only once: we first rotate the object with the second finger and then complete the rotation of the object with the first finger. We now apply the technique outlined in Section 6.2. Let t_1 be the time when the system switches from D_1 to D_2. We extend the state vector x with t_1 and impose the state equation:

$$\dot{t_1} = 0.$$

Next, we define a new independent variable s with the equation:

$$t = \begin{cases} 2t_1 s, & 0 < s \leq 0.5 \\ 2(1 - t_1)s + 2t_1 - 1, & 0.5 < s \leq 1.0. \end{cases} \tag{6.15}$$

The cost function L from Equation (6.14) becomes:

$$\hat{L}(\hat{x}, u, t) = \begin{cases} 2t_1 L, & 0 < s \leq 0.5 \\ 2(1 - t_1)L, & 0.5 < s \leq 1.0. \end{cases} \tag{6.16}$$

Similarly, if $f = f_1 = f_2$ is the system function for the independent variable t, the new system functions become:

$$\begin{aligned} \hat{f}_1 &= 2t_1 f \\ \hat{f}_2 &= 2(1-t_1)f. \end{aligned} \tag{6.17}$$

In each region we also have to satisfy the following constraints:

$$\begin{array}{lll} D_1: & r_1 > R & D_2: \quad r_1 = R \\ & r_2 = R & \quad\quad r_2 > R \\ & \dot{\varphi} = \dot{\theta}_2 & \quad\quad \dot{\varphi} = \dot{\theta}_1 \\ & F_1 = 0 & \quad\quad F_2 = 0 \end{array} \tag{6.18}$$

Analogous expressions can be obtained if the fingers change their roles more than once.

To solve the resulting optimal control problem we can now use one of the methods in Chapter 3. As the first-order method did not converge to a minimum, we used the direct minimization method that was developed in Section 3.3. One of the advantages of this method is that the adjoint variables and the multipliers are not the unknowns for the optimal control problem, they are only updated after the solution of the optimal control problem has been found. Including state, inputs, adjoint variables and the Lagrange multipliers there are 46 unknown functions (for a single switch). The unknowns for the variational problem are only the state and input variables which amounts to 20 unknown functions. The variational problem that has to be solved in each iteration is therefore significantly simpler than for the method from Section 3.2 and the computation time decreases.

6.3.2 Simulation results

For the simulation, we chose $R = 1$ and $m_1 = m_2 = 1$. The task was to rotate the object for 60° counterclockwise. We show motion plans for different choices of the workspace for the two fingers.

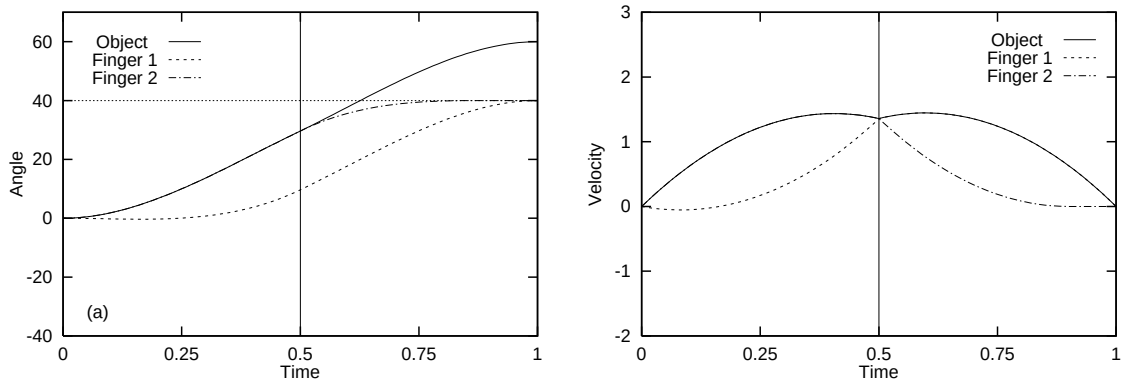


Figure 6.4: (a) Angles φ, θ_1 and $\theta_2 - \pi$; and (b) velocities $\dot{\varphi}$, $\dot{\theta_1}$ and $\dot{\theta_2}$ for the workspace $\alpha = 40°$.

Figure 6.4 shows the trajectories when the workspace is given by $\alpha = 40°$. We subtracted π from θ_2 to compare it with the other two angles. The new independent variable s

is shown on the abscissa. Therefore, the switch between the two regions occurs at $s = 0.5$. For $\alpha = 40°$, the optimal value for the switching time is $t_1 = 0.497s$. In the first half of the task, the finger 2 rotates the object while the finger 1 is not in contact with the object. This can be clearly seen from the velocity plots, where $\dot{\theta}_2 = \dot{\varphi}$, but the finger 1 moves along the surface of the object with lower speed. At $s = 0.5$ ($t = 0.497s$), the object is rotated for 29.6° and the two fingers switch the roles. For $s > 0.5$, the finger 2 moves freely in space and continues its motion until it reaches the limit of the workspace. In the meantime, the finger 1 rotates the object to 60° and also reaches the limit of its workspace.

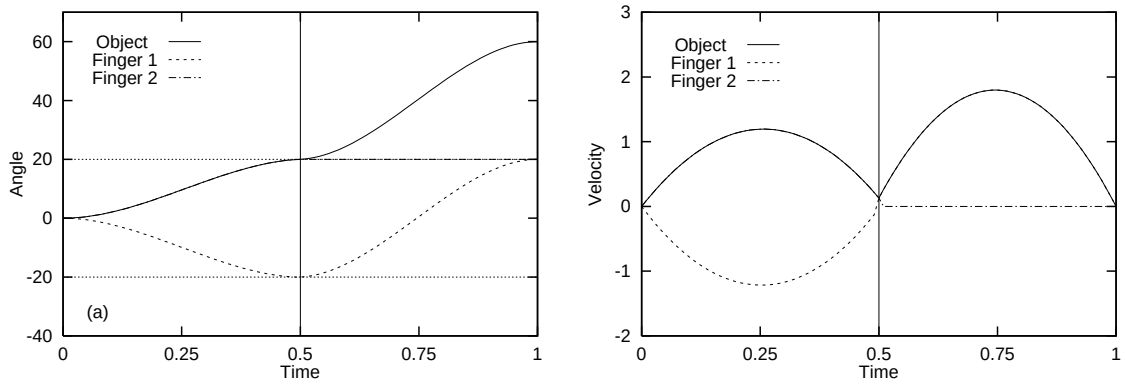


Figure 6.5: (a) Angles φ, θ_1 and $\theta_2 - \pi$; and (b) velocities $\dot{\varphi}$, $\dot{\theta_1}$ and $\dot{\theta_2}$ for the workspace $\alpha = 20°$.

Figure 6.5 shows the optimal trajectories for $\alpha = 20°$. This is a limiting case, since the 60° rotation can be barely achieved with a single switch (the maximal achievable rotation is $3\,\alpha$). For $s < 0.5$, the finger 1 is not in contact with the object and it moves to the lower edge of the workspace. Meanwhile, the finger 2 rotates the object for the entire allowable range $\alpha = 20°$. At $s = 0.5$ ($t = 0.427s$), the finger 1 comes into contact with the object. The finger is at the lower edge of the workspace so that it can subsequently use the whole range $2\,\alpha$ to complete the rotation of the object. The switch between the discrete states D_1 and D_2 occurs at $t = 0.427s$. Also in this example the fingers end their motion at the upper edge of the workspace.

For the trajectories in Figure 6.6, the workspace of the fingers was $\alpha = 15°$. In this case, the rotation of the object for 60° can not be achieved with a single switch; the fingers change their roles twice. The switches occur at $s_1 = 0.33$ and $s_2 = 0.67$, which correspond to $t_1 = 3.22s$ and $t_2 = 7.01s$. In this case the pattern observed in Figure 6.5 is repeated twice: while one finger rotates the object, the other finger moves towards the lower edge of its workspace to have a wider range of motion once it comes into contact with the object. During the first third of the maneuver, finger 2 rotates the object almost for the entire range of the allowable workspace of 15°. Meanwhile, finger 1 moves to the lower edge of the workspace and subsequently it rotates the object for almost 30° in the second third of the task. While finger 1 is rotating the object, finger 2 moves back towards the middle of its allowable workspace so that it can complete the rotation of the object in the third stage. In the third stage finger 1 stays at the upper edge of its workspace.

We also make the following observations:

- The velocities of the contact points on the object and on the finger at the time

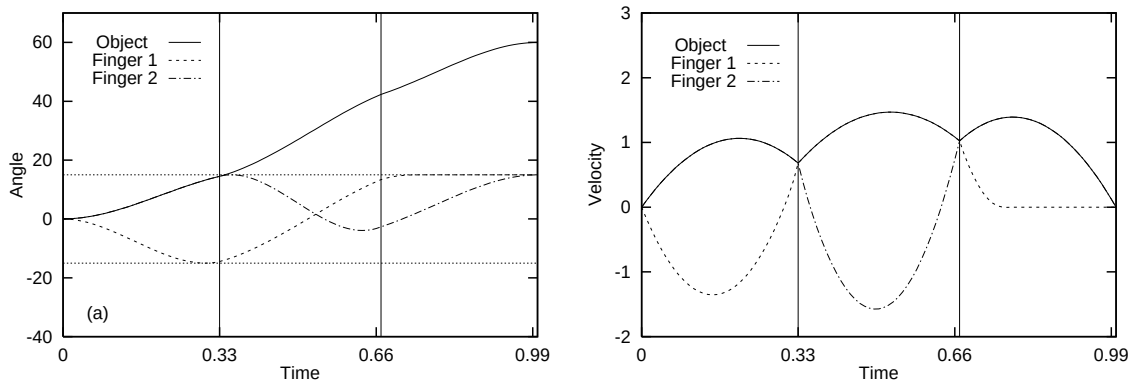


Figure 6.6: (a) Angles φ, θ_1 and $\theta_2 - \pi$; and (b) velocities $\dot{\varphi}$, $\dot{\theta_1}$ and $\dot{\theta_2}$ for the workspace $\alpha = 15°$.

of establishing or breaking the contact are equal so no impact occurs. Hence, the continuous state is continuous across the switches between discrete states.

- The positions of the fingers on the object at the switch are computed as part of the motion plan. In other words, the sequence of discrete states is assumed to be known, but the value of the continuous state at the switches is computed from the optimization.

- The workspace constraints of the two fingers are respected. We could further generalize the task by including frictional constraints.

6.4 Discussion

The existing approaches to motion planning for systems with discrete and continuous state concentrate either on planning the sequence of discrete states to achieve the task or on planning and control of the system within each of the discrete states. Several authors address the problem of planning the task sequence for planar manipulation. Erdmann [36] and Mason [85] study manipulation sequences for orienting polygonal objects in the plane. This work is extended in [82]. Similar is also the work by Goldberg [47] on fixturing for assembly, although the manipulation sequence in this case consists of a single element. Planning of grasp gaits is studied by Leveroni & Salisbury [76], while Donald *et al.* [35] describe planar manipulation with multiple mobile robots. These studies are based on quasi-static analysis and it is not clear whether the results can be used when the quasi-static assumption is violated. Further, the emphasis is on the algorithmic issues of planning the manipulation sequence. The method described in this chapter is an attempt to relax the quasi-static assumption from the above works and replace it with dynamic analysis. In this way, the problem of planning the manipulation sequence can be addressed in the same framework and at the same level as planning the open-loop trajectories for the control of the dynamical system. Although we had to assume that the sequence of the discrete states is planned in advance, our approach guarantees that the discrete sequence is compatible with the dynamics of the system.

We formulated the motion planning problem for systems with continuous and discrete states as an optimal control problem. This optimal control problem has two important features:

1. Dynamic equations describing the system change during the task. It is difficult to address such changes within the framework of optimal control. We showed that the optimal control problem can be simplified if the sequence of discrete states for the task is given.

2. Partition of the state space into regions representing different discrete states can be usually described with a set of equality and inequality constraints. The motion planning method must be therefore capable of dealing with such constraints.

We demonstrated the approach on the example of two fingers rotating a circular object in a plane. In this case the (continuous) state space of the system is partitioned into regions (discrete states) that correspond to different grasp configurations. By choosing the cost function appropriately we can guarantee desired level of smoothness in the velocities, accelerations, and forces, as the system switches between discrete states.

The main limitation of the proposed approach is that the sequence of discrete states must be known *a priori*. However, the value of the continuous state at the switches between discrete states is computed as part of the optimization which is important for example in walking, where this value corresponds to placement of the legs on the ground.

Chapter 7

Task space and the special Euclidean group $SE(3)$

In the next two chapters we turn our attention to kinematic motion planning. We concentrate on planning in the task space, but similar framework could be used to study planning in the joint space. When the trajectories are planned in the task space, we can take advantage of the special structure of this space. In this chapter, we establish the differential geometric framework for study of the task space and investigate some properties of the space that are important for kinematic analysis and kinematic motion planning.

Position and orientation of an end-effector that moves in $\mathbb{R}^3$ can be represented with the rigid-body displacement between an inertial reference frame and a frame attached to the end-effector. The task space can be therefore represented with the set of rigid body displacements in $\mathbb{R}^3$. This set forms a group known as the special Euclidean group in three dimensions, $SE(3)$. The structure of $SE(3)$ is very different from the structure of a Euclidean space $\mathbb{R}^6$, but locally there exists a bijection between the two. This implies that the group $SE(3)$ is a manifold and therefore a Lie group. Because of the non-Euclidean structure of the group $SE(3)$, we study its properties within the framework of differential geometry, which is a natural generalization of differential calculus from the Euclidean spaces.

In motion planning and other applications we need to measure distances on $SE(3)$. The notion of distance on a manifold is provided by choosing a Riemannian metric. Once we choose a metric, the manifold becomes "rigid" since the distances between the points get fixed. A weaker structure on the manifold is obtained by introducing an affine connection. With an affine connection, we can differentiate vector fields and define the covariant derivative, a generalization of differentiation from Euclidean space. The affine connection is used to formally define acceleration and higher derivatives of the velocity. For motion planning, metrics and connections are crucial since they are used to define cost functionals for measuring smoothness of kinematic trajectories.

The tangent space to $SE(3)$ at the identity, denoted by $se(3)$, has the structure of a Lie algebra. The Lie algebra $se(3)$ is isomorphic to the set of twists which is used for velocity analysis in kinematics [4, 68, 79, 137]. An inner product on $se(3)$ can be extended to a Riemannian metric on $SE(3)$. It is well known that there is no inner product on the space of twists that is invariant under the change of the inertial and body-fixed reference frames

that are used to describe motion of a rigid body [79, 80, 92]. But on $se(3)$ there exist two quadratic forms, the Killing form and the Klein form, which are invariant under changes of the reference frames [64, 92]. However neither form defines an inner product: the Killing form is degenerate while the Klein form is indefinite.

After a brief overview of some basic notions from differential geometry, we investigate metrics and connections on $SE(3)$ that are important for kinematic analysis. Chasles's theorem guarantees the existence of a screw motion[1] between any two points on $SE(3)$. Given a metric on $SE(3)$, we can also find a shortest distance curve between any two points with respect to this metric. Such curves are called *geodesics* and can be considered a generalization of straight lines from Euclidean geometry to Riemannian manifolds. A natural question is whether there exists a metric for which geodesics are screw motions. We show that no such Riemannian metric exists. We also identify a two-parameter family of semi-Riemannian metrics, which consists precisely of the metrics having screw motions for geodesics. These metrics can be obtained from the bi-invariant forms on $se(3)$.

In the next section, we study connections that lead to physically meaningful acceleration. We show that there is a unique symmetric connection that yields the acceleration that is used for kinematic analysis. We also identify a family of Riemannian metrics that are compatible with this connection. Any metric in this family is a product of a bi-invariant metric on $SO(3)$ and a Euclidean metric on $\mathbb{R}^3$.

7.1 Kinematics, Lie groups and differential geometry

Consider a rigid body moving in free space. Assume any inertial reference frame {F} fixed in space and a frame {M} fixed to the body at point O' as shown in Figure 7.1. At each instant, the configuration (position and orientation) of the rigid body can be described by a homogeneous transformation matrix corresponding to the displacement from frame {F} to frame {M}. The set of all such matrices is called $SE(3)$, the special Euclidean group of rigid body transformations in three-dimensions:

$$SE(3) = \left\{ \begin{bmatrix} R & d \\ 0 & 1 \end{bmatrix} \mid R \in \mathbb{R}^{3\times 3}, d \in \mathbb{R}^3, R^T R = I, \det(R) = 1 \right\}. \tag{7.1}$$

It is easy to show [92] that $SE(3)$ is a group for the standard matrix multiplication and that it is a manifold. It is therefore a Lie group [121].

On any Lie group the tangent space at the group identity has the structure of a Lie algebra. The Lie algebra of $SE(3)$, denoted by $se(3)$, is given by:

$$se(3) = \left\{ \begin{bmatrix} \Omega & v \\ 0 & 0 \end{bmatrix} \mid \Omega \in \mathbb{R}^{3\times 3}, v \in \mathbb{R}^3, \Omega^T = -\Omega \right\}. \tag{7.2}$$

A 3×3 skew-symmetric matrix Ω can be uniquely identified with a vector $\omega \in \mathbb{R}^3$ so that for an arbitrary vector $x \in \mathbb{R}^3$, $\Omega x = \omega \times x$, where $\times$ is the vector cross product operation in $\mathbb{R}^3$. Each element $S \in se(3)$ can be thus identified with a vector pair $\{\omega, v\}$.

[1] A screw motion is a rigid body motion in which the rigid body rotates about an axis while concurrently translating along that axis. It will be formally defined later in the chapter.

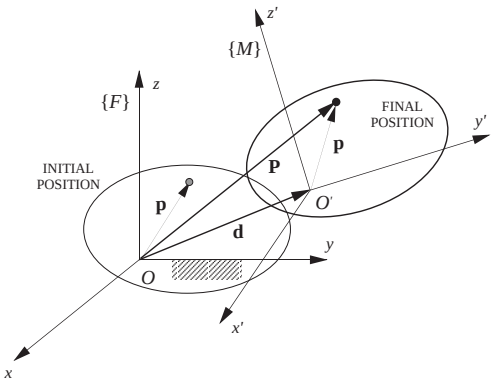


Figure 7.1: The inertial (fixed) frame and the moving frame attached to the rigid body

Given a curve $A(t) : [-a, a] \to SE(3)$, an element $S(t)$ of the Lie algebra $se(3)$ can be associated to the tangent vector $\dot{A}(t)$ at an arbitrary point t by:

$$S(t) = A^{-1}(t)\dot{A}(t) = \begin{bmatrix} R^T\dot{R} & R^T\dot{d} \\ 0 & 0 \end{bmatrix}. \tag{7.3}$$

A curve on $SE(3)$ physically represents a motion of the rigid body. If $\{\omega(t), v(t)\}$ is the vector pair corresponding to $S(t)$, then ω physically corresponds to the angular velocity of the rigid body while v is the linear velocity of the origin O' of the frame {M}, both expressed in the frame {M}. In kinematics, elements of this form are called twists [87] and $se(3)$ thus corresponds to the space of twists. It is easy to check that the twist $S(t)$ computed from Equation (7.3) does not depend on the choice of the inertial frame {F}. For this reason, $S(t)$ is called the left invariant representation of the tangent vector $\dot{A}$. Alternatively, the tangent vector $\dot{A}$ can be identified with a right invariant twist (invariant with respect to the choice of the body-fixed frame {M}).

Since $se(3)$ is a vector space, any element can be expressed as a 6×1 vector of components corresponding to a chosen basis. The standard basis for $se(3)$ is:

$$\begin{aligned} L_1 &= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} & L_2 &= \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} & L_3 &= \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \\ L_4 &= \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} & L_5 &= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} & L_6 &= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \end{aligned} \tag{7.4}$$

The twists L_1, L_2 and L_3 represent instantaneous rotations about and L_4, L_5 and L_6 instantaneous translations along the Cartesian axes x, y and z, respectively. The components of a twist $S \in se(3)$ in this basis are given precisely by the velocity vector pair, $\{\omega, v\}$.

The Lie bracket of two elements $S_1, S_2 \in se(3)$ is defined by:

$$[S_1, S_2] = S_1S_2 - S_2S_1.$$

It can easily be verified that if $\{\omega_1, v_1\}$ and $\{\omega_2, v_2\}$ are vector pairs corresponding to the twists S_1 and S_2, the vector pair $\{\omega, v\}$ corresponding to their Lie bracket $[S_1, S_2]$ is given by

$$\{\omega, v\} = \{\omega_1 \times \omega_2, \omega_1 \times v_2 + v_1 \times \omega_2\}. \tag{7.5}$$

In kinematics, this product operation is called the motor product of the two twists.

The Lie bracket of two elements of a Lie algebra is an element of the Lie algebra and can be expressed as a linear combination of the basis vectors. The coefficients C_{ij}^k corresponding to the Lie brackets of the basis vectors are called *structure constants* of the Lie algebra [121]:

$$[L_i, L_j] = \sum_k C_{ij}^k L_k. \tag{7.6}$$

The expressions (7.6) for $se(3)$, can be directly computed from Equation (7.5) and are listed in Appendix E.

7.1.1 Left invariant vector fields

A *differentiable vector field* is a smooth assignment of a tangent vector to each element of the manifold. At each point, a vector field defines a unique *integral curve* to which it is tangent [34]. Formally, a vector field X is a (derivation) operator which, given a differentiable function f, returns its derivative (another function) along the integral curves of X. In other words, if $\gamma(t)$ is a curve tangent to a vector field X at point $p = \gamma(t_0)$, then:

$$X(f)|_p = \left.\frac{df(\gamma(t))}{dt}\right|_{t_0}. \tag{7.7}$$

We will be particularly interested in *left invariant vector fields* on $SE(3)$. From any twist, $S \in se(3)$, we can generate a differentiable vector field $\hat{S}$ by assigning to all $A \in SE(3)$ a vector, $\hat{S}(A)$, in the tangent space at that point by the formula:

$$\hat{S}(A) = AS, \tag{7.8}$$

The vector field generated by Equation (7.8) is called a left invariant vector field and we use the notation $\hat{S}$ to indicate that the vector field was obtained by left translating the Lie algebra element S. Right invariant vector fields can be defined analogously. In general, a vector field need not be left or right invariant! By construction, the space of left invariant vector fields is isomorphic to the Lie algebra $se(3)$ and vector fields $\hat{L}_1, \hat{L}_2, \ldots, \hat{L}_6$ are its basis. We also have (see [34]):

$$[\hat{L}_i, \hat{L}_j] = \widehat{[L_i, L_j]} = \sum_k C_{ij}^k \hat{L}_k. \tag{7.9}$$

Since the vectors $L_1, L_2, \ldots, L_6$ are a basis for the Lie algebra $se(3)$, the vectors $\hat{L}_1(A), \ldots, \hat{L}_6(A)$ form a basis of the tangent space at any point $A \in SE(3)$. Therefore, any vector field X can be expressed as

$$X = \sum_{i=1}^{6} X^i \hat{L}_i, \tag{7.10}$$

where the coefficients X^i are real-valued functions. If the coefficients are constant, then X is left invariant. Equation (7.3) shows that:

$$\dot{A} = A\,S = A\left(\sum_{i=1}^{6} S^i L_i\right) = \sum_{i=1}^{6} S^i (A\,L_i) = \sum_{i=1}^{6} S^i \hat{L}_i(A). \tag{7.11}$$

We conclude that if the velocity of the rigid body, $\dot{A}$, is expressed in the basis $\hat{L}_1, \ldots, \hat{L}_6$, the components with respect to this basis are equal to the components of the instantaneous twist:

$$\omega = [S^1, S^2, S^3]^T,$$
$$v = [S^4, S^5, S^6]^T.$$

We will therefore refer to the elements $\hat{L}_1, \ldots, \hat{L}_6$ of the basis for the left invariant vector fields as *basis twists*. Equation (7.3) also implies that ω is the vector corresponding to the skew-symmetric matrix $\Omega = R^T\dot{R}$, while $v = R^T\dot{d}$.

7.1.2 Exponential map and local coordinates

A curve, $A(t)$, on $SE(3)$ describes a motion of the rigid body. If $V = \frac{dA}{dt}$ is the vector field tangent to $A(t)$, the vector pair $\{\omega, v\}$ associated with V corresponds to the instantaneous twist (screw axis) for the motion. In general, the twist $\{\omega, v\}$ changes with time. Motions for which the twist $\{\omega, v\}$ is constant are known in kinematics as *screw motions*. In this case the twist $\{\omega, v\}$ is called *the screw axis* of the motion. If the vector pair $\{\omega, v\}$ is interpreted as Plücker coordinates of a line in space, it is not difficult to see that the screw motion physically corresponds to rotation around this line with a constant angular velocity and concurrent translation along the line with a constant translational velocity.

Let the twist $S \in se(3)$ be represented by a vector pair $\{\omega, v\}$ and let $A(t)$ be a screw motion with the screw axis $\{\omega, v\}$ such that $A(0) = I$. We define *the exponential map* $\exp : se(3) \to SE(3)$ by:

$$\exp(tS) = A(t), \tag{7.12}$$

Using Equation (7.3) we can show that the exponential map agrees with the usual exponentiation of the matrices in $\mathbb{R}^{4\times4}$:

$$\exp(tS) = \sum_{k=0}^{\infty} \frac{t^k \mathrm{S}^k}{k!}, \tag{7.13}$$

where S denotes the matrix representation of the twist S. The sum of this series can be computed explicitly and the resulting expression, when restricted to $SO(3)$, is known as Rodrigues' formula. The formula for the sum in $SE(3)$ is derived in [92].

Given a twist $T(t)$, the motion of the rigid body is given by:

$$\frac{dA(t)}{dt} = A(t)T(t). \tag{7.14}$$

Since for all t, $T(t)$ belongs to the Lie algebra $se(3)$, it can be expressed as a linear combination (with varying coefficients) of the basis vectors. According to Wei and Norman

[150], the solution of this differential equation can be written as the product of exponentials in the form:

$$A(t) = \prod_{i=1}^{6} \exp(\xi^i(t) L_i), \tag{7.15}$$

where $\xi^i(t)$ are analytic functions. The coefficients ξ^i only depend on A and can be therefore taken as a set of local coordinates. This representation of $SE(3)$ is valid in some neighborhood of the identity element and is not global [19]. The coordinates obtained from (7.15) are called *the canonical coordinates of the second kind.*

Although the order of taking the product in (7.15) is arbitrary, once chosen we must adhere to it to get consistent values for the coordinates since, in general:

$$\exp(\xi^i L_i) \exp(\xi^j L_j) \neq \exp(\xi^j L_j) \exp(\xi^i L_i). \tag{7.16}$$

We choose the following ordering:

$$A(t) = \exp(\xi^6 L_6) \exp(\xi^5 L_5) \cdots \exp(\xi^1 L_1). \tag{7.17}$$

An alternative way of defining a parameterization for $SE(3)$ is to use the so called canonical coordinates of the first kind [19, 51, 92]:

$$A(\xi) = \exp\left(\xi^1 L_1 + \xi^2 L_2 + \xi^3 L_3 + \xi^4 L_4 + \xi^5 L_5 + \xi^6 L_6\right). \tag{7.18}$$

Also this parameterization is only valid in some neighborhood of the identity.

7.1.3 Riemannian metrics on Lie groups

If a smoothly varying, positive definite, bilinear, symmetric form $< .,. >$ is defined on the tangent space at each point on the manifold, such form is called a *Riemannian metric* and the manifold is *Riemannian* [34]. If the form is non-degenerate but indefinite, the metric is called *semi-Riemannian* [9]. On a n-dimensional manifold, the metric is locally characterized by a $n \times n$ matrix of $\mathcal{C}^\infty$ functions $g_{ij} =< X_i, X_j >$ where X_i are basis vector fields[2]. If the basis vector fields can be defined globally, then the matrix $[g_{ij}]$ completely defines the metric.

On $SE(3)$ (on any Lie group), an inner product on the Lie algebra can be extended to a Riemannian metric over the manifold using left (or right) translation. To see this, consider the inner product of two elements $S_1, S_2 \in se(3)$ defined by

$$< S_1, S_2 >|_I = s_1^T W s_2, \tag{7.19}$$

where s_1 and s_2 are the 6×1 vectors of components of S_1 and S_2 with respect to some basis and W is a positive definite matrix. If V_1 and V_2 are tangent vectors at an arbitrary group element $A \in SE(3)$, the inner product $< V_1, V_2 >|_A$ in the tangent space $T_A SE(3)$ can be defined by:

$$< V_1, V_2 >|_A = < A^{-1} V_1, A^{-1} V_2 >\Big|_I . \tag{7.20}$$

The metric obtained in such a way is said to be left invariant [34] since left translation by any element A is an isometry.

[2]The basis vector fields need not be the coordinate basis; we only require that they are smooth.

7.1.4 Affine connection and covariant derivative

Motion of a rigid body can be represented by a curve on $SE(3)$. The velocity at an arbitrary point is the tangent vector to the curve at that point. In order to obtain the acceleration, or to engage in a dynamic analysis, we need to be able to differentiate a vector field along the curve. At each point $A \in SE(3)$, the value of a vector field belongs to the tangent space $T_A\,SE(3)$ and to differentiate a vector field along a curve, we must be able to subtract vectors from tangent spaces at different points on the curve. But tangent spaces at different points are not related. We thus have to specify how to transport a vector along the curve from one tangent space to another. This process is called *parallel transport* and is formalized with the affine connection [122]. Given any curve $A(t)$, a parameter value t_0 and a vector V in $T_{A(t_0)}\,SE(3)$, the tangent space at point $A(t_0)$, the affine connection assigns to each other parameter value t a vector $V' \in T_{A(t)}\,SE(3)$. By definition, V' is the parallel transport of V along the curve $A(t)$. Vectors V' and V are also said to be *parallel* along $A(t)$.

A derivative of a vector field along a curve $A(t)$ is defined through the parallel transport. Let X be a vector field defined along $A(t)$, and let $X(t)$ stand for $X(A(t))$. Denote by $X^{t_0}(t)$ the parallel transport of the vector $X(t)$ to the point $A(t_0)$. The *covariant derivative* of X along $A(t)$ is:

$$\left.\frac{DX}{dt}\right|_{t_0} = \lim_{t \to t_0} \frac{X^{t_0}(t) - X(t_0)}{t}. \tag{7.21}$$

By taking covariant derivatives along integral curves of a vector field Y, we obtain a covariant derivative of the vector field X with respect to the vector field Y. This derivative is also denoted by $\nabla_Y X$:

$$\nabla_Y X|_{A_0} = \left.\frac{DX}{dt}\right|_{t_0}, \tag{7.22}$$

where $\frac{DX}{dt}$ is taken along the integral curve of Y passing through A_0 at $t = t_0$.

The notions of the covariant derivative and parallelism are equivalent. In this section we defined the covariant derivative through parallel transport. However, given a covariant derivative, we can always define that a field X is parallel along a curve $A(t)$ if:

$$\nabla_{\frac{dA}{dt}} X = 0.$$

The covariant derivative of a vector field is another vector field so it can be expressed as a linear combination of the basis vector fields. The coefficients Γ^k_{ji} of the covariant derivative of a basis vector field along another basis vector field,

$$\nabla_{\hat{L}_i} \hat{L}_j = \sum_k \Gamma^k_{ji} \hat{L}_k, \tag{7.23}$$

are called *Christoffel symbols*[3]. Note the reversed order of the indices i and j.

The velocity, $V(t)$, of the rigid body describing the motion $A(t)$ is given by the tangent vector field along the curve:

$$V(t) = \frac{dA(t)}{dt}.$$

[3]In the literature, different definitions for the Christoffel symbols can be found. Some texts (e.g. [34]) reserve the term for the case of the coordinate basis vectors. We follow the more general definition from [122] in which the basis vectors can be arbitrary.

The acceleration, $\mathcal{A}(t)$, is the covariant derivative of the velocity along the curve

$$\mathcal{A} = \frac{\mathrm{D}}{dt}\left(\frac{dA}{dt}\right) = \nabla_V V. \tag{7.24}$$

Note that the acceleration depends on the choice of the connection. We can also define jerk, $\mathcal{J}$, as the covariant derivative of the acceleration:

$$\mathcal{J} = \frac{\mathrm{D}}{dt}\mathcal{A}(t) = \nabla_V \nabla_V V. \tag{7.25}$$

Given a Riemannian manifold, there exists a unique connection [34] which is compatible with the metric:

$$X < Y, Z > = < \nabla_X Y, Z > + < Y, \nabla_X Z > \tag{7.26}$$

and symmetric:

$$\nabla_X Y - \nabla_Y X = [X, Y]. \tag{7.27}$$

This connection is called the *Levi-Civita* or *Riemannian connection*. It can be shown [34] that the compatibility condition (7.26) is equivalent to saying that the parallel transport preserves the inner product. In other words, if $A(t)$ is a curve and X and Y are two vector fields obtained by parallel transporting two vectors X_0 and Y_0 from $T_{A_0}SE(3)$ along $A(t)$, then $< X, Y >|_{A(t)} = < X_0, Y_0 >|_{A_0} = \text{const}$.

7.1.5 Geodesics

Given a Riemannian metric $< ., . >$ on $SE(3)$ we can define the length, $L(A)$, of a smooth curve $A : [a, b] \to SE(3)$ by:

$$L(A) = \int_a^b < \frac{dA}{dt}, \frac{dA}{dt} >^{\frac{1}{2}} dt \tag{7.28}$$

Among all the curves connecting two points, we are usually interested in the curve of minimal length. It is not difficult to see [34] that a curve of minimal length also minimizes the *energy functional*:

$$E(A) = \int_a^b < \frac{dA}{dt}, \frac{dA}{dt} > dt. \tag{7.29}$$

If a curve minimizes a functional, it must also be a critical point. Critical points of the energy functional E satisfy the following equation [34]:

$$\nabla_{\frac{dA}{dt}} \frac{dA}{dt} = 0. \tag{7.30}$$

where ∇ is the Riemannian connection, and are called *geodesics*. From what we said about the covariant derivative and parallel transport, it follows that a geodesic is a curve $A(t)$ for which the tangent vector field $\frac{dA}{dt}$ is parallel: from a value at a point we can obtain its value at any other point by simply parallel transporting it along the curve $A(t)$. On the other hand, According to Equation (7.24), the expression $\nabla_{\frac{dA}{dt}} \frac{dA}{dt}$ is the acceleration of motion described by $A(t)$. Motion along a geodesic therefore produces zero acceleration.

7.2 Metrics and screw motions

One of the fundamental results in rigid body kinematics [87] was proved by Chasles at the beginning of the 19th century:

Theorem 7.1 (Chasles) *Any rigid body displacement can be realized by a rotation about an axis combined with a translation parallel to that axis.*

Note that a displacement must be understood as an element of $SE(3)$ while a motion is a curve on $SE(3)$. If the rotation from the Chasles's theorem is performed at constant angular velocity and the translation at constant translational velocity, the motion leading to the displacement clearly becomes a screw motion. Chasles's theorem therefore says that any rigid body displacement can be realized with a screw motion.

Another family of curves of particular interest on $SE(3)$ are the *one-parameter subgroups*. A curve $A(t)$ is a one-parameter subgroup, if $A(t_1 + t_2) = A(t_1)A(t_2)$. The one-parameter subgroups on $SE(3)$ are given by [34]:

$$A(t) = \exp(t\,S) \tag{7.31}$$

where S is an element of $se(3)$. From our discussion in Section 7.1.2 it is clear that the one parameter subgroups are exactly the screw motions which start at the identity. In the Chasles's theorem, we can obviously assume that the body fixed frame {M} is initially aligned with the inertial frame {F}. Therefore, in the language of differential geometry, the theorem can be restated as follows:

Theorem 7.2 (Chasles restated) *For every element in $SE(3)$ there is a one-parameter subgroup (screw motion through the identity) to which that element belongs.*

Except for the identity, the screw axis for every element is unique, but there are infinitely many screw motions along that axis which contain the element, each characterized by the number of rotations along the screw axis. The following corollary immediately follows from the theorem:

Corollary 7.3 *If A_1 and A_2 are two distinct elements of $SE(3)$, then:*

(1) There exists a one-parameter subgroup, $\gamma_L(t) = \exp(t\,S_L)$, which when left translated by A_1 contains A_2:

$$A_L(t) = A_1\ \exp(t\,S_L), \quad A_2 = A_L(1) = A_1 \exp(S_L).$$

(2) There exists a one-parameter subgroup, $\gamma_R(t) = \exp(t\,S_R)$, which when right translated by A_1 contains A_2:

$$A_R(t) = \exp(t\,S_R)A_1, \quad A_2 = A_R(1) = \exp(S_R)A_1.$$

(3) For each S_L in (1) we can find the corresponding S_R in (2) by:

$$S_R = A_1 S_L A_1^{-1},$$

and in this case

$$A_L(t) = A_R(t).$$

Proof: Statement (1) (respectively (2)) follows from Theorem 7.2 if we left (right) translate the one-parameter subgroup to which $A^{-1}A_2$ (A_2A^{-1}) belongs. To see (3), note that:

$$A_1 \exp(t\, S_L) = A_1 \sum_{k=0}^{\infty} \frac{t^k\, S_L^k}{k!} = \sum_{k=0}^{\infty} \frac{t^k (A_1\, S_L A_1^{-1})^k}{k!}\, A_1 .$$

□

Park and Brockett [106] proposed a left invariant Riemannian metric on $SE(3)$ given by:

$$W = \begin{bmatrix} \alpha I & 0 \\ 0 & \beta I \end{bmatrix} \tag{7.32}$$

where α and β are positive scalars.

Park [105] derived the geodesics for the metric (7.32) and showed that they are products of the geodesics for the bi-invariant metric on $SO(3)$ and geodesics in the Euclidean space $\mathbb{R}^3$. Geodesics for the bi-invariant metric on $SO(3)$ are the restrictions of the screw motions to $SO(3)$, while straight lines parameterized proportionally to the line length are the geodesics for the Euclidean space $\mathbb{R}^3$. A geodesic between two elements

$$\begin{bmatrix} R_1 & d_1 \\ 0 & 1 \end{bmatrix} \qquad \text{and} \qquad \begin{bmatrix} R_2 & d_2 \\ 0 & 1 \end{bmatrix}$$

thus physically corresponds to a translation of the origin O' of the body fixed frame {M} with a constant translational velocity along the line connecting the points described with the position vectors d_1 and d_2, and concurrent rotation of the frame {M} with constant angular velocity about an axis passing through the origin O' of the frame {M} which translates together with the point O'. It is clear that such motion is in general not a screw motion since the axis around which the body rotates is not fixed in space. However, if the axis of rotation is collinear with the line between d_1 and d_2 along which the body translates, the rotational axis does not change as the body moves and the geodesic is therefore a screw motion. It follows that a screw motion is a geodesic if and only if it is obtained by left translation of a one-parameter subgroup for which the screw axis passes through the origin O of the frame {F}.

7.2.1 Screw motions as geodesics

Given that any two elements of $SE(3)$ can be connected with a screw motion, and given that there exists a left invariant metric whose geodesics include certain screw motions, it is natural to ask whether there are Riemannian metrics for which every geodesic is a screw motion.

Before we proceed, we turn our attention back to Corollary 7.3. The corollary says that any screw motion can be obtained in two ways: either by a left or by a right translation of a (in general different) one-parameter subgroup. Now suppose that the screw motions are geodesics. Corollary 7.3 implies that a left or a right translation of a geodesic produces another geodesic. We might therefore wrongly conclude that any metric for which the screw motions are geodesics must be invariant under left and right translations and therefore bi-invariant. Such reasoning is false since a map which preserves geodesics does not necessarily preserve the metric (is not an isometry)! This is clear if we consider affine transformations

in Euclidean space: They map lines into lines, but in general they do not preserve lengths of vectors. Therefore, we can not limit our search only to the left or right invariant metrics.

We now derive the family of metrics which have screw motions for geodesics. As we saw in Section 7.1.2, the twist associated with a screw motion $\gamma(t)$ is constant. The tangent vector field $V = \frac{d\gamma}{dt}$ is therefore a left invariant vector field and it has constant components with respect to the chosen basis vector fields: $V = V^i \hat{L}_i$. If γ solves Equation (7.30), we have:

$$0 = \nabla_V V = \sum_j \frac{dV^j}{dt}\hat{L}_j + \sum_{i,j} V^i V^j \nabla_{\hat{L}_i}\hat{L}_j = \sum_{i,j} V^i V^j \nabla_{\hat{L}_i}\hat{L}_j.$$

The above equation is satisfied for any screw motion (arbitrary choice of the components V^i) if and only if

$$\nabla_{\hat{L}_i}\hat{L}_j + \nabla_{\hat{L}_j}\hat{L}_i = 0.$$

Since ∇ is a metrical connection, it is symmetric (Equation 7.27):

$$\nabla_{\hat{L}_i}\hat{L}_j - \nabla_{\hat{L}_j}\hat{L}_i = [\hat{L}_i, \hat{L}_j].$$

It immediately follows that:

$$\nabla_{\hat{L}_i}\hat{L}_j = \frac{1}{2}[\hat{L}_i, \hat{L}_j]. \tag{7.33}$$

Further, ∇ must be compatible with the metric (Equation 7.26), so we have:

$$\hat{L}_k < \hat{L}_i, \hat{L}_j > = < \nabla_{\hat{L}_k}\hat{L}_i, \hat{L}_j > + < \hat{L}_i, \nabla_{\hat{L}_k}\hat{L}_j > . \tag{7.34}$$

Letting $g_{ij} = < \hat{L}_i, \hat{L}_j >$, the last equation implies:

$$\hat{L}_k(g_{ij}) = \frac{1}{2}\left(< [\hat{L}_k, \hat{L}_i], \hat{L}_j > + < \hat{L}_i, [\hat{L}_k, \hat{L}_j] > \right). \tag{7.35}$$

By expressing the Lie brackets from Equation (7.6), we finally obtain:

$$\hat{L}_k(g_{ij}) = \frac{1}{2}\sum_l (C^l_{ki} g_{lj} + C^l_{kj} g_{li}). \tag{7.36}$$

Note that the coefficients C^k_{ij} are constant over the manifold (Equation 7.9). The above derivation is summarized in the following proposition:

Proposition 7.4 *Screw motions satisfy the geodesic equation (7.30) for a Riemannian metric given by the matrix of coefficients $G = [g_{ij}]$ if and only if the coefficients g_{ij} satisfy Equation (7.36).*

The metric coefficients g_{ij} are symmetric by definition. Since $SE(3)$ is a six-dimensional manifold, there are 21 independent coefficients $\{g_{ij} \mid 1 \leq i \leq j \leq 6\}$. Further, there are 6 basis vector fields hence Equation (7.36) expands to a total of 126 equations. Each vector field represents a derivation implying that these are partial differential equations. The complete set of equations is given in Appendix C.

We need the following lemma to derive the solution for the system of equations given by (7.36):

Lemma 7.5 *Given a set of partial differential equations*

$$X(f) = g_x \tag{7.37}$$
$$Y(f) = g_y \tag{7.38}$$
$$Z(f) = g_z \tag{7.39}$$

where X, Y, and Z are vector fields such that $Z = [X, Y]$, f is twice differentiable, and g_x, g_y and g_z are differentiable (real valued) functions, the solution exists only if

$$X(g_y) - Y(g_x) = g_z. \tag{7.40}$$

Proof: By applying X on Equation (7.38), Y on Equation (7.37) and subtracting the two resulting equations, we get:

$$X\,Y(f) - Y\,X(f) = X(g_y) - Y(g_x). \tag{7.41}$$

But the left-hand side is by definition $[X, Y](f)$, which is by assumption equal to $Z(f)$. Equation (7.40) then follows from Equation (7.39). □

We next state the first key theorem of this chapter:

Theorem 7.6 *A matrix of coefficients $G = [g_{ij}]$ satisfies the system of partial differential equations (7.36) if and only if it has the form*

$$G = \begin{bmatrix} \alpha\, I_{3\times 3} & \beta\, I_{3\times 3} \\ \beta\, I_{3\times 3} & 0_{3\times 3} \end{bmatrix}, \tag{7.42}$$

where α and β are constants.

Proof: To find the metric coefficients, we start with the following subset of (C.2):

$$\hat{L}_1(g_{11}) = 0 \quad \hat{L}_2(g_{11}) = -g_{13} \quad \hat{L}_3(g_{11}) = g_{12}. \tag{7.43}$$

First, observe that $[\hat{L}_1, \hat{L}_2] = \hat{L}_3$ (see Appendix E). By application of Lemma 7.5, the following equation must hold:

$$-\hat{L}_1(g_{13}) = g_{12}. \tag{7.44}$$

But from (C.2), we have:

$$\hat{L}_1(g_{13}) = -\frac{1}{2} g_{12}.$$

Therefore, Equation (7.44) becomes:

$$\frac{1}{2} g_{12} = g_{12}.$$

Obviously, this implies that $g_{12} = 0$. We next observe that $g_{12} = 0$ implies $\hat{L}_i(g_{12}) = 0$, $1 \le i \le 6$. From the system (C.2) we obtain:

$$\begin{array}{lll} g_{13} = 0 & g_{23} = 0 & g_{11} = g_{22} \\ g_{16} = 0 & g_{26} = 0 & g_{14} = g_{25}. \end{array} \tag{7.45}$$

From these equations and (C.2) we further obtain:

$$\begin{array}{lll} g_{15}=0 & g_{24}=0 & g_{11}=g_{33} \\ g_{34}=0 & g_{35}=0 & g_{14}=g_{36} \\ g_{44}=0 & g_{45}=0 & g_{46}=0 \\ g_{55}=0 & g_{56}=0 & g_{66}=0. \end{array} \tag{7.46}$$

Next observation is that $\hat{L}_i(g_{11}) = 0, \;\; 1 \leq i \leq 6$. This, together with Equations (7.45) and (7.46) implies:

$$g_{11} = g_{22} = g_{33} = \alpha,$$

where α is an arbitrary constant. Similarly, we obtain

$$g_{14} = g_{25} = g_{36} = \beta,$$

for an arbitrary constant β. In this way we have obtained all 21 independent values of G. The reader can easily check that the system of equations (C.2) is satisfied by the above values.

□

Corollary 7.7 *There is no Riemannian metric whose geodesics are screw motions.*

Proof: It is easy to check that a matrix of the form

$$G = \begin{bmatrix} \alpha\, I_{3\times 3} & \beta\, I_{3\times 3} \\ \beta\, I_{3\times 3} & 0_{3\times 3} \end{bmatrix},$$

has two distinct real eigenvalues

$$\lambda_1 = \frac{1}{2}(\alpha + \sqrt{\alpha^2 + 4\beta^2}) \qquad \lambda_2 = \frac{1}{2}(\alpha - \sqrt{\alpha^2 + 4\beta^2}),$$

which both have multiplicity 3. For any choice of α and β, the product of the eigenvalues is $\lambda_1\lambda_2 = -\beta^2 \leq 0$. Therefore, G is not positive definite as required for a Riemannian metric.

□

7.2.2 Invariance of the family of semi-Riemannian metrics (7.42)

Metrics of the form (7.42) form a two-parameter family of semi-Riemannian metrics and can be studied in a similar way as Riemannian metrics. In particular, we can investigate their invariance properties. By definition, a metric is left invariant if for any $A, B \in SE(3)$ and for any vector fields X and Y:

$$< X(B), Y(B) >|_B = < AX(B), AY(B) >|_{AB}\,, \tag{7.47}$$

and it is right invariant if:

$$< X(B), Y(B) >|_B = < X(B)A, Y(B)A >|_{BA}\,. \tag{7.48}$$

Lemma 7.8 *If S_1 and S_2 are two elements of $se(3)$ and a metric of the form (7.42) is defined on $SE(3)$, then for any $A \in SE(3)$*

$$< S_1, S_2 >|_I = < \mathrm{Ad}_A(S_1), \mathrm{Ad}_A(S_2) >|_I . \tag{7.49}$$

(The map $\mathrm{Ad} : se(3) \to se(3)$ is called *the adjoint map* and if S is represented by a matrix, the map is defined by $\mathrm{Ad}_A(S) = A\,S\,A^{-1}$.)

Proof: Let $S_1 = \{\omega_1, v_1\}$ and $S_2 = \{\omega_2, v_2\}$. By a straightforward algebraic calculation it can be shown that for $S = \{\omega, v\} \in se(3)$ and $A \in SE(3)$, where

$$A = \begin{bmatrix} R & d \\ 0 & 1 \end{bmatrix}$$

the value of $\mathrm{Ad}_A(S)$ is given by $\mathrm{Ad}_A(S) = \{R\,\omega, R\,v - (R\omega) \times d\}$ where $\times$ is the usual vector cross product. Therefore, we have:

$$\begin{aligned}
&< \mathrm{Ad}_A(S_1), \mathrm{Ad}_A(S_2) >|_I \\
&\quad = < \{R\,\omega_1, R\,v_1 - (R\omega_1) \times d\}, \{R\,\omega_2, R\,v_2 - (R\omega_2) \times d\} >|_I \\
&\quad = \alpha\,(R\,\omega_1)^T (R\,\omega_2) + \beta\,(R\,\omega_1)^T (R\,v_2 - (R\omega_2) \times d) + \beta\,(R\,\omega_2)^T (R\,v_1 - (R\omega_1) \times d) \\
&\quad = \alpha\,\omega_1^T \omega_2 + \beta\,(\omega_1^T v_2 + \omega_2^T v_1) - \beta((R\,\omega_1)^T ((R\omega_2) \times d) + (R\,\omega_2)^T ((R\omega_1) \times d)) \\
&\quad = \alpha\,\omega_1^T \omega_2 + \beta\,(\omega_1^T v_2 + \omega_2^T v_1) = < \{\omega_1, v_1\}, \{\omega_2, v_2\} >|_I = < S_1, S_2 >|_I
\end{aligned}$$

□

Proposition 7.9 *Any left invariant metric G that satisfies Equation (7.49) is bi-invariant (both, left and right invariant).*

Proof: We have to prove that G is right invariant. Take two vector fields X and Y. Since the metric G is left invariant, we have:

$$\begin{aligned}
&< X(B)A, Y(B)A >|_{B\,A} \\
&\quad = < (B\,A)^{-1} X(B)A, (B\,A)^{-1} Y(B)A >\Big|_I \\
&\quad = < A^{-1} B^{-1} X(B)A, A^{-1} B^{-1} Y(B)A >\Big|_I .
\end{aligned}$$

By Equation (7.49),

$$< A^{-1} B^{-1} X(B)A, A^{-1} B^{-1} Y(B)A >\Big|_I \tag{7.50}$$

$$= < B^{-1} X(B), B^{-1} Y(B) >\Big|_I . \tag{7.51}$$

But because of the left invariance of G, the last expression is:

$$< B^{-1} X(B), B^{-1} Y(B) >\Big|_I = < X(B), Y(B) >|_B ,$$

as required.

□

Corollary 7.10 *Any metric G of the form (7.42) is bi-invariant.*

Proof: It is obvious that a metric G of the form (7.42) is left invariant, since it is constant for the basis of the left invariant vector fields $\hat{L}_i$. By Lemma 7.8 and Proposition 7.9, G is bi-invariant.

□

7.2.3 Geodesics of the family of semi-Riemannian metrics (7.42)

Analogous to the Riemannian case, we could define the length of a curve $A(t)$ between two points $A(t_1)$ and $A(t_2)$ on $SE(3)$ by:

$$L(A; t_1, t_2) = \int_{t_1}^{t_2} < \frac{dA}{dt}, \frac{dA}{dt} >^{\frac{1}{2}} dt. \tag{7.52}$$

But G is not positive definite, so the length of a curve would be in general a complex number. Therefore, it is more useful to define the measure of the energy of a curve:

$$E(A; t_1, t_2) = \int_{t_1}^{t_2} < \frac{dA}{dt}, \frac{dA}{dt} > dt. \tag{7.53}$$

Since G is not positive definite, the energy of a curve can be in general negative. There are also non-trivial curves (that is, curves that are not identically equal to a point) which have zero energy.

Two special cases of metric (7.42) are of particular interest. With $\alpha = 0$ and $\beta = 1$ we obtain the metric:

$$G = \begin{bmatrix} 0_{3\times3} & I_{3\times3} \\ I_{3\times3} & 0_{3\times3} \end{bmatrix}.$$

This metric, taken as a quadratic form on $se(3)$, is known as the Klein form. The eigenvalues for the metric are $\{1, 1, 1, -1, -1, -1\}$ and the form is therefore non-degenerate. For a screw motion $A(t) = A_0 \exp(t\, S)$ where $S = \{\omega, v\} \in se(3)$, the energy of the segment $t \in [0, 1]$ is given by $E(A) = 2\, \omega^T v$. If $\omega \neq 0$, the quantity:

$$h = \frac{\omega^T v}{|\omega|^2} \tag{7.54}$$

is called the *pitch* of the screw motion [68]. The pitch measures the amount of translation along the screw axis during the screw motion. Zero energy screw motions therefore either have zero pitch (the motion is pure rotation) or infinite pitch ($\omega = 0$, the motion is pure translation). Screw motions with positive energy are those with positive pitch. Trajectories for such motions correspond to right-handed helices and the motions are thus called right-handed screw motions. Analogously, screw motions with negative energy are the left-handed screw motions. Since pure rotations and pure translations are zero-energy motions, it is always possible to find a zero energy curve between two arbitrary points by breaking the motion into a segment consisting of a pure rotation followed by a segment of a pure translation.

By letting $\alpha = 1$ and $\beta = 0$, we get the semi-definite metric:

$$G = \begin{bmatrix} I_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} \end{bmatrix}.$$

This metric, as a form on $se(3)$, is called the Killing form. Its eigenvalues are $\{1, 1, 1, 0, 0, 0\}$ hence it is degenerate. The energy of a screw motion with $S = \{\omega, v\}$ is equal to $\omega^T \omega$ so it is always non-negative. Pure translations are zero-energy motions while any motion involving rotation has positive energy.

In the general case, $\alpha \neq 0$ and $\beta \neq 0$, the energy of a unit screw motion $A(t) = A_0 \exp(t\,S)$ where $S = \{\omega, v\}$ and $t \in [0,1]$, is $\omega^T(\alpha\,\omega + 2\beta\,v)$. Pure translations ($\omega = 0$) thus have zero energy. For a general screw motion ($\omega \neq 0$), the energy of the segment $t \in [0,1]$ is $|\omega|^2(\alpha + 2\beta h)$. The sign of the energy of a general motion therefore depends on α and β.

7.3 Affine connections on $SE(3)$

There is no natural choice of a metric on $SE(3)$. In the previous section, we chose a particular family of curves and found the metric G for which these curves were geodesics. Through the metric connection, a metric also provides a natural way to differentiate vector fields.

In this section, our primary objective is to find a connection which produces an acceleration vector that is physically meaningful. We therefore start by introducing a connection on $SE(3)$ which allows us to compute the covariant derivative and obtain the acceleration. By requiring that the acceleration computed with the covariant derivative agrees with the acceleration as computed in kinematics, we obtain a family of possible affine connections. We then show that in this family there is a unique symmetric connection. Finally, we determine the class of Riemannian metrics which are compatible with this symmetric affine connection.

7.3.1 Kinematic connection

As discussed in Section 7.1, the Lie algebra $se(3)$ represents the space of twists. We saw that the basis of left invariant vector fields $\hat{L}_i$ (Equation 7.10) provides a natural framework for studying motion on $SE(3)$: The components of tangent vector fields with respect to this basis correspond to the components of the instantaneous twist associated with the motion, expressed in the body-fixed coordinate frame. To obtain the acceleration, we have to compute the covariant derivative of the velocity (the tangent vector field) along the curve describing the motion.

We now turn our attention to the acceleration vector. Let $A(t)$ be a curve describing motion of a rigid body. Let $S(t) = \{\omega, v\}$ represent the instantaneous twist of the rigid body, expressed in the moving frame {M} fixed to the rigid body. More precisely, ω represents the angular velocity of the rigid body while v is the linear velocity of the origin O' of the body fixed reference frame {M} (see Figure 7.1). We would like to differentiate the velocity of the point O' to obtain the acceleration. The acceleration can be represented by a vector pair $\{\alpha, a\}$, where α is the angular acceleration of the rigid body and a is the acceleration of the point O', both expressed in the frame {M}. This acceleration vector pair can be shown [37] to be equal to:

$$\{\alpha, a\} = \{\dot{\omega}, \dot{v}\} + \{0, \omega \times v\}. \tag{7.55}$$

The first term in this equation is simply the derivative of the components of the angular and linear velocity of the point O'. This term is also called the spatial acceleration [37]. However, angular and linear velocities were expressed in the body fixed frame {M} which rotates as the body moves. We must therefore add a convective term to describe the

contribution of this rotation to the acceleration of the rigid body. This term is an artifact of expressing the velocities in a frame that rotates.

On the other hand, geometrically, the acceleration of the rigid body, $\mathcal{A}$, is given by the covariant derivative of the velocity vector field $V = \frac{dA}{dt}$ along $A(t)$:

$$\mathcal{A} = \nabla_V V. \tag{7.56}$$

It is important to see the difference between the twist $S(t)$ used to obtain Equation (7.55) and the velocity vector field $V(t) = V(A(t))$: the twist $S(t)$ belongs to $se(3)$, while $V(t)$ belongs to the tangent space $T_{A(t)}\,SE(3)$!

Let $V = V^i \hat{L}_i$ and let $\{\omega, v\}$ be the corresponding vector of components. According to Equation (7.55), the affine connection that produces physically meaningful acceleration must satisfy:

$$\nabla_V V = \{\dot{\omega}, \dot{v}\} + \{0, \omega \times v\}. \tag{7.57}$$

In components, $\nabla_V V$ can be rewritten as:

$$\nabla_V V = \sum_k \frac{dV^k}{dt}\hat{L}_k + \sum_k \sum_{1 \le i,j \le 6} V^i V^j \Gamma^k_{ji} \hat{L}_k, \tag{7.58}$$

where Γ^k_{ji} are the Christoffel symbols (which define the affine connection) for the basis $\hat{L}_i$. Expressions (7.57) and (7.58) will be the same if the first and the second term in Equation (7.57) correspond to the first and the second term in Equation (7.58), respectively. Obviously, the first terms are the same regardless of the choice of the affine connection. However, because of the symmetry, the coefficient for the product $V^i V^j$, $1 \le i < j \le 6$ is $\Gamma^k_{ji} + \Gamma^k_{ij}$ (for $i = j$, the coefficient is Γ^k_{ii}). Therefore, from Equation (7.57) we conclude that:

$$\Gamma^k_{ij} + \Gamma^k_{ji} = a^k_{ij} \quad 1 \le i < j \le 6 \tag{7.59}$$

where a^k_{ij} are constants that can be directly obtained from Equation (7.57). The only non-zero values a^k_{ij} are:

$$\begin{aligned} &a^6_{24} = -1 \quad a^5_{34} = 1 \quad\quad a^6_{15} = 1 \\ &a^4_{35} = -1 \quad a^5_{16} = -1 \quad a^4_{26} = 1. \end{aligned} \tag{7.60}$$

The system (7.59) does not contain enough equations to solve for Γ^k_{ij} if $i \neq j$ (for $i = j$ we have $\Gamma^k_{ij} = 0$). Therefore, there is a whole family of affine connections on $SE(3)$ that produce a physically meaningful acceleration.

To further restrict the choice of the connection, we might ask if any of the connections that satisfy Equation (7.59) can be a Riemannian connection. Since every Riemannian connection is symmetric, we require that in addition to Equation (D.2) the connection also satisfies Equation (7.27):

$$\nabla_X Y - \nabla_Y X = [X, Y]. \tag{7.61}$$

It immediately follows that for the basis $\hat{L}_i$, the symmetry of the connection implies:

$$\Gamma^k_{ji} - \Gamma^k_{ij} = C^k_{ij}. \tag{7.62}$$

Equations (7.59) and (7.62) together uniquely determine the Christoffel symbols Γ^k_{ji} and therefore the connection. We call this connection the *kinematic connection*. The non-zero Christoffel symbols for the kinematic connection are:

$$\begin{aligned} &\Gamma^3_{21} = \Gamma^2_{13} = \Gamma^1_{32} = \tfrac{1}{2}, \quad \Gamma^3_{12} = \Gamma^2_{31} = \Gamma^1_{23} = -\tfrac{1}{2} \\ &\Gamma^6_{51} = \Gamma^4_{62} = \Gamma^5_{43} = 1, \quad \Gamma^6_{42} = \Gamma^4_{53} = \Gamma^5_{61} = -1 \end{aligned} \tag{7.63}$$

7.3.2 Metrics compatible with the kinematic connection

As seen in Section 7.1.4, a connection is Riemannian if and only if it is symmetric and compatible with the metric. Since we explicitly required that the kinematic connection be symmetric, we can try to find a metric which is compatible with the connection. In general, such metric may not exist!

If a metric is compatible with the connection, then:

$$Z < X, Y > = < \nabla_Z X, Y > + < X, \nabla_Z Y >, \tag{7.64}$$

where X, Y and Z are arbitrary vector fields. By substituting the basis vector fields $\hat{L}_i$, $\hat{L}_j$ and $\hat{L}_k$ for X, Y and Z, the compatibility condition becomes:

$$\hat{L}_k(g_{ij}) = \sum_l (\Gamma^l_{ik} g_{lj} + \Gamma^l_{jk} g_{li}) \tag{7.65}$$

where the Christoffel symbols Γ^k_{ji} were computed above. Equation (7.65) generates a system of 126 partial differential equations for metric coefficients $\{g_{ij} \mid 1 \leq i \leq j \leq 6\}$. Note that for $k > 3$, $\Gamma^j_{ik} = 0$, so $\hat{L}_k(g_{ij}) = 0$. The system of equations obtained from Equation (7.65) for $k \leq 3$ is given in Appendix D. The first step in finding the solution of this system of equations is represented by the following lemma.

Lemma 7.11 *If the coefficients of a Riemannian metric G satisfy Equation (7.65), the metric has the form:*

$$G = \begin{bmatrix} \alpha I & 0 \\ 0 & G_p \end{bmatrix}, \tag{7.66}$$

where α is a constant and G_p is an arbitrary positive definite symmetric matrix that smoothly varies from point to point.

Proof: We use Lemma 7.5 again. Take the following subset of equations of the system (D.2):

$$\hat{L}_1(g_{11}) = 0 \quad \hat{L}_2(g_{11}) = -g_{13} \quad \hat{L}_3(g_{11}) = g_{12} \tag{7.67}$$

According to Lemma 7.5 the following equality holds:

$$-\hat{L}_1(g_{13}) = g_{12}.$$

By substituting for $\hat{L}_1(g_{13})$ from (D.2), we obtain $\frac{1}{2} g_{12} = g_{12}$, which gives $g_{12} = 0$. Substituting in the system (D.2), we next obtain:

$$g_{13} = 0 \quad g_{23} = 0 \quad g_{11} = g_{22} \quad g_{11} = g_{33} \tag{7.68}$$

It is easy to see that these equations imply $\hat{L}_i(g_{11}) = 0, \;\; 1 \leq i \leq 6$, which together with Equation (7.68) results in:

$$g_{11} = g_{22} = g_{33} = \alpha,$$

where α is a constant. Therefore, the upper-left 3×3 block in the matrix G is of the form $\alpha I_{3\times 3}$, where I is the identity matrix.

By taking equations:

$$\hat{L}_1(g_{14}) = 0 \quad \hat{L}_2(g_{14}) = -\tfrac{1}{2}g_{34} - g_{16} \quad \hat{L}_3(g_{14}) = \tfrac{1}{2}g_{24} + g_{15}, \tag{7.69}$$

and again using Lemma 7.5, we get $g_{24} = 0$. By substituting this in the system (D.2) it is easy to see that all the entries in the upper-right 3×3 block of the matrix G (and due to the symmetry of G also in the lower-left 3×3 block) are equal to 0. The matrix G must be positive definite symmetric, therefore also G_p must be positive definite symmetric. □

Proposition 7.12 *A left invariant metric is compatible with the kinematic connection if and only if the matrix of metric coefficients $G = [g_{ij}]$ is of the form:*

$$G = \begin{bmatrix} \alpha I & 0 \\ 0 & \beta I \end{bmatrix}, \tag{7.70}$$

where α and β are arbitrary constants.

Proof: If a metric is left invariant then the matrix G is constant. But if $g_{ij} = \text{const.}$, then $\hat{L}_i(g_{ij}) = 0$. It is then easy to check that the system of equations (D.2) implies the form of G in (7.70). □

Note that the metric (7.70) is exactly the same as the metric (7.32) that we investigated earlier.

To determine whether there are other solutions of the system (D.2), it helps to investigate what is common to metrics which have the same metrical connection. The geometric entity determined from the connection are geodesics (Equation 7.30). Two metrics with the same connection thus have the same geodesics. By reasoning similar to that which led to the kinematic connection we can prove that also the converse is true: The family of geodesics uniquely determines the symmetric connection. To find other solutions of the system (D.2), we therefore try to find metrics which have the same geodesics as metric (7.70). Alternatively, this process can be viewed as the study of diffeomorphisms of $SE(3)$ which map geodesics to geodesics (such maps are called *affine maps*).

We saw earlier that a geodesic for the metric (7.70) between two points A_1 and A_2 on $SE(3)$ is a product of a geodesic on $SO(3)$ equipped with a bi-invariant metric (a constant velocity rotation around an appropriate axis) and a geodesic on Euclidean space $\mathbb{R}^3$ (a straight line). But on $\mathbb{R}^3$, straight lines are geodesics for an arbitrary inner product (defined by a positive definite constant matrix). Therefore, any product metric on $SO(3) \times \mathbb{R}^3$ with the bi-invariant metric on $SO(3)$ and an inner product metric on $\mathbb{R}^3$ has the same geodesics as the metric (7.70). It can be shown (see Appendix A) that for the basis $\hat{L}_i$ such metric has the form:

$$G(R, d) = \begin{bmatrix} \alpha I & 0 \\ 0 & R^T W R \end{bmatrix} \tag{7.71}$$

where W is a constant positive definite symmetric matrix which defines the inner product on $\mathbb{R}^3$. We state this as a lemma:

Lemma 7.13 *If a metric G has the form (8.5), then it is a solution of the system of equations (D.2).*

Note that the form of G in Lemma 7.13 is consistent with that in Lemma 7.11. It is also obvious that the metric (7.70) can be obtained as a special case of (8.5) by substituting $W = \beta I$.

Lemma 7.13 identifies a family of Riemannian metrics that are compatible with the kinematic connection. We would like to know whether there are any other metrics which are compatible with the kinematic connection. Before we answer this question, we prove the following lemma:

Lemma 7.14 *Two metrics on $SE(3)$ which have the same Riemannian connection and are equal at a point, are equal everywhere.*

Proof: Let G_1 and G_2 be the two metrics and let A_0 be the point where they are equal: $G_1(A_0) = G_2(A_0)$. If a connection is compatible with the metric then the parallel transport preserves the inner product. Take an arbitrary point $A \in SE(3)$. We can always find a curve γ which connects A_0 with A. Since at A_0, G_1 and G_2 are equal, we can choose a basis X_i for the tangent space $T_{A_0}SE(3)$ which is orthonormal in both metrics. The two metrics have the same connection and we can parallel transport vectors X_i at A_0 to vectors X_i' at A along γ. Since both metrics are compatible with the connection and the parallel transport preserves the inner product, vectors X_i' at A are orthonormal in both metrics. But this means that the two metrics are equal at A.

□

Remark 7.15 Parallel transport along a closed curve γ which starts and ends at $A \in SE(3)$, maps an element $X \in T_ASE(3)$ to another element $X' \in T_ASE(3)$. The collection of such mappings that we obtain by taking all possible closed curves that start and end at A forms a group called *the holonomy group* of the connection with the reference point A. Since the parallel transport preserves the inner product, the elements of the holonomy group are orthogonal transformations with respect to any metric compatible with the connection.

We can now state the second major result of the chapter:

Theorem 7.16 *A metric $G(R,d)$ is compatible with the kinematic connection given by Equation (7.59) and (7.62) if and only if it has the form:*

$$G(R,d) = \begin{bmatrix} \alpha I_{3\times 3} & 0_{3\times 3} \\ 0_{3\times 3} & R^T W R \end{bmatrix} \tag{7.72}$$

where W is a constant positive definite symmetric matrix.

Proof: The "if" part of the Theorem is just Lemma 7.13. To prove the "only if" direction, let G be a metric compatible with the kinematic connection. According to Lemma 7.11, at the identity the metric G has the form:

$$G = \begin{bmatrix} \alpha I & 0 \\ 0 & M \end{bmatrix}, \tag{7.73}$$

where M is a positive definite, smoothly varying, symmetric matrix. But there exists a metric G' of the form (7.72) with $W = M$ which is equal to G at the identity. The two metrics are both compatible with the kinematic connection, so according to Lemma 7.14, they are the same. Metric G therefore has the form (7.72). □

Remark 7.17 All the metrics of the form (7.72) are isometric. In other words, if G_1 and G_2 are two such metrics, there is an isometry between $SE(3)$ equipped with the metric G_1 and $SE(3)$ equipped with the metric G_2. (This isometry does not preserve the group structure on $SE(3)$, though!) The two manifolds are isometric because of the product structure of the metrics and the fact that any two metrics on a Euclidean space are isometric.

7.4 Discussion

In this chapter we saw how $SE(3)$ can be endowed with additional structure so that some well-known results in kinematics can be interpreted through the geometry of $SE(3)$. First, we show that a natural setting to study screw motions is $SE(3)$ equipped with a metric chosen from a two-parameter family of semi-Riemannian metrics. These metrics are indefinite and in general they are non-degenerate. Viewed as quadratic forms on $se(3)$, the metrics are linear combinations of the Killing form and the Klein form. Any non-degenerate metric in this family defines the unique symmetric connection for which geodesics are screw motions. When the metric is degenerate (as a form on $se(3)$ it is a scalar multiple of the Killing form), the symmetric connection compatible with the metric is no longer unique and neither are the geodesics. However, screw motions are still one possible set of geodesics of degenerate metrics derived from the Killing form.

To study acceleration or higher order derivatives of the velocity, $SE(3)$ must be equipped with an affine connection. The choice of the affine connection is restricted if we want to obtain the acceleration of a rigid body by covariant differentiation of the velocity vector field. Further, if we require that the connection is symmetric, such connection is unique. In this case, a family of Riemannian metrics which are compatible with the symmetric connection can be identified. All of them are product metrics – as a Riemannian manifold, $SE(3)$ is a Cartesian product of $SO(3)$ with the bi-invariant metric and $\mathbb{R}^3$ with the inner product metric. Alternatively, the symmetric connection studied in Section 7.3 can be viewed as the symmetric part of a general connection with non-zero torsion (asymmetric part). Since geodesics do not depend on torsion, the family of metrics compatible with the symmetric part of the connection consists precisely of those metrics which have the same geodesics as the connection.

Chapter 8

Task space trajectory planning

For applications, kinematic motion planning is employed more frequently than dynamic motion planning. The main reason is that it is much simpler. Kinematic motion planning will be sufficient as long as there exists a suitable way of computing the actuator forces from the kinematic trajectory. This is the case, for example, if there exists a 1-1 mapping between the task space and the actuator space. We also use kinematic motion planning when a dynamic model of the system is difficult to derive: we abstract the dynamic model with a kinematic model.

Kinematic motion planning can be performed either in the task space or in the joint space. Since the task is specified in the task space, it is natural to compute the motion plan directly in this space. Such motion plan will be in general independent of the mechanical structure of the robot. In robotics we are usually interested in smooth trajectories. They are preferred because the electro-mechanical system is limited by its actuator size and its control bandwidth so it can not produce large velocities and accelerations, and because movements with high acceleration and/or jerk can excite the structural natural frequencies in the system. A typical example is programming of industrial robots for tasks such as welding and painting where a "teaching" process is employed to record intermediate positions and the final trajectory is obtained by interpolation [108]. Similarly, in computer graphics, smoothness is required to obtain realistic motions or motions that "look" natural when key frames are employed for three-dimensional animation. Each key frame represents an intermediate position and orientation (pose) and is specified by the user or programmer. It is then necessary to automatically generate a smooth trajectory passing through the key frames [58].

When comparing different methods for kinematic motion planning and trajectories that these methods generate, we usually look for the following properties:

- The trajectories must be independent of the description of the space. In this way, computations performed with different choices of coordinates will produce consistent results.

- The trajectories should be independent of the choice of the inertial reference frame {F} and the body fixed frame {M} (Figure 7.1). That is, it should not matter where we put the inertial frame and which frame we choose on the rigid body, the resulting trajectory that the rigid body follows should be the same.

- The trajectories must have good performance for the chosen task.

Many commonly used motion planning schemes produce trajectories that depend on the choice of the parameterization of the task space. An example is shown in Figure 8.1. The trajectories in the figure were obtained by using a straight line interpolation in the chosen coordinate space. In Figure 8.1a, we used the canonical coordinates of the first kind defined by Equation (7.18). In Figure 8.1b, we used the canonical coordinates of the second kind defined by Equation (7.17), while for Figure 8.1c we chose the canonical coordinates of the second kind with the order of the basis vectors in Equation (7.17) reversed. The object moves in the plane $z = 0$.

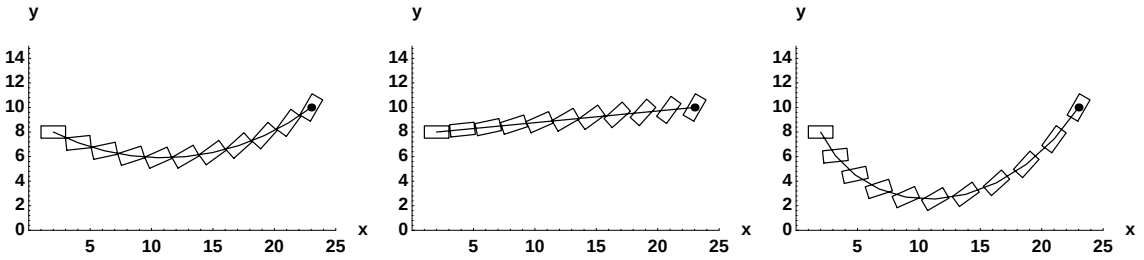


Figure 8.1: Motion of an object following a straight line in: (a) canonical coordinates of the first kind; (b) canonical coordinates of the second type (ordering $L_6, \ldots, L_1$); and (c) canonical coordinates of the second type (ordering $L_1, \ldots, L_6$).

Coordinate independence of the trajectories is assured if the motion planning scheme is derived within the framework of differential geometry. Differential geometry also provides a consistent way of extending the notion of differentiation from Euclidean space to an arbitrary manifold. In this way we can define different measures of smoothness of the trajectories. In addition, because of the group structure of the task space manifold, we can establish whether the trajectories are left or right invariant (see Sections 7.1.1 and 7.1.3) and thus independent of the choice of the reference frames.

Finding trajectories on $SE(3)$, that are left and right invariant and yield good performance turns out to be difficult if not impossible. From Corollary 7.3 it follows that screw motions are invariant with respect to the choice of the inertial and body fixed reference frames. However, according to Corollary 7.7 screw motions are not geodesics for any Riemannian metric and are therefore not minimizing any physically meaningful distance. Further, on $SE(3)$ there is no bi-invariant metric [80]. This implies that the invariance of the trajectories must be sacrificed for performance and all we can hope for is either independence of the trajectories with respect to the choice of the inertial frame {F} or independence with respect to the body fixed frame {M}.

There is extensive literature on trajectory generation in kinematics, robotics and computer graphics. In order to generate a smooth motion for a robotic arm from an initial to the final position, Whitney [152] and Pieper [110] proposed calculating the screw axis of the end effector displacement. In other words, they advocated a screw motion from the initial to the final position. Waldron [147] developed an algorithm that is based on a slight variation of Pieper's scheme and proposed a trapezoidal velocity profile (a constant

acceleration phase, followed by a constant velocity phase and finally a constant deceleration phase) for the screw motion. In all these schemes, although the screw motion is invariant with respect to rigid body transformations, it does not optimize a meaningful cost function. Paul [108] decomposes the desired displacement into a translation and two rotations each of which is smoothly parameterized with respect to time. The motion of the end-effector is obtained by a composition of these three displacements. He employs a fourth-order polynomial of time to obtain a smooth motion. Although there is some justification for the proposed trajectory, the approach will lead to different trajectories if different parameterization is chosen for the rotation or if the coordinate frames in which the trajectory is computed are changed. There is also no attempt to develop a measure of smoothness for three-dimensional motions.

Shoemake [130] proposed a scheme for interpolating rotations (in $SO(3)$) with Bezier curves. This idea was extended by Ge and Ravani [46] to $SE(3)$ and proposed for computer-aided geometric design. In both cases, the interpolating curves are screw motions and therefore invariant with respect to the choice of reference frames. However, the interpolating scheme produces a motion that does not posses these invariance properties. Further, these motions are not of minimal length for any meaningful metric. In contrast, Park and Ravani [107] use a scale-dependent left invariant metric to design Bezier curves for three-dimensional rigid body motion interpolation.

As in the first part of the dissertation, we formulate the motion planning problem as a variational problem; in particular, we compute maximally smooth trajectories between an initial and a final position and orientation. The measure of the lack of smoothness is chosen to be the integral over the trajectory of a cost function depending on velocity or its higher derivatives. Boundary conditions on the derivatives of desired order can be enforced by appropriately choosing the cost function. For example, by minimizing the norm of the velocity we obtain the shortest distance paths. The minimum-acceleration (minimum-jerk) trajectories can be made to satisfy boundary conditions on the velocities (accelerations). Dynamically smooth trajectories can be obtained by incorporating the inertia of the system into the cost function. A simple extension of the ideas in this chapter allows inclusion of intermediate positions and orientations and lends itself to motion interpolation.

Necessary conditions for smooth curves on general manifolds were derived by Noakes *et al.* [101], and in parallel with our work by Camarinha *et al.* [25] and Crouch and Silva Leite [31]. In [101], necessary conditions for cubic splines which correspond to our minimum-acceleration curves are derived for an arbitrary manifold. These results are extended in [31] to the dynamic interpolation problem. In [25] necessary conditions for curves minimizing the integral of the norm of an arbitrary derivative of velocity are derived. None of these works deals specifically with computing the trajectories on $SE(3)$, nor do they address the choice of the metric for the space. Since there is no natural metric for $SE(3)$ [27, 92], the choice of metric for trajectory planning becomes an important issue.

The chapter is organized as follows. We start with a discussion on the choice of metric for trajectory planning on $SE(3)$. We propose a left invariant metric given by the kinetic energy of a rigid body and derive the expressions for the covariant derivative given by this metric. We use these geometric constructs to formalize the ideas of acceleration and jerk on $SE(3)$. Most of these results are presented here for the first time. We then describe the variational problems that need to be solved in order to calculate the shortest distance, minimum-acceleration and minimum-jerk trajectories. While some of these results were

derived in [101] and [25], we present alternative proofs and specialize the results to $SE(3)$. We continue by deriving analytical solutions for smooth trajectories in some special cases. At the end of the chapter, we show some numerically computed trajectories to illustrate their dependence on the choice of the metric, reference frames and boundary conditions.

8.1 Choice of metric on $SE(3)$ for motion planning

To measure the length of a vector or a distance between two points on a manifold, we need a Riemannian metric. There is no clear choice of metric on $SE(3)$. In the previous chapter, we obtained two families of metrics by specifying the family of geodesics and enforcing a particular expression for covariant derivative. In this section we are interested in metrics that are suitable for motion planning on $SE(3)$. We focus on the left invariant metrics (see Section 7.1.3) which produce trajectories that are invariant with respect to the choice of the inertial frame, but in general change if the body-fixed frame is changed. An example of a left invariant metric was proposed by Park and Brockett [106] and was discussed in Section 7.2. The matrix W in Equation (7.19) for this metric is given by:

$$W = \begin{bmatrix} \alpha I & 0 \\ 0 & \beta I \end{bmatrix} \tag{8.1}$$

where α and β are positive scalars. This metric possesses several attractive features.

1. By construction, the metric is left invariant and does not depend on the choice of the inertial (fixed) reference frame. This property is important for dynamic analysis of mechanical systems.

2. When restricted to the group of rotations, $SO(3)$, it is bi-invariant [121].

3. It preserves the isotropy of $\mathbb{R}^3$ [105], which means that the metric properties of the space are the same in every direction.

4. For an arbitrary choice of the positive constants α and β, the basis vector fields $\hat{L}_i$ are orthogonal. When $\alpha = \beta = 1$, they become orthonormal.

5. The metric is compatible with the acceleration connection (Proposition 7.12). The Riemannian connection corresponding to this metric thus yields the acceleration that is used in kinematics.

However, metric (8.1) depends on the choice of the two scalars, α and β, which act like scaling factors for angular velocities and linear velocities. In *kinematic analysis* there is no *a priori* justification for choosing them.

Another metric that is attractive for trajectory planning can be obtained by considering the dynamic properties of the rigid body. The kinetic energy of a rigid body is a scalar that does not depend on the choice of the inertial reference frame. It thus defines a left invariant metric. For this metric, the matrix W in Equation (7.19) is the inertia matrix and $\frac{1}{2} < V, V >$ corresponds to the kinetic energy of the rigid body moving with a velocity V.

If the body-fixed reference frame is attached at the centroid and aligned with the principal axes, then we have:

$$W = \begin{bmatrix} H & 0 \\ 0 & mI \end{bmatrix}, \tag{8.2}$$

where m is the mass of the rigid body and H is the matrix:

$$H = \begin{bmatrix} H_{xx} & 0 & 0 \\ 0 & H_{yy} & 0 \\ 0 & 0 & H_{zz} \end{bmatrix},$$

with H_{xx}, H_{yy}, and H_{zz} denoting the moments of inertia about the x, y, and z axes, respectively. If $\{\omega, v\}$ is the vector pair associated with the vector V, this vector pair represents the instantaneous twist associated with the motion, expressed in the body-fixed reference frame. The norm of the vector V thus assumes the familiar expression:

$$< V, V >= \omega^T H \omega + m v^T v. \tag{8.3}$$

Now assume that the body fixed frame {M} is displaced by the matrix:

$$C = \begin{bmatrix} R & d \\ 0 & 1 \end{bmatrix}$$

to a new frame $\{\mathrm{M}\}_C$. The kinetic energy does not change if the body-fixed frame is changed. It is not difficult to check that this implies that the matrix W_C defining the energy metric for the new description of the motion of the rigid body is:

$$W_C = \begin{bmatrix} R^T H R - m R^T D^2 R & -m R^T D R \\ m R^T D R & mI \end{bmatrix}, \tag{8.4}$$

where D is the skew-symmetric matrix corresponding to the vector d. This is therefore the most general form of the inertia matrix and can be viewed as a spatial version of Steiner's parallel-axis theorem.

Remark 8.1 In the Appendix A we show that the matrix of metric coefficients, G, for a product metric on $SO(3) \times \mathbb{R}^3$ induced by a left invariant metric Q on $SO(3)$ and a metric W on $\mathbb{R}^3$, has the following form:

$$G(R, d) = \begin{bmatrix} Q & 0 \\ 0 & R^T W R \end{bmatrix}, \tag{8.5}$$

when expressed in the basis $\hat{L}_i$. Here, Q and W are arbitrary positive-definite 3×3 matrices. If we set $W = \gamma I$, the product metric becomes:

$$G = \begin{bmatrix} Q & 0 \\ 0 & \gamma I \end{bmatrix}. \tag{8.6}$$

The metrics (8.1) and (8.2) both have this form and are therefore obtained from a product metric. In other words, there is an isometry between $SE(3)$ endowed with any of these

metrics and the product space $SO(3) \times \mathbb{R}^3$ with appropriately defined metrics on $SO(3)$ and $\mathbb{R}^3$, respectively. Further, since any metric of the form (8.4) is isometric to a metric of the form (8.2), any metric induced by the kinetic energy will be isometric to a product metric. Since none of the functionals that we later use to define smoothness of a curve depend on the group structure of $SE(3)$, calculations in the examples could be simplified by performing them on the product space $SO(3) \times \mathbb{R}^3$ instead. However, the key results in this chapter are derived for a general metric and are not limited to product metrics. Hence, for the derivations we do not take advantage of the product structure for $SE(3)$.

8.1.1 Riemannian connection on $SE(3)$

In this section we find the Riemannian connections that correspond to the left invariant metrics (8.1) and (8.2). We start with an elementary result relating the Christoffel symbols and the structure constants for an arbitrary Lie group. It is not difficult to show [34] that if ∇ is the Riemannian connection then for any three vector fields X, Y and Z:

$$\begin{aligned} < Z, \nabla_X Y >= \frac{1}{2} \{ & Y < X, Z > + X < Z, Y > - Z < X, Y > + \\ & + < [Z, Y], X > + < [Z, X], Y > + < [X, Y], Z > \} \end{aligned} \tag{8.7}$$

This immediately implies:

Proposition 8.2 *If ∇ is the Riemannian connection compatible with a left invariant metric described by a matrix $W = [w_{ij}]$, the Christoffel symbols for the basis $\hat{L}_i$ are given by*

$$\Gamma_{ji}^k = \frac{1}{2} \sum_m w_{km}^{-1} \left(C_{ij}^s w_{sm} + C_{mj}^s w_{si} + C_{mi}^s w_{sj} \right), \tag{8.8}$$

where C_{ij}^k are the structure constants of the Lie algebra and $w_{km}^{-1} = (W^{-1})_{km}$.

Note that if $X = X^i \hat{L}_i$ and $Y = Y^i \hat{L}_i$ are any two vector fields[1], then

$$\nabla_X Y = \nabla_{X^j \hat{L}_j} Y^i \hat{L}_i = \frac{dY^i}{dt} \hat{L}_i + X^i Y^j \nabla_{\hat{L}_i} \hat{L}_j = \frac{dY^i}{dt} \hat{L}_i + X^i Y^j \Gamma_{ji}^k \hat{L}_k, \tag{8.9}$$

where $\frac{d}{dt}$ is the derivative along the integral curve of X and Γ_{ji}^k are obtained from Equation (8.8). However, for $SE(3)$ and the metrics (8.1) and (8.2), more compact expressions can be obtained directly from Equation (8.7). First, we prove the following lemma for $SE(3)$:

Lemma 8.3 *Let $X = X^i \hat{L}_i$, $Y = Y^i \hat{L}_i$ and $Z = Z^i \hat{L}_i$ be three arbitrary vector fields and let the corresponding vector pairs be $\{\omega_x, v_x\}$, $\{\omega_y, v_y\}$, and $\{\omega_z, v_z\}$, respectively. If ∇ is the Riemannian connection corresponding to the left-invariant Riemannian metric (7.19), then*

$$\begin{aligned} < Z, \nabla_X Y > \quad = \quad & < Z, X(Y^i) \hat{L}_i > \\ & + \frac{1}{2} [< \{(\omega_z \times \omega_y), (\omega_z \times v_y + v_z \times \omega_y)\}, \{\omega_x, v_x\} > \\ & + < \{(\omega_z \times \omega_x), (\omega_z \times v_x + v_z \times \omega_x)\}, \{\omega_y, v_y\} > \\ & + < \{(\omega_x \times \omega_y), (\omega_x \times v_y + v_x \times \omega_y)\}, \{\omega_z, v_z\} >] \end{aligned} \tag{8.10}$$

[1] Starting from this point we use the Einstein summation convention to simplify the notation.

Proof: The result of the Lemma follows directly from Equation (8.7). The Lie bracket of any two vector fields is:

$$[X,Y] = X^i Y^j [\hat{L}_i, \hat{L}_j] + X(Y^i)\hat{L}_i - Y(X^i)\hat{L}_i,$$

where $X(f)$ denotes the action of the vector field on a scalar function f (See Section 7.1.1). Rewritten in terms of the pairs $\{\omega_x, v_x\}$ and $\{\omega_y, v_y\}$, the first term becomes

$$X^i Y^j [\hat{L}_i, \hat{L}_j] = \{\omega_x \times \omega_y, \omega_x \times v_y + v_x \times \omega_y\}$$

Thus, in Equation (8.7),

$$\begin{aligned} < Z, [X,Y] > &= < \{\omega_z, v_z\}, \{(\omega_x \times \omega_y), (\omega_x \times v_y + v_x \times \omega_y)\} > \\ &\quad + < Z, X(Y^i)\hat{L}_i > - < Z, Y(X^i)\hat{L}_i > \end{aligned}$$

Furthermore, if W_{ij} are the entries of W in Equation (7.19),

$$\begin{aligned} X < Y, Z > &= X(Y^i W_{ij} Z^j) = X(Y^i) W_{ij} Z^j + Y^i W_{ij} X(Z^j) \\ &= < X(Y^i)\hat{L}_i, Z > + < Y, X(Z^i)\hat{L}_i > \end{aligned} \tag{8.11}$$

The result in Equation (8.10) follows if we similarly expand and add all the terms in Equation (8.7).

□

Proposition 8.4 *Let $X = X^i \hat{L}_i$ and $Y = Y^i \hat{L}_i$ be two arbitrary vector fields. If ∇ is the Riemannian connection corresponding to the Riemannian metric (8.1), then*

$$\nabla_X Y = \{\frac{d\omega_y}{dt} + \frac{1}{2}\omega_x \times \omega_y, \frac{dv_y}{dt} + \omega_x \times v_y\}, \tag{8.12}$$

where $\frac{d}{dt}$ is the derivative along the integral curve of X. The expression is independent of the scaling constants α and β.

Proof: We employ Lemma 8.3 and use the metric (8.1):

$$\begin{aligned} < Z, \nabla_X Y > &= < Z, X(Y^i)\hat{L}_i > \\ &\quad + \frac{1}{2}[\alpha\ (\omega_z \times \omega_y) \cdot \omega_x + \beta\ (\omega_z \times v_y + v_z \times \omega_y) \cdot v_x \\ &\quad + \alpha\ (\omega_z \times \omega_x) \cdot \omega_y + \beta\ (\omega_z \times v_x + v_z \times \omega_x) \cdot v_y \\ &\quad + \alpha\ (\omega_x \times \omega_y) \cdot \omega_z + \beta\ (\omega_x \times v_y + v_x \times \omega_y) \cdot v_z] \\ &= < Z, X(Y^i)\hat{L}_i > + \frac{1}{2}[\alpha\ (\omega_x \times \omega_y) \cdot \omega_z + 2\ \beta\ (\omega_x \times v_y) \cdot v_z] \\ &= < Z, X(Y^i)\hat{L}_i + \{\frac{1}{2}(\omega_x \times \omega_y), (\omega_x \times v_y)\} >, \end{aligned}$$

where '$\cdot$' denotes the dot product on $\mathbb{R}^3$. Since the equation must hold for an arbitrary Z, we can write:

$$\nabla_X Y = X(Y^i)\hat{L}_i + \{\frac{1}{2}\omega_x \times \omega_y, \omega_x \times v_y\}$$

If $\frac{d}{dt}$ is the derivative along the integral curve of X,

$$X(Y^i)\hat{L}_i = \frac{dY^i}{dt}\hat{L}_i = \{\frac{d\omega_y}{dt}, \frac{dv_y}{dt}\}$$

Hence the result in Equation (8.12). □

Proposition 8.5 *Let $X = X^i\hat{L}_i$ and $Y = Y^i\hat{L}_i$ be two arbitrary vector fields. If ∇ is the Riemannian connection corresponding to the Riemannian metric (8.2), then*

$$\nabla_X Y = \begin{bmatrix} \frac{d\omega_y}{dt} + \frac{1}{2}\left[(\omega_x \times \omega_y) + H^{-1}(\omega_x \times (H\omega_y)) + H^{-1}(\omega_y \times (H\omega_x))\right] \\ \frac{dv_y}{dt} + \omega_x \times v_y \end{bmatrix} \tag{8.13}$$

where $\frac{d}{dt}$ is the derivative along the integral curve of X. The translational component of the expression $\nabla_X Y$ is independent of the choice of matrix H and thus independent of the choice of the metric on $SO(3)$.

Proof: We start with Lemma 8.3 and use the metric (8.2):

$$\begin{aligned}
< Z, \nabla_X Y > &= < Z, X(Y^i)\hat{L}_i > \\
&\quad + \frac{1}{2}[(\omega_z \times \omega_y)\cdot(H\omega_x) + m(\omega_z \times v_y + v_z \times \omega_y)\cdot v_x \\
&\quad + (\omega_z \times \omega_x)\cdot(H\omega_y) + m(\omega_z \times v_x + v_z \times \omega_x)\cdot v_y \\
&\quad + (\omega_x \times \omega_y)\cdot(H\omega_z) + m(\omega_x \times v_y + v_x \times \omega_y)\cdot v_z] \\
&= < Z, X(Y^i)\hat{L}_i > + \frac{1}{2}\,[2\,m(\omega_x \times v_y)\cdot v_z \\
&\quad - ((H\omega_x) \times \omega_y)\cdot\omega_z + (\omega_x \times (H\omega_y))\cdot\omega_z + (\omega_x \times \omega_y)\cdot(H\omega_z)] \\
&= < Z, X(Y^i)\hat{L}_i > + \frac{1}{2}\Big[2\,m(\omega_x \times v_y)\cdot v_z - (H^{-1}((H\omega_x) \times \omega_y))\cdot(H\omega_z) \\
&\quad + (H^{-1}(\omega_x \times (H\omega_y)))\cdot(H\omega_z) + (\omega_x \times \omega_y)\cdot(H\omega_z)\Big] \\
&= < Z, X(Y^i)\hat{L}_i > + < Z, \{\frac{1}{2}\,[(\omega_x \times \omega_y) \\
&\quad + H^{-1}(\omega_x \times (H\omega_y)) + H^{-1}(\omega_y \times (H\omega_x))\Big], (\omega_x \times v_y)\} >
\end{aligned}$$

Since the above is true for an arbitrary Z, this proves the proposition.

In the derivation, we have used the fact that H is symmetric and therefore for arbitrary vectors x and y,

$$x \cdot y = x \cdot (H^{-1}Hy) = (H^{-1^T}x)\cdot(Hy) = (H^{-1}x)\cdot(Hy).$$ □

8.1.2 Curvature of $SE(3)$

In the subsequent sections we will also need the expressions for the Riemannian curvature of $SE(3)$. By definition, the Riemannian curvature tensor $R(X,Y)$ is [34]:

$$R(X,Y)\,Z = \nabla_Y\nabla_X Z - \nabla_X\nabla_Y Z + \nabla_{[X,Y]}Z \tag{8.14}$$

The Riemannian curvature for the metric (8.1) can be computed directly in the components of the corresponding vector fields:

Proposition 8.6 *If X, Y and Z are three arbitrary vector fields on $SE(3)$ with the associated vector pairs $\{\omega_x, v_x\}$, $\{\omega_y, v_y\}$, and $\{\omega_z, v_z\}$, and $SE(3)$ has the Riemannian connection defined in Equation (8.12), then the Riemannian curvature $R(X,Y)\,Z$ is*

$$R(X,Y)Z = \{\frac{1}{4}(\omega_x \times \omega_y) \times \omega_z, 0\} \tag{8.15}$$

Proof: See Appendix B.1. □

8.1.3 Acceleration and jerk of rigid body motions in $\mathbb{R}^3$

Knowing how to compute the covariant derivative, we are in a position to obtain formulas for the acceleration and jerk. Here we use the scale-dependent left invariant metric from Equation (8.1). Since the connection coefficients and the covariant derivative are independent of the choice of constants α and β, the resulting expressions for acceleration and jerk will also be independent of these scale factors.

If V is the velocity (tangent to the curve) associated with the motion $A(t)$ of a rigid body, the acceleration is the covariant derivative of V along the curve $A(t)$. If $\{\omega, v\}$ is the corresponding velocity pair it immediately follows from Equation (8.12) that the acceleration is given by

$$\nabla_V V = \begin{bmatrix} \dot{\omega} \\ \dot{v} + \omega \times v \end{bmatrix}. \tag{8.16}$$

The resulting expression for the acceleration corresponds to the acceleration that is used in kinematics. This is not surprising given that the metric (8.1) is compatible with the acceleration connection (see Section 7.3.2).

The third derivative of motion, jerk, can be obtained by considering the covariant derivative of the acceleration along the curve $A(t)$:

$$\nabla_V \nabla_V V = \begin{bmatrix} \frac{d\dot{\omega}}{dt} + \frac{1}{2}\omega \times \dot{\omega} \\ \frac{d(\dot{v}+\omega \times v)}{dt} + \omega \times (\dot{v} + \omega \times v) \end{bmatrix}. \tag{8.17}$$

8.2 Necessary conditions for smooth trajectories

8.2.1 Variational calculus on manifolds

In this section we consider trajectories between a starting and a final position and orientation that minimize an integral cost functions while satisfying additional boundary conditions on the velocities and/or accelerations. The cost functions can be the kinetic energy of the rigid body, or some other measure of smoothness involving velocity or its higher derivatives. More specifically, we will be interested in curves $A : [a,b] \to SE(3)$ that minimize integrals of the form

$$J = \int_a^b < h(\frac{dA}{dt}), h(\frac{dA}{dt}) > dt \tag{8.18}$$

where boundary conditions on $A(t)$ and its derivatives may be specified at the end points a and b. The function h returns a vector field and usually involves one or more recursive applications of the covariant derivative. We will use a left invariant metric and the corresponding Riemannian connection. Unless otherwise specified, the scale-dependent left invariant metric from Equation (8.1) will be used.

independent of the choice of the inertial reference frame {F}, we will use a left invariant metric and the corresponding Riemannian connection.

We adapt methods from the classical calculus of variations to the differential geometric setting [34]. Noakes *et al.* [101] use such a framework to derive expressions for cubic splines on a general manifold and they provide the formulas for the group of rotations $SO(3)$. The cubic splines correspond to our minimum-acceleration curves and we derive the results from [101] using more direct approach. The necessary conditions for minimum-jerk trajectories were obtained in parallel with our work and using similar approach by Camarinha *et al.* [25].

In the calculus of variations, the first-order necessary conditions for the minimal solution are derived by studying variations of the optimal trajectory. These conditions take the form of Euler-Lagrange equations. We will pursue a coordinate-free formulation of the Euler-Lagrange equations which is independent of the particular choice of a coordinate chart. We illustrate the basic approach with the simple example in which the cost functional is the energy functional, where the solution is known to be a minimal geodesic on $SE(3)$.

If $A : [a, b] \to SE(3)$ is a piecewise smooth curve on the manifold, a variation of A is a continuous mapping $f : (-\epsilon, \epsilon) \times [a, b] \to SE(3)$ such that

1. $f(0, t) = A(t), \quad t \in [a, b]$

2. f is differentiable on all intervals $(-\epsilon, \epsilon) \times [t_i, t_{i+1}]$, where $[t_i, t_{i+1}]$ is an interval on which A is differentiable.

The variation is *proper*, if $f(s, a) = A(a)$ and $f(s, b) = A(b)$. For each s, we get a curve $\gamma_s(t) = f(s, t)$ which is called *a curve of variation* of the curve $A(t)$ [34]. The first requirement above implies that $A(t) = f_0(t)$. On the other hand, curves obtained by fixing t, therefore $\delta_t(s) = f(s, t)$, are called *transversal curves of variation*. The tangent vectors to the transversal curves at $s = 0$ (that is $\frac{d\delta_t(s)}{ds}\Big|_{s=0}$) define a vector field along $A(t)$, denoted by S, called *the variational field of* f. The tangent along a curve of variation of the curve $A(t)$ is $V = \frac{\partial f(s,t)}{\partial t}$.

Let $f(s, t)$ be a variation of $A(t)$, that is $f(0, t) = A(t)$, and let the vector fields V and S be defined by $V = \frac{\partial f(s,t)}{\partial t}$ and $S = \frac{\partial f(s,t)}{\partial s}$. We also require that the curves of variation $f_s(t) = f(s, t)$ satisfy the boundary conditions (i.e., the variation is proper). The value of the cost functional (8.18) on a curve of variation $\gamma_s(t)$ is

$$J(s) = \int_a^b < h(\frac{\partial f(s,t)}{\partial t}), h(\frac{\partial f(s,t)}{\partial t}) > dt, \quad s \in (-\epsilon, \epsilon). \tag{8.19}$$

If the curve $A(t) = f(0, t)$ is a stationary point of the above functional then the first variation $\frac{dJ(s)}{ds}$ must vanish for $s = 0$. In the next subsection we compute the first variation of the energy functional and show that the critical points, as expected, are geodesics. In

subsequent subsections, we consider minimization of the L^2 norms of acceleration and jerk along the trajectories.

8.2.2 Minimum-distance curves – geodesics

Park [105] has extensively studied geodesics on $SE(3)$ for the left invariant metric (8.1). The main objective of considering geodesics here is to illustrate the generality of our approach and to show how the approach can be used to obtain the result in [105].

Consider the energy functional

$$J = E(s) = \int_a^b < \frac{\partial f(s,t)}{\partial t}, \frac{\partial f(s,t)}{\partial t} > dt \tag{8.20}$$

The minimum-distance curves between two points on the manifold are critical points of this functional [34]. Let $V = \frac{\partial f(s,t)}{\partial t}$. The first variation of $E(s)$ can be simplified as follows:

$$\begin{aligned}
\frac{1}{2}E'(s) &= \frac{1}{2}\frac{d}{ds}\int_a^b < V,V > dt \\
&\stackrel{1}{=} \frac{1}{2}S\int_a^b < V,V > dt \\
&\stackrel{2}{=} \int_a^b < \nabla_S V, V > dt \\
&\stackrel{3}{=} \int_a^b < \nabla_V S, V > dt \\
&\stackrel{2}{=} \int_a^b (V < S,V > - < S, \nabla_V V >)dt \\
&\stackrel{4}{=} < S,V >|_a^b - \int_a^b < S, \nabla_V V > dt
\end{aligned} \tag{8.21}$$

The numbers above the equal signs correspond to the following identities that were used in the derivation:

(1) $\frac{df(s)}{ds} = Sf$

(2) $< \nabla_V S, U >= V < S,U > - < S, \nabla_V U >$

(3) $\nabla_V S = \nabla_S V + [V,S] = \nabla_S V$ (since S and V are derivatives with respect to coordinate curves t and s, $[V,S] = 0$).

(4) for $V = \frac{\partial}{\partial t}$, $\int_a^b V(f)dt = \int_a^b \frac{df(t)}{dt}dt = f(t)|_a^b$

The first and the fourth identity explain how a vector field operates on a function (see Section 7.1.1). The second and the third identity state that ∇ is a Riemannian connection, hence compatible with the metric (2) and symmetric (3).

If a curve is a critical point of the functional the first variation on that curve must vanish for an arbitrary variational vector field S. Since the endpoints $A(a)$ and $A(b)$ are fixed, $S(a) = S(b) = 0$, and the first term in Equation (8.21) vanishes. Therefore, if $A(t) = f(0,t)$ is a critical point:

$$\nabla_V V = 0, \tag{8.22}$$

where $V = \frac{dA(t)}{dt}$. We thus obtain, as expected, the equation defining a geodesic.

So far we have only used the fact that the set of all positions and orientations is a Riemannian manifold. To solve Equation (8.22) and find the geodesics on $SE(3)$, we express V as a linear combination of left invariant vector fields $\hat{L}_1, \ldots, \hat{L}_6$ according to Equation (7.10) and assemble the coefficients in the vector pair $\{\omega, v\}$.

Proposition 8.7 *A curve*

$$A(t) = \begin{bmatrix} R(t) & d(t) \\ 0 & 1 \end{bmatrix}$$

is a geodesic on $SE(3)$ equipped with the metric (8.1) if and only if the vector pair $\{\omega, v\}$ corresponding to the velocity vector field $V = \frac{dA}{dt}$ satisfies the equations:

$$\begin{aligned} \frac{d\omega}{dt} &= 0 \\ \frac{dv}{dt} &= -\omega \times v. \end{aligned} \tag{8.23}$$

The second equation in (8.23) can be simplified to the equation:

$$\ddot{d} = 0.$$

Proof: A curve $A(t)$ is a geodesic if and only if Equation (8.22) is satisfied. Substituting for $\nabla_V V$ from Equation (8.16), we obtain the equations in (8.23). The second equation in (8.23) can be written as:

$$\dot{v} + \omega \times v = 0.$$

By writing $\Omega = R^T \dot{R}$ and $v = R^T \dot{d}$ (see Section 7.1.1) and using the identity $\dot{R}^T = -R^T \dot{R} R^T$, we obtain

$$\dot{v} + \omega \times v = \dot{v} + \Omega v = (\dot{R}^T \dot{d} + R^T \ddot{d}) + R^T \dot{R} R^T \dot{d} = R^T \ddot{d} = 0,$$

which proves $\ddot{d} = 0$.

□

Remark 8.8 It is worth noting that the above result is independent of the choice of the scale factors α and β. The necessary conditions for minimum-acceleration and minimum-jerk curves derived in the subsequent subsections will also have the same property.

8.2.3 Minimum-acceleration curves

We follow the same route as in the previous subsection to compute curves that minimize the square of the L^2 norm of the acceleration. We derive the necessary conditions by considering the first variation of the minimum-acceleration functional

$$J_{\text{acc}} = \int_a^b < \nabla_V V, \nabla_V V > dt, \tag{8.24}$$

where $V(t) = \frac{dA(t)}{dt}$ and $A(t)$ is a curve on the manifold. The initial and final point as well as the initial and final velocity along the curve are prescribed.

Theorem 8.9 *Let $A(t)$ be a curve on a Riemannian manifold that satisfies the boundary conditions (that is, it starts and ends at the prescribed points with the prescribed velocities) and let $V = \frac{dA}{dt}$. If $A(t)$ minimizes the functional J_{acc}, then:*

$$\nabla_V \nabla_V \nabla_V V + R(V, \nabla_V V)V = 0. \tag{8.25}$$

Proof: See Appendix B.2. □

We can directly apply Theorem 8.9 to $SE(3)$ with the Riemannian connection computed from the metric (8.1).

Proposition 8.10 *Let*

$$A(t) = \begin{bmatrix} R(t) & d(t) \\ 0 & 1 \end{bmatrix}$$

be a curve between two prescribed points on $SE(3)$ that has prescribed initial and final velocities. If $\{\omega, v\}$ is the vector pair corresponding to $V = \frac{dA}{dt}$, the curve minimizes the cost functional J_{acc} derived from the metric (8.1) only if the following equations hold:

$$\begin{aligned} \omega^{(3)} + \omega \times \ddot{\omega} = 0 \\ d^{(4)} = 0, \end{aligned} \tag{8.26}$$

where $(\cdot)^{(n)}$ denotes the n^{th} derivative of $(\cdot)$.

Proof: We start by using Equations (8.15) and (8.16) to compute the second term in Equation (8.25):

$$\nabla_V \nabla_V \nabla_V V + R(V, \nabla_V V)V = \nabla_V \nabla_V \nabla_V V + \begin{bmatrix} \frac{1}{4}(\omega \times \dot{\omega}) \times \omega \\ 0 \end{bmatrix} = 0. \tag{8.27}$$

By repeated application of Equation (8.12) the term $\nabla_V \nabla_V \nabla_V V$ can be simplified and the rotational part of the above equation reduces to:

$$\left(\omega^{(3)} + \omega \times \ddot{\omega} + \frac{1}{4}\omega \times (\omega \times \dot{\omega})\right) + \left(\frac{1}{4}(\omega \times \dot{\omega}) \times \omega\right) = 0.$$

After some simplification we arrive at the first equation in (8.26). To simplify the translational component, we first observe that the translational component of $\nabla_V V$ can be written as (see the proof of Proposition 8.7):

$$\dot{v} + \omega \times v = R^T \ddot{d}.$$

It follows that the translational component of $\nabla_V \nabla_V V$ is:

$$\frac{d}{dt}(R^T \ddot{d}) + R^T \dot{R}(R^T \ddot{d}) = (\dot{R}^T \ddot{d} + R^T d^{(3)}) + R^T \dot{R} R^T \ddot{d} = R^T d^{(3)}.$$

Similarly, the translational component of $\nabla_V \nabla_V \nabla_V V$ can be simplified to get

$$R^T d^{(4)} = 0$$

from which the second equation in (8.26) directly follows. □

Remark 8.11 The above result is independent of the choice of the scale factors α and β. Further, the translational component of the necessary condition for minimum-acceleration does not depend on the angular motion, and the rotational component is independent of the translational motion (the latter is clear from the expressions (8.12) and (8.15)).

Remark 8.12 As observed in [101], the first equation (8.26) can be integrated to obtain

$$\omega^{(2)} + \omega \times \dot{\omega} = \text{constant} \tag{8.28}$$

However, this equation can not be further integrated analytically for arbitrary boundary conditions. In Section 8.3.2 we will show how to obtain the solution for a special choice of initial and final velocities.

8.2.4 Minimum-jerk curves

The minimum-jerk curves between two points are obtained by minimizing the square of L^2 norm of the Cartesian jerk, provided that the appropriate boundary conditions are given. In particular, it is possible to solve for minimum-jerk trajectories when the initial and final velocities and the initial and final accelerations are specified. Minimum-jerk trajectories are particularly useful in robotics where one is generally able to control the acceleration of the robot end effector (and therefore the velocity and the position) but the electro-mechanical actuators can not produce sudden changes in the acceleration. It is interesting to note that Flash and Hogan [42] suggest that humans plan trajectories that minimize such an integral measure of the jerk when reaching for an object in a horizontal plane and provide experimental evidence in support of this claim.

The minimum-jerk cost functional is:

$$J_{\text{jerk}} = \int_a^b < \nabla_V \nabla_V V, \nabla_V \nabla_V V > dt \tag{8.29}$$

where as usual $V = \frac{dA(t)}{dt}$. The curve must start and end at the desired points on the manifold and with the desired velocities and accelerations. We arrive at the necessary conditions for the solution by following the same approach as in the previous two subsections.

Theorem 8.13 *Let $A(t)$ be a curve on a Riemannian manifold that satisfies the boundary conditions (that is, it starts and ends at the prescribed points with the prescribed velocities and the prescribed accelerations) and let $V = \frac{dA}{dt}$. If $A(t)$ minimizes the functional J_{jerk}, then:*

$$\nabla_V^5 V + R(V, \nabla_V^3 V)V - R(\nabla_V V, \nabla_V^2 V)V = 0. \tag{8.30}$$

Proof: See Appendix B.3. □

The expressions for minimum-jerk trajectories on $SE(3)$ for the metric (8.1) immediately follow.

Proposition 8.14 *Let $A(t)$ be a curve between two prescribed points on $SE(3)$ that has prescribed initial and final velocities, and initial and final accelerations. If $\{\omega, v\}$ is the vector pair corresponding to $V = \frac{dA}{dt}$, the curve minimizes the cost functional J_{jerk} for the metric (8.1) only if the following equations hold:*

$$\begin{aligned}
&\omega^{(5)} + 2\,\omega \times \omega^{(4)} + \tfrac{5}{4}\omega \times (\omega \times \omega^{(3)}) + \tfrac{5}{2}\dot{\omega} \times \omega^{(3)} \\
&+\tfrac{1}{4}\omega \times (\omega \times (\omega \times \ddot{\omega})) + \tfrac{3}{2}\omega \times (\dot{\omega} \times \ddot{\omega}) - (\omega \times \ddot{\omega}) \times \dot{\omega} \\
&-\tfrac{1}{4}(\omega \times \dot{\omega}) \times \ddot{\omega} - \tfrac{3}{8}\omega \times ((\omega \times \dot{\omega}) \times \dot{\omega}) - \tfrac{1}{8}(\omega \times (\omega \times \dot{\omega})) \times \dot{\omega} \quad = \quad 0
\end{aligned}$$

$$d^{(6)} \;=\; 0. \tag{8.31}$$

Proof: The proof follows the same reasoning as the proof of Proposition 8.10 and use formulas (8.12) and (8.15) to evaluate the three terms in Equation (8.30). The first expression in Equation (8.31) was derived using Mathematica.

□

8.2.5 Minimum energy curves

In this subsection, we take the cost functional to be the kinetic energy of the rigid body and determine trajectories that minimize the integral of the kinetic energy over the entire trajectory. Critical points of this functional are the geodesics for the metric (8.2). According to Hamilton's principle in rigid body dynamics, for holonomic systems the integral of the Lagrangian over the entire trajectory is stationary [118]. Therefore, the minimization of the integral of the energy yields a trajectory consistent with the dynamic equations of motion when no external forces act on the body.

Proposition 8.15 *A curve $A(t)$ is a geodesic on $SE(3)$ equipped with the metric (8.2) if and only if the vector pair $\{\omega, v\}$ corresponding to the velocity vector field $V = \frac{dA}{dt}$ satisfies the equations:*

$$\begin{aligned}
\frac{d\omega}{dt} &= -H^{-1}(\omega \times (H\omega)) \\
\frac{dv}{dt} &= -\omega \times v.
\end{aligned} \tag{8.32}$$

The second equation in (8.32) can be simplified to the equation:

$$\ddot{d} = 0.$$

Proof: A curve $A(t)$ is a geodesic if and only if Equation (8.22) is satisfied. Substituting for $\nabla_V V$ from Equation (8.13), and letting $\{\omega_x, v_x\} = \{\omega_y, v_y\} = \{\omega, v\}$ we get Equation (8.32). Since the expression for the translational part is the same as that in Equation (8.23), to prove the second part we proceed in the same way as in the proof of Proposition 8.7.

□

8.3 Analytical expressions for optimal trajectories

8.3.1 Shortest distance path on $SE(3)$

Using properties of the Riemannian covering maps, Park showed [105] that for the metric (8.1) the geodesics on $SE(3)$ can be obtained by lifting the geodesics from $SO(3)$ and $\mathbb{R}^3$. We come to the same result constructively using Equation (8.23).

Proposition 8.16 *Given two configurations*

$$A_1 = \begin{bmatrix} R_1 & d_1 \\ 0 & 1 \end{bmatrix} \qquad A_2 = \begin{bmatrix} R_2 & d_2 \\ 0 & 1 \end{bmatrix}$$

the shortest distance path (minimal geodesic)

$$A(t) = \begin{bmatrix} R(t) & d(t) \\ 0 & 1 \end{bmatrix}$$

between these configurations with respect to the metric (8.1) is given by

$$R(t) = R_1 \exp(\Omega_0 t) \tag{8.33}$$
$$d(t) = (d_2 - d_1)\, t + d_1, \tag{8.34}$$

where

$$\Omega_0 = \log(R_1^T R_2).$$

(See Appendix F for the definition of the matrix log *function.) The path is unique unless* $\mathrm{Trace}(R_1^T R_2) = -1$ *when there exist two geodesics of equal minimum length (see Remark 8.17).*

Proof: The result follows from Proposition 8.7. The first equation in (8.23) can be readily integrated to obtain

$$\omega(t) = \omega_0. \tag{8.35}$$

Let Ω be the matrix equivalent of the vector ω. From Equation (7.3) we have:

$$\Omega = R^T \dot{R}. \tag{8.36}$$

Equation (8.35) can be thus integrated:

$$R^T \dot{R} = \Omega_0 \Rightarrow R(t) = R_0 \exp(\Omega_0 t). \tag{8.37}$$

From the initial condition we get $R_0 = R_1$ and from the boundary condition $\Omega_0 = \log(R_1^T R_2)$.

The expression for the vector $d(t)$ is obtained by integrating the equation $\ddot{d} = 0$ twice. As a result, we get:

$$d(t) = c_1 t + c_0$$

and Equation (8.34) immediately follows from the initial and final conditions on d. □

Remark 8.17 The log function on $SO(3)$ is multi-valued. If $\log(R)$ yields a solution (u, θ), where u is a unit vector along the axis of rotation and θ is the angle of rotation, then for any integer k, $(u, \theta + 2k\pi)$ is also a solution. This indeterminacy is resolved partially by restricting θ to lie in the interval $[0, \pi]$ (the interval $[-\pi, 0]$ is covered by using the axis $-u$). The geodesic computed by restricting θ to lie in the interval $[0, \pi]$ can be shown to give the unique minimal-length geodesic [105] unless $\theta = \pi$. If we think of $SO(3)$ as a unit hyper-sphere in $\mathbb{R}^4$ with antipodal points identified, the minimal-length geodesic is unique between any two general points, R_1 and R_2, except when $\text{Trace}(R_1^T R_2) = -1$ and there exist two geodesics of equal minimum length.

The above result suggests that the traveled distance between two configurations of the rigid body will be minimal only if the point O' on the rigid body translates along a straight line with a constant linear velocity while rotating along a fixed axis with a constant angular velocity. The geodesic is therefore obtained by combining geodesics on $SO(3)$ and $\mathbb{R}^3$, corroborating the result in [105].

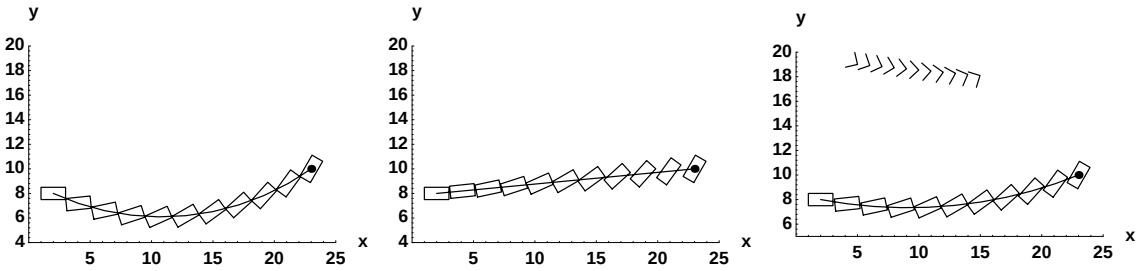


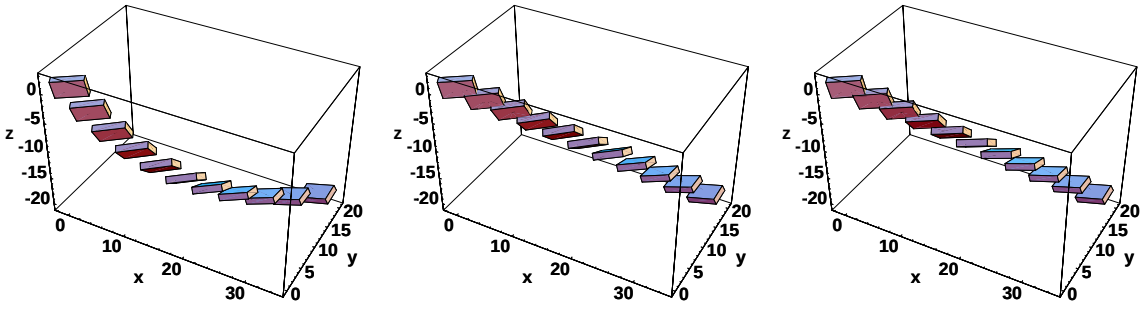
Figure 8.2: Motions in a plane: (a) a screw motion; (b) a geodesic for metrics (8.1) and (8.2); and (c) a geodesic after the body-fixed frame {M} is displaced.

Figure 8.2 shows various trajectories for a motion in the plane $z = 0$. Figure 8.2a shows a screw motion which in the planar case corresponds to rotation about a fixed point in the plane. A geodesic for the metric (8.1) is shown in Figure 8.2b. For planar motions, the geodesic for the metric (8.2) will be the same. Since the trajectory is computed by using a left invariant metric, it does not change if the inertial reference frame {F} is moved. But the trajectory changes if we change the body-fixed frame {M}. The trajectory is shown in Figure 8.2c and is different from the motion in Figure 8.2b. We also show the motion of the new body-fixed frame. The figure clearly shows that the new body-fixed frame follows a geodesic for metric (8.1).

It is interesting to compare the shortest distance trajectory to the trajectory that minimizes the kinetic energy. According to Hamilton's principle, the trajectory that minimizes the kinetic energy is obtained by solving the dynamic equations of motion. We stated the necessary conditions for this trajectory in Equation (8.32). The rotational part of Equation (8.32) are the well known Euler equations:

$$H\dot{\omega} + \omega \times H\omega = 0,$$

which, in general, do not admit an analytical solution. However, it is trivial to compute the translational trajectory of the rigid body: The point O' (which was chosen to coincide



a) b) c)

Figure 8.3: Trajectory of an object following: (a) a screw motion; (b) a geodesic for the metric (8.1); and (c) a geodesic for the metric (8.2).

with the center of mass) travels in a straight line with a uniform velocity between the initial and final position of the rigid body. This is true for an arbitrary choice of the positive-definite matrix H and as a special case contains the solution for the metric (8.1) obtained in Equations (8.33) and (8.34). The result is also the geometric statement of Newton's second law: The center of mass of the rigid body (the origin O') travels in a straight line if no external force acts on the body.

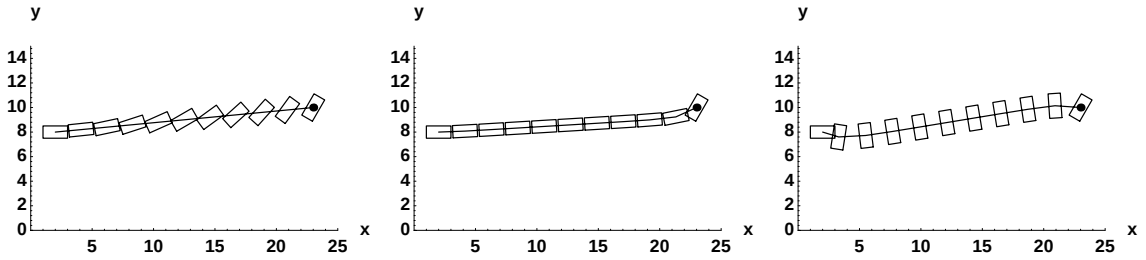
Figures 8.3.(a), (b) and (c) compare the screw motion, the shortest distance trajectory given by Equations (8.33) and (8.34), and the trajectory consistent with the dynamic equations (8.32) for a homogeneous rectangular prism with sides 5, 1 and 3 units between two given configurations. Figure 8.3.(a) shows the screw motion of the rigid body between the two configurations. It can be seen that the object rotates about a fixed axis while translating along the same axis. Figure 8.3.(b) shows the motion along the shortest distance path (the geodesic) given by Equations (8.33) and (8.34). The center of mass of the object moves along the straight line connecting the initial and final position while the body rotates around the center of mass as during the screw motion. Figure 8.3.(c) shows the trajectory that the rigid body would follow according to the principles of rigid body dynamics if it were launched with the appropriate initial velocity and not subjected to external forces. This is a geodesic for the metric defined in Equation (8.2). The center of mass again travels along the straight line, but the body rotation is governed by the Euler (dynamic) equations. The trajectory was computed numerically by solving a two-point boundary value problem [115]. Note that the shape and mass distribution of the rigid body are only relevant to the trajectory in Figure 8.3.(c).

Figure 8.4 shows geodesics on $SE(2)$ for left-invariant metrics of the form:

$$W = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \beta_1 & 0 \\ 0 & 0 & \beta_2 \end{bmatrix}. \tag{8.38}$$

The rows correspond to components ω_z, v_x and v_y, respectively. When $\beta_1 \neq \beta_2$, the metric of this form is not a product metric. Figure 8.4a is the same as Figure 8.2b and shows a geodesic for the product metric, when $\beta_1 = \beta_2$. The geodesic is a product of geodesics

on $S(1)$ and $\mathbb{R}^2$. The other two figures show geodesics for the cases when $\beta_1 \neq \beta_2$ and the metrics are not product metrics. In this case the rotational and the translational components of motion are coupled. In particular, the translational motion does not follow a straight line. These geodesics were computed numerically.



a) b) c)

Figure 8.4: Geodesics for different metrics on $SE(2)$: (a) product metric, $\beta_1 = \beta_2 = 1$; (b) a metric with $\beta_1 = 1$, $\beta_2 = 10$; and (c) a metric with $\beta_1 = 5$, $\beta_2 = 1$.

8.3.2 Minimum-acceleration and minimum-jerk trajectories

In general, the rotational components of the minimum-acceleration curves (Equation 8.26) and minimum-jerk curves (Equation 8.31) can not be computed analytically. However, when the initial velocities and accelerations are collinear with the initial velocity of the geodesic between the two endpoints, and the final velocities and accelerations are collinear with the final velocity of the geodesic, it is easy to obtain a solution for these trajectories in terms of the geodesic curve. If the geodesic curve can be computed analytically, so can minimum-acceleration and minimum-jerk curves. The results in this section are valid for any geodesically complete manifold.

Proposition 8.18 *Given an initial point q_0 and a final point q_1 on a Riemannian manifold Σ, let $\gamma : [0,1] \to \Sigma$ be a geodesic connecting these two points so that $\gamma(0) = q_0$ and $\gamma(1) = q_1$. Let $V_0 = \frac{d\gamma}{dt}\Big|_{t_0}$ and $V_1 = \frac{d\gamma}{dt}\Big|_{t_1}$. If the boundary conditions for the minimum-acceleration curve are of the form:*

$$V(t_0) = \eta_1 V_0 \qquad\qquad V(t_1) = \rho_1 V_1, \tag{8.39}$$

then the minimum-acceleration curve is given by $\gamma(p(t))$, where $p(t)$ is a third degree polynomial that satisfies:

$$\begin{array}{ll} p(0) = 0, & p(1) = 1 \\ p'(0) = \eta_1, & p'(1) = \rho_1, \end{array} \tag{8.40}$$

where $p' = \frac{dp}{dt}$.

Proof: Assume that the minimum-acceleration curve α has the form $\alpha(t) = \gamma(p(t))$, where p is an arbitrary scalar function, $p : \mathbb{R} \to \mathbb{R}$. It is easy to see that $V = \frac{d\alpha}{dt} = p' \frac{d\gamma}{dt}$. Let $T = \frac{d\gamma}{dt}$. Since γ is a geodesic, $\nabla_T T = 0$. It then follows:

$$\nabla_V V = V(p')T + p'p'\nabla_T T = V(p')T. \tag{8.41}$$

But $V(p')$ is a derivative of p' along α, so $V(p') = p''$. It immediately follows that:

$$\nabla_V^n V = p^{(n+1)} T. \tag{8.42}$$

Using the linearity of the curvature, we also get:

$$R(V, \nabla_V V) = R(p'T, p''T) = p'p''R(T,T) = 0. \tag{8.43}$$

Equation (8.31) therefore reduces to:

$$p^{(4)} T = 0, \tag{8.44}$$

and since T is a tangent vector for a geodesic and therefore never vanishes, we must have:

$$p^{(4)} = 0. \tag{8.45}$$

Solution of this differential equation is a polynomial of degree 3 and the boundary conditions transform into Equation (8.40). □

An analogous argument can be used to prove the following proposition:

Proposition 8.19 *Given an initial point q_0 and a final point q_1 on a Riemannian manifold, let $\gamma : [0,1] \to \Sigma$ be a geodesic connecting these two points so that $\gamma(0) = q_0$ and $\gamma(1) = q_1$. Let $V_0 = \frac{d\gamma}{dt}\Big|_{t_0}$ and $V_1 = \frac{d\gamma}{dt}\Big|_{t_1}$. If the boundary conditions for the minimum-jerk curve are of the form:*

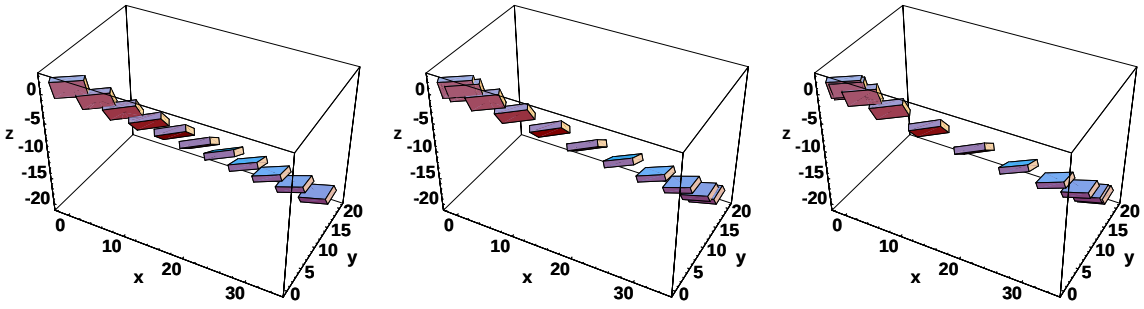
$$\begin{aligned} V(t_0) &= \eta_1 V_0, & V(t_1) &= \rho_1 V_1, \\ \nabla_V V|_{t_0} &= \eta_2 V_0, & \nabla_V V|_{t_1} &= \rho_2 V_1, \end{aligned} \tag{8.46}$$

then the minimum-jerk curve is given by $\gamma(p(t))$, where $p(t)$ is a fifth degree polynomial that satisfies:

$$\begin{aligned} p(0) &= 0, & p(1) &= 1 \\ p'(0) &= \eta_1, & p'(1) &= \rho_1, \\ p''(0) &= \eta_2, & p''(1) &= \rho_2. \end{aligned} \tag{8.47}$$

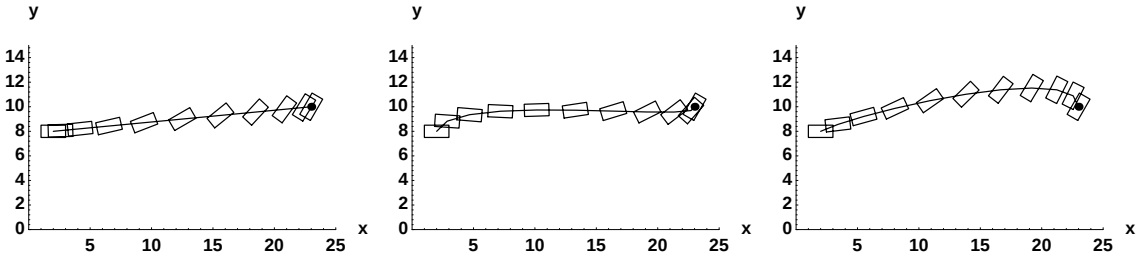
For this special form of the boundary conditions, the minimum-acceleration, minimum-jerk and minimum-distance paths are therefore the same, only the parameterization along the path varies. The shortest distance geodesic, the minimum-acceleration trajectory and the minimum-jerk trajectory when $\eta_1 = \eta_2 = \rho_1 = \rho_2 = 0$ and for $SE(3)$ equipped with the metric (8.1) are illustrated in Figure 8.5. The path followed by the rigid body in all three cases is the same. The minimum-acceleration trajectory starts and ends with zero velocity, so the velocity starts from zero, rises to a peak and then decreases to zero. The rise to peak and the fall to zero are steeper in the minimum-jerk trajectory because the starting and ending accelerations are also zero.

Figure 8.6 shows that for more general boundary conditions the path of the minimum-acceleration curve does not follow a geodesic. Further, the path changes with the boundary conditions. The figure shows minimum-acceleration motions in plane for different choices of the initial and final velocities, again for $SE(3)$ equipped with the metric (8.1). In Figure



a) b) c)

Figure 8.5: Trajectories for zero boundary conditions: (a) the shortest distance path; (b) the minimum-acceleration motion; and (c) the minimum-jerk motion.

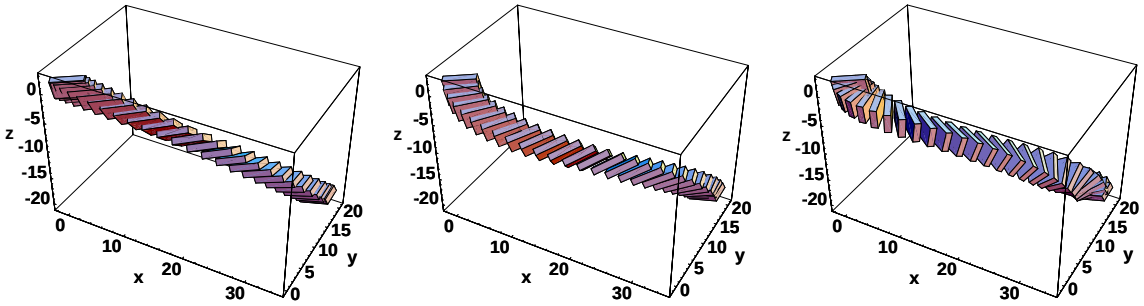


a) b) c)

Figure 8.6: Minimum-acceleration motions in plane for general boundary conditions: (a) $V(0) = V(1) = \{0,0,0\}^T$; (b) $V(0) = \{-1,3,10\}^T$, $V(1) = \{2,2,5\}^T$; and (c) $V(0) = \{1,10,5\}^T$, $V(1) = \{-1,-10,-5\}^T$.

8.6a, the initial and final velocities are 0, so the object follows the geodesic path shown in Figure 8.2b, but with a different velocity profile. The initial and final velocities for Figs. 8.6b,c are not collinear with the initial and final velocities of the geodesic in the figure 8.2b and it is clear the paths are different from the previous case. On the figure, only the components of the velocities in the plane are given, $V = \{\omega_z, v_x, v_y\}$.

The next figure, 8.7, shows three-dimensional minimum-acceleration motions for general boundary conditions. Similarly to the previous example, minimum-acceleration curves do not follow a geodesic. The figure also demonstrates that the trajectories change considerably with boundary conditions. Because of the product structure of the metric 8.1 which was used to compute the trajectories, the rotational motion is independent from the translational motion. However, the translational component will depend on the rotational component. Boundary conditions for the motion in Figure 8.7a are zero, so the motion follows the geodesic path. The motion in Figure 8.7b starts and ends with large translational velocity and small rotational velocity. For the motion in Figure 8.7c, the translational velocities at the initial and final points are smaller, but the rotational velocity is large.

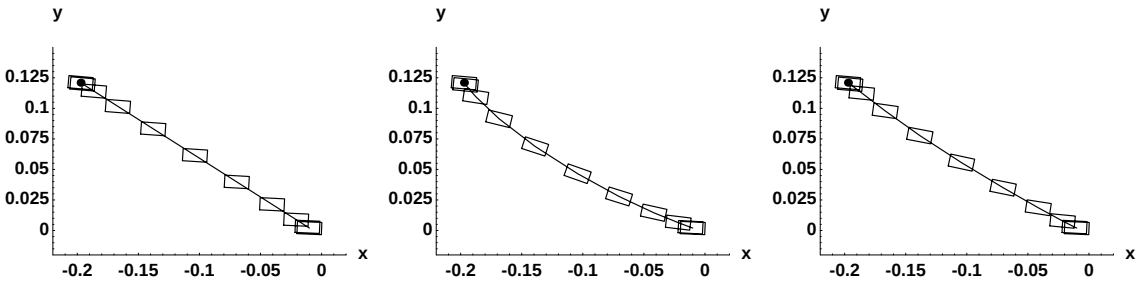


a) b) c)

Figure 8.7: Minimum-acceleration motions in space: (a) $V(0) = V(1) = \{0, 0, 0, 0, 0, 0\}^T$; (b) $V(0) = \{0, 0, 2, 0, -20, -20\}^T$, $V(1) = \{0, -2, 0, 0, -10, 0\}^T$; and (c) $V(0) = \{0, 0, 20, 0, -10, -10\}^T$, $V(1) = \{0, -20, 0, 0, -10, 0\}^T$.

8.3.3 Comparison between kinematic and dynamic motion planning

We conclude the chapter with comparison between the motion plans computed with kinematic and dynamic motion planning. Figure 8.8 shows a minimum-jerk trajectory and minimum torque-change trajectories that were computed using the model from Chapter 4. In all three cases, the initial and final velocities and accelerations were set to 0. Minimum-jerk trajectory, shown in Figure 8.8a, does not depend on the mechanism to which the object (end-effector) is attached and follows a straight line. For the minimum torque-change trajectory in Figure 8.8b, we did not impose any boundary conditions on the preload forces. The translational trajectory of the object in this case is quite curved and the object also rotates differently than in the previous case. For the trajectory in Figure 8.8c, the initial preload force was $2N$, while the final preload force was not specified. The translational motion of the object is now closer to the straight line, although it is still slightly curved. Also the rotation of the object is similar to the minimum-jerk motion (see also Figure 4.3).



a) b) c)

Figure 8.8: Kinematic *vs.* dynamic motion planning: (a) minimum-jerk; (b) minimum torque-change (preload forces not specified); and (c) minimum torque-change (initial preload force $2N$).

8.4 Discussion

This chapter addressed the problem of generating smooth trajectories for a rigid body between an initial and a final position and orientation. The main idea was to define a functional measuring the smoothness of a trajectory and find a trajectory that minimizes this cost functional. Using differential geometry, the problem was formulated as a variational problem on the Lie group of rigid body displacements, $SE(3)$. We defined an inner product on the Lie algebra $se(3)$ leading to a left invariant Riemannian metric on $SE(3)$. This metric gave rise to a Riemannian connection and a covariant derivative. We derived analytical expressions for the covariant derivative and the curvature of $SE(3)$. The covariant derivative was used to define acceleration and jerk for spatial rigid body motions. We stated the necessary conditions for minimum-distance, minimum-acceleration and minimum-jerk trajectories and specialized these conditions for $SE(3)$. We computed the analytical solutions for minimum-distance trajectories by choosing an appropriate basis for the space of the vector fields. We also found analytical solutions for the minimum-acceleration and minimum-jerk trajectories for a special class of boundary conditions. We provided several numerical examples to illustrate how the generated solutions are affected by (a) the metric; (b) the choice of the body-fixed reference frame; and (c) the boundary conditions. A simple extension of the ideas in this chapter allows the inclusion of intermediate positions and orientations and lends itself to motion interpolation (see [25]). The presented methods also have applications in computer graphics and computer-aided design.

Chapter 9

Concluding remarks

One of the advantages of robots over fixed automation is that they are versatile and can be programmed for a variety of tasks. But to make a robot sufficiently dextrous to perform different tasks, it must be equipped with multiple degrees of freedom. Coordination of these degrees of freedom requires motion planning.

The problem of finding a suitable motion for a given task is usually under-determined. We argued that in such cases it makes sense to define the performance of a motion for the task so that different motions can be compared. This enabled us to treat motion planning as a variational problem where the motion plan is obtained by maximizing the chosen measure of performance. This method is continuous in the sense that the motion planning problem is not discretized. To further motivate the chosen approach we presented an investigation of human reaching motions. The study suggested that trajectories used by humans minimize a certain cost functional. We defined dynamic and kinematic motion planning: Dynamic motion planning is used when the actuator forces must be provided as part of the motion plan; kinematic motion planning is simpler and is used when it suffices to compute the kinematic trajectory (in the task space or joint space).

The method that we proposed for dynamic motion planning consists of finding a motion that satisfies the dynamic equations of the system and maximizes the cost functional measuring the performance of the motion for the task. Limits on the capabilities of the system and task requirements restrict the set of feasible motions. The resulting set can in general be described with equality and inequality constraints. To compute a motion plan, we developed a novel numerical method for minimizing a given cost functional subject to equality and inequality constraints. The method was used to find smooth trajectories and actuator forces for two planar cooperating manipulators holding an object. It was then extended for systems that change the dynamic equations as they move. The example of a simple grasping task illustrated that for such systems variational approach unifies motion planning and task planning.

An important feature of the proposed method is that planning in the task space, joint space, and actuator space is performed in one step. In this way, constraints at all three levels can be incorporated into the motion planning process. Further, kinematic motion planning is not separated from planning the actuator trajectory. Since it is in general necessary to plan the kinematic as well as the actuator trajectory, finding the trajectories separately is unnatural and can result in poor motion plans. Finally, motion plans obtained

with the variational methods will globally maximize performance and will thus be superior to motion plans computed with other methods.

Kinematic motion planning is simpler and more efficient than dynamic motion planning. It is thus preferred to dynamic motion planning if there exists a suitable way to compute the actuator forces from the kinematic trajectory. Kinematic motion planning is also the only feasible alternative when a dynamic model of the system is too complicated or is not available and is abstracted with a kinematic model. We formulated kinematic motion planning as a variational problem on a Riemannian manifold. In this way, the computed trajectories do not depend on the description of the space. Using an affine connection, we defined the notions of acceleration and jerk on an arbitrary manifold. The affine connection can be chosen arbitrarily which implies that there is no intrinsic notion of acceleration and jerk. We also showed that a Riemannian metric must be introduced on the manifold to define cost functionals that measure the smoothness of a curve.

The proposed method for kinematic motion planning can be used either in the task space or in the joint space. However, the task space has more structure and we take advantage of this structure to find a Riemannian metric. In most cases, the task space is the group of spatial rigid body motions, $SE(3)$. We used the group structure of $SE(3)$ to define metrics that have certain invariance properties with respect to the choice of the coordinate frames describing the motion in the task space. In this process we also identified metrics and connections on $SE(3)$ that are useful for kinematic analysis.

By choosing the class of explicit methods for motion planning we assumed that controllers to follow the computed trajectories can be synthesized. If the system is controllable, such controllers can readily be obtained. We also assumed that we have a complete, deterministic model of the system so that variational methods can be used. Such models are available in many robotic applications so this assumption is not overly restrictive.

9.1 Summary

In the first chapter we reviewed the literature on motion planning in humans and existing approaches to motion planning for robotic systems. The main deficiency of most of these methods is their inability to provide a set of actuator forces that are consistent with the constraints imposed by the task and limited capabilities of the robot. This motivated formulating the motion planning problem as a variational calculus problem. In the next chapter, we discussed optimal control and its relation to variational calculus. We showed that when the constraints on the controls can be described with equalities and inequalities, the optimal control problem can be transformed into a problem of Bolza in the calculus of variations. We also constructed an unconstrained variational problem that has the same set of extremals as the problem of Bolza.

Equality and inequality constraints will be present in any realistic motion planning problem. The most important criterion in choosing the numerical method for solving the motion planning problem formulated as a problem of Bolza was therefore how well the method handles such constraints. As the inequality constraints lead to corner points, most of the methods based on the minimum principle or Euler-Lagrange equations are difficult to apply. In Chapter 3 we presented two numerical methods that can deal with inequality constraints. The first method was proposed by Gregory and Lin [49]. This method avoids

problems with corners by using the integral form of the necessary conditions. By a procedure that resembles finite-element analysis, these necessary conditions are transformed into a set of nonlinear algebraic equations in the values of the unknown functions at discrete mesh points. We provided a detailed analysis of this method for the case of equality and inequality constraints that only depend on the state. We showed that state-variable inequality constraints lead to a singular problem which means that the numerical solutions can exhibit oscillations. We next described a new method for finding a (local) minimum for the problem of Bolza. The method was based on approximating the continuous problem with a sequence of finite-dimensional nonlinear programming problems through collocation. We described the method of augmented Lagrangian for dealing with the constraints of a nonlinear programming problem and Newton's method for minimization. We also showed that the choice of collocating functions motivated by finite-element analysis leads to simple expressions for the gradients that are needed for minimization of the cost functional. We next discussed the choice of a cost functional for the motion planning problem. We showed that depending on the cost functional we get a kinematic or a dynamic motion planning method. We concluded the chapter with a discussion on cost functionals that can be used to obtain smooth motion plans.

The numerical method of Gregory and Lin was used in Chapter 4 to compute smooth motion plans for two cooperating manipulators holding an object. Cooperative tasks are characterized by closed kinematic chains and typically result in actuator redundancy. Based on the consideration of smoothness requirements, we chose to plan trajectories that minimize the integral of the square norm of derivatives of actuator forces. In this way, it is possible to satisfy boundary conditions on the acceleration and obtain a smooth transition from one motion to another. We solved the motion planning problem when the object was grasped rigidly and when a friction-assisted grasp was used to manipulate the object. When a friction-assisted grasp is used, the forces must satisfy frictional constraints. Using the chosen numerical method, we computed typical motion plans for both cases. The numerical method proved to be quite robust and efficient. Results from this chapter are reported in [156] and [162].

In Chapter 5, we showed that principles used for robot motion planning can serve as a model for motion planning in biological systems. We presented experimental measurements of human two-arm trajectories. A qualitative analysis of these data was performed with special focus on the internal forces. These forces indicate what coordination processes take place between the two arms during the cooperative task. We were not able to obtain a consistent pattern except when the motion took place in the sagittal plane and the task was symmetric for both arms. In this case, the kinematic features of the measured trajectories and the general shape of the interaction forces were well modeled with the trajectories that minimize the minimum torque-change criterion. This work is described in [45] and [161].

The next chapter introduced motion planning problem for systems that change the dynamic equations as they move. In robotics, examples of such systems are a multifingered hand manipulating an object or a legged machine walking on the ground. We showed that the motion planning problem for such systems is quite complex. However, the variational formulation of the problem unifies motion planning and task planning. We developed a technique for computing motion plans when the sequence of discrete states that the system traverses is known in advance. In such case, the planning task is to determine continuous trajectories for the system and the optimal times when the system should switch from one

discrete state to another. We argued that the sequence of discrete states can be computed in advance in some important robotic tasks. The method was illustrated by computing motion plans for a two-fingered hand rotating a circular object. In this case it was essential to use the newly developed numerical method from Chapter 3 to obtain the solution. The main ideas from this chapter are reported in [155].

The next two chapters were devoted to planning smooth trajectories in the task space. Smooth trajectory in the task space is required for example to move through a sequence of points that were recorded by a teaching process in the task space. Task space can be represented with the special Euclidean group $SE(3)$, hence we started by investigating the geometric properties of this group. In particular, we studied choices of metrics and connections on $SE(3)$. We established that there is no Riemannian metric which has screw motions for geodesics. We next identified a symmetric connection which leads to kinematically meaningful acceleration and found a family of Riemannian metrics which are compatible with this connection. Publications describing the work in this chapter are [157] and [160].

We developed a method for computing smooth trajectories on the task space manifold in Chapter 8. We argued that by using a geometric framework, trajectories that are independent of the representation of the task space can be obtained. Consequently, the trajectory planning problem was formulated as a variational problem on a Riemannian manifold. We showed that a Riemannian metric (which also gives rise to a unique connection) is necessary to define the notions of the acceleration and jerk. Results from the previous chapter were used to find the necessary conditions for trajectories that minimize energy, square of the L^2 norm of the acceleration and square of the L^2 norm of the jerk on $SE(3)$. Analytical expressions for the trajectories were derived for some special cases. We also provided several numerical examples to illustrate how the generated trajectories are affected by the metric, the choice of the body-fixed reference frame, and the boundary conditions. The results from this chapter are reported in [158] and [159].

9.2 Contributions

Contributions of this dissertation fall into several areas:

- We propose a unified framework for motion planning. The underlying idea is to use the methods of calculus of variations. In this framework, trajectories in the task space, joint space and actuator space are obtained in a single step while simultaneously taking into account limitations of robotic mechanism and task constraints. The validity of the approach is demonstrated by solving several complex motion planning problems.

- Central to the application of the methods of variational calculus is an efficient numerical method. We develop a new numerical method for solving variational problems using the ideas from nonlinear programming and finite-element analysis. An important feature of the method is that the gradient of the cost functional is easy to compute. As a result, the method is efficient and easy to implement. We also present a detailed analysis of the numerical method developed by Gregory and Lin [49] and suggest a number of improvements.

- A technique for solving variational problems when the dynamical system goes through a known sequence of transitions between regions representing different dynamic behavior is derived. Previous methods used iteration on the unknown switching times, requiring a solution of an optimal control problem at each iteration. Instead, we solve the optimal control problem only once by introducing additional state variables. This method is general and does not depend on the numerical method used for solving the optimal control problem.

- By describing tasks such grasping and walking as a system that changes dynamic equations when it moves, we are able to unify motion planning and task planning. We therefore establish a link between motion planning methods and algorithms for task planning. While we do not have a complete solution for the generalized motion planning problem, our formulation provides a framework for further investigation of this problem.

- We present experimental investigation of human two-arm manipulation and propose the minimum torque-change model to account for the observed trajectories. This is the first attempt (see also [44]) to model trajectory generation for two-arm motions. While the proposed model has considerable shortcomings, it predicts kinematic features of the trajectories and force profiles for sagittal plane motions, which are characterized by a symmetric role of the two arms. The study suggests that the role of the two arms in two-arm reaching tasks is not symmetric. This observation can serve to formulate a criterion that better describes human motion planning.

- We develop a practical method for task space trajectory planning using the geometry of the task space. Most other methods assume Euclidean structure of the task space for computations or are limited to theoretical results and hard to implement. We provide analytical solutions for optimal trajectories on $SE(3)$ for some special but important cases. The trajectories generated by the proposed method are independent of the description of the task space. We also show how some further invariant properties can be obtained by an appropriate choice of a metric for $SE(3)$.

- We attach geometric meaning to several concepts from kinematics and present some new results regarding the choice of metrics and connections for kinematic analysis. In particular, we prove that there is no Riemannian metric for which screw motions are geodesics. Further, we show that an affine connection must be chosen to define the acceleration and identify the connection that produces a physically meaningful acceleration.

9.3 Possible applications and future work

While we have made considerable progress in applying variational methods to kinematic and dynamic motion planning for robotic systems, the work opened several avenues for future research. Traditional approaches to motion planning are based on discretized representation of the configuration manifold and graph search techniques. In contrast, the basis for many continuous trajectory planning methods are optimization techniques. Further advances in motion planning can be expected if partial solutions of the continuous

problem are used as building blocks for the discrete method. In this way, it is possible to construct quasi-optimal solutions. By employing optimization methods for local planning there is also no need for constructive controllability algorithms that are difficult to derive for systems with constraints.

In the dissertation we have not addressed the problem of tracking the trajectories produced by the planning process. While the results in the cases when the system is controllable are well known in control literature, such methods fail for systems that are subject to unilateral constraints and therefore need further investigation.

Design and control of hybrid systems, where continuous and discrete behaviors coexist, is a rapidly growing area in control. The optimal control method developed in the dissertation represents an attempt to design open-loop trajectories for a subset of hybrid systems where the discrete behavior corresponds to switches between regions of the state space characterized by different sets of dynamic equations. When the discrete switches are controlled by another process, a problem becomes one in game theory and the optimal control method must be appropriately extended. This extension seems feasible since game theory is a natural extension of optimal control. Manufacturing, control of interaction between a human operator and a virtual or a mechanical environment and coordination of multiple agents are some possible applications where such method could be used.

Another area to which this work could be extended is modeling and control of mechanical systems subject to unilateral and nonholonomic constraints. Nonholonomic constraints can be relatively easily handled in optimal control and numerical solutions for optimal trajectories can be readily computed. Theoretically, such variational problems are studied in sub-Riemannian geometry and applications of this theory to control are still rare. Through proper extension of Hamilton's principle, variational calculus also provides an alternative to modeling dynamical systems subject to inequality constraints, that appear, for example, in contact mechanics.

Study of metrics and affine connections on the group of rigid body motions $SE(3)$ provides some new insights in the theory of screws and its use in kinematics. These results can be further formalized and have consequences for the formulation of the position/force control schemes. An important open question is also the existence of analytical solutions for minimum acceleration and minimum jerk trajectories. Such expressions would be of considerable value for applications.

Initial motivation for this research have been studies of human trajectory generation. Knowing how humans move can aid in developing motion planning algorithms. On the other hand, methods that were primarily designed for robot trajectory planning can also be used to model human motor behavior. The two areas therefore complement each other and it is important to further our understanding of human motor behavior. A possible area of application of this work is also computer animation where understanding of principles that govern human motion and good techniques for planning actuator forces are essential for generating a naturally looking animated figure.

Appendix A

Product metric in the basis $\{\hat{L}_i\}$

In this section we derive the expression in the basis $\{\hat{L}_i\}$ for the product metric on $SE(3)$, obtained by taking an arbitrary left-invariant metric Q on $SO(3)$ and an inner product metric on $\mathbb{R}^3$ given by a constant 3×3 positive-definite matrix W. If the basis $\hat{L}_1$, $\hat{L}_2$ and $\hat{L}_3$ is chosen for the vector fields on $SO(3)$ and the Euclidean basis E_4, E_5, E_6 for the vector fields on $\mathbb{R}^3$, the matrix G describing the metric in this basis has the form:

$$G_p = \begin{bmatrix} Q_{3\times 3} & 0 \\ 0 & W \end{bmatrix} \tag{A.1}$$

To compare metric (A.1) with (7.70), we have to express the vectors E_4, E_5, E_6 in the basis $\hat{L}_1, \ldots, \hat{L}_6$.

Take a point $A \in SE(3)$, where:

$$A = \begin{bmatrix} R & d \\ 0 & 1 \end{bmatrix}. \tag{A.2}$$

Note that A, as an element of $SO(3) \times \mathbb{R}^3$, is represented by a pair (R, d). Accordingly, $SE(3)$ is parameterized as a product manifold $SO(3) \times \mathbb{R}^3$ with the usual parameterization of $\mathbb{R}^3$ (any parameterization of $SO(3)$ can be chosen). Take a vector field:

$$X = X^4 \hat{L}_4 + X^5 \hat{L}_5 + X^6 \hat{L}_6 = v^T \begin{bmatrix} \hat{L}_4 \\ \hat{L}_5 \\ \hat{L}_6 \end{bmatrix}, \tag{A.3}$$

where the components X^4, X^5 and X^6 are constant and $v = \{X^4, X^5, X^6\}^T$. The integral curve of this vector field passing through A is given by $\gamma(t) = A \exp(tS)$, where $S \in se(3)$ is the matrix representation of the vector $s = \{0, 0, 0, X^4, X^5, X^6\}$. It is easy to see that:

$$\gamma(t) = \begin{bmatrix} R & tR\, v + d \\ 0 & 1 \end{bmatrix}. \tag{A.4}$$

The tangent vector to a curve $\gamma(t) = \{\xi_1(t), \ldots, \xi_6(t)\}$ is given by:

$$\frac{d\gamma}{dt} = \frac{d\xi_i}{dt} \frac{\partial}{\partial \xi_i} \tag{A.5}$$

We first note that on the curve $\gamma(t)$, the rotational part is constant which means that the coefficients $\frac{d\xi_i}{dt}$ for $i = 1, 2, 3$ are all 0. Further, since the parameterization of $SE(3)$ is induced by the parameterizations of $SO(3)$ and $\mathbb{R}^3$, we have:

$$\begin{bmatrix} \xi_4 \\ \xi_5 \\ \xi_6 \end{bmatrix} = tR\ v + d. \tag{A.6}$$

The expression for the tangent vector $X(A)$ in the basis $E_i = \frac{\partial}{\partial \xi_i}$ is therefore:

$$v^T \begin{bmatrix} \hat{L}_4 \\ \hat{L}_5 \\ \hat{L}_6 \end{bmatrix} = (R\ v)^T \begin{bmatrix} E_4 \\ E_5 \\ E_6 \end{bmatrix} \tag{A.7}$$

Since the equation must be true for arbitrary v, we get:

$$\begin{bmatrix} \hat{L}_4 \\ \hat{L}_5 \\ \hat{L}_6 \end{bmatrix} = R^T \begin{bmatrix} E_4 \\ E_5 \\ E_6 \end{bmatrix} \tag{A.8}$$

The last expression also implies that change of the basis vector fields on $\mathbb{R}^3$ from E_4, E_5 and E_6 to $\hat{L}_4$, $\hat{L}_5$ and $\hat{L}_6$ will only change the lower-right block of matrix G. The entries in this block are obtained by:

$$\begin{bmatrix} \hat{L}_4 \\ \hat{L}_5 \\ \hat{L}_6 \end{bmatrix} \cdot \begin{bmatrix} \hat{L}_4 & \hat{L}_5 & \hat{L}_6 \end{bmatrix} = R^T \begin{bmatrix} E_4 \\ E_5 \\ E_6 \end{bmatrix} \cdot \begin{bmatrix} E_4 & E_5 & E_6 \end{bmatrix} R = R^T W R, \tag{A.9}$$

where $E_i \cdot E_j \stackrel{\text{def}}{=} < E_i, E_j >$.

Appendix B

Proofs

B.1 Proof of Proposition 8.6

In the proof we will use the following conventions:

$$X = X^i \hat{L}_i \stackrel{\text{def}}{=} \begin{bmatrix} \omega_x \\ v_x \end{bmatrix},$$

$$X(Y^i)\hat{L}_i \stackrel{\text{def}}{=} \begin{bmatrix} X(\omega_y) \\ X(v_y) \end{bmatrix},$$

and

$$[X,Y] \stackrel{\text{def}}{=} \begin{bmatrix} \omega_{[X,Y]} \\ v_{[X,Y]} \end{bmatrix}.$$

By definition, the Riemannian curvature is given by

$$R(X,Y)\,Z = \nabla_Y \nabla_X Z - \nabla_X \nabla_Y Z + \nabla_{[X,Y]} Z.$$

Using the expressions derived in Proposition 8.4, one obtains:

$$\begin{aligned}
R(X,Y)\,Z = {} & \nabla_Y \begin{bmatrix} X(\omega_z) + \frac{1}{2}\omega_x \times \omega_z \\ X(v_z) + \omega_x \times v_z \end{bmatrix} - \nabla_X \begin{bmatrix} Y(\omega_z) + \frac{1}{2}\omega_y \times \omega_z \\ Y(v_z) + \omega_y \times v_z \end{bmatrix} \\
& + \begin{bmatrix} [X,Y](\omega_z) + \frac{1}{2}\omega_{[X,Y]} \times \omega_z \\ [X,Y](v_z) + \omega_{[X,Y]} \times v_z \end{bmatrix} \\
= {} & \begin{bmatrix} YX(\omega_z) + \frac{1}{2}\left(Y(\omega_x) \times \omega_z + \omega_x \times Y(\omega_z) + \omega_y \times X(\omega_z)\right) + \frac{1}{4}\omega_y \times (\omega_x \times \omega_z) \\ YX(v_z) + Y(\omega_x) \times v_z + \omega_x \times Y(v_z) + \omega_y \times X(v_z) + \omega_y \times (\omega_x \times v_z) \end{bmatrix} \\
& - \begin{bmatrix} XY(\omega_z) + \frac{1}{2}\left(X(\omega_y) \times \omega_z + \omega_y \times X(\omega_z) + \omega_x \times Y(\omega_z)\right) + \frac{1}{4}\omega_x \times (\omega_y \times \omega_z) \\ XY(v_z) + X(\omega_y) \times v_z + \omega_y \times X(v_z) + \omega_x \times Y(v_z) + \omega_x \times (\omega_y \times v_z) \end{bmatrix} \\
& + \begin{bmatrix} [X,Y](\omega_z) + \frac{1}{2}(\omega_x \times \omega_y + X(\omega_y) - Y(\omega_x)) \times \omega_z \\ [X,Y](v_z) + (\omega_x \times \omega_y + X(\omega_y) - Y(\omega_x)) \times v_z \end{bmatrix} \\
= {} & \begin{bmatrix} \frac{1}{4}(\omega_x \times \omega_y) \times \omega_z \\ 0 \end{bmatrix}
\end{aligned} \tag{B.1}$$

Note that the final expression only depends on the values of the vector fields X, Y and Z and not on their derivatives. For this reason, the Riemannian curvature can be used to define the so called *curvature tensor* [34].

B.2 Proof of Theorem 8.9

The proof follows the same lines as the derivation of the geodesic equation (8.22). In addition to the identities (1)-(4) from Section 8.2.1 we will use an additional identity:

(5) $\nabla_S \nabla_T U = \nabla_T \nabla_S U + R(T,S)U$.

which is just the definition of the curvature operator when $[S,T] = 0$, and one of the symmetry properties of the curvature tensor

(6) $< R(X,Y)Z,T >=< R(Z,T)X,Y >$.

In the proof below, the numbers over the equal sign indicate which identities were employed.

$$
\begin{aligned}
\frac{1}{2}L_a'(s) &= \frac{1}{2}\frac{d}{ds}\int_a^b < \nabla_V V, \nabla_V V > dt \qquad \text{(B.2)}\\
&\overset{1}{=} \frac{1}{2} S \int_a^b < \nabla_V V, \nabla_V V > dt\\
&\overset{2}{=} \int_a^b < \nabla_S \nabla_V V, \nabla_V V > dt\\
&\overset{5}{=} \int_a^b < \nabla_V \nabla_S V + R(V,S)V, \nabla_V V > dt\\
&\overset{3,6}{=} \int_a^b (< \nabla_V \nabla_V S, \nabla_V V > + < R(V, \nabla_V V)V, S >) dt\\
&\overset{2}{=} \int_a^b (V < \nabla_V S, \nabla_V V > - < \nabla_V S, \nabla_V \nabla_V V > + < R(V, \nabla_V V)V, S >) dt\\
&\overset{4,2}{=} < \nabla_V S, \nabla_V V >|_a^b + \int_a^b (-V < S, \nabla_V \nabla_V V > + < S, \nabla_V \nabla_V \nabla_V V >\\
&\quad + < R(V, \nabla_V V)V, S >) dt\\
&\overset{4}{=} [< \nabla_V S, \nabla_V V > - < S, \nabla_V \nabla_V V >]_a^b\\
&\quad + \int_a^b (< S, \nabla_V \nabla_V \nabla_V V + R(V, \nabla_V V)V > dt
\end{aligned}
$$

Every curve of variation must satisfy the boundary conditions. Since positions and velocities are prescribed at $t = a$ and $t = b$, both S and $\nabla_S V = \nabla_V S$ vanish at the endpoints. This implies that the integrated part (the first two terms) in Equation (B.3) equals 0. Furthermore, on the critical point, the integral must vanish for any admissible variational field S, which means that

$$\nabla_V \nabla_V \nabla_V V + R(V, \nabla_V V)V = 0. \qquad \text{(B.3)}$$

B.3 Proof of Theorem 8.13

Once again we employ identities (1)-(4) from Section 8.2.1 and (5)-(6) from Section B.2. We first obtain the expression for the first variation of the functional L_j:

$$\begin{aligned}
\frac{1}{2}L_j'(s) &= \frac{1}{2}\frac{d}{ds}\int_a^b < \nabla_V^2 V, \nabla_V^2 V > dt \qquad \text{(B.4)}\\
&\overset{1}{=} \frac{1}{2}S\int_a^b < \nabla_V^2 V, \nabla_V^2 V > dt\\
&\overset{2}{=} \int_a^b < \nabla_S \nabla_V^2 V, \nabla_V^2 V > dt\\
&\overset{5}{=} \int_a^b (< \nabla_V \nabla_S \nabla_V V + R(V,S)\nabla_V V, \nabla_V^2 V >)\, dt\\
&\overset{2,6}{=} \int_a^b (V < \nabla_S \nabla_V V, \nabla_V^2 V > - < \nabla_S \nabla_V V, \nabla_V^3 V > + < R(\nabla_V V, \nabla_V^2 V)V, S >)\, dt\\
&\overset{4,5}{=} < \nabla_S \nabla_V V, \nabla_V^2 V >\Big|_a^b + \int_a^b (- < \nabla_V \nabla_S V + R(V,S)V, \nabla_V^3 V >\\
&\quad + < R(\nabla_V V, \nabla_V^2 V)V, S >)\, dt\\
&\overset{3}{=} < \nabla_S \nabla_V V, \nabla_V^2 V >\Big|_a^b + \int_a^b (- < \nabla_V^2 S, \nabla_V^3 V > - < R(V,S)V, \nabla_V^3 V >\\
&\quad + < R(\nabla_V V, \nabla_V^2 V)V, S >)\, dt\\
&\overset{2,6}{=} < \nabla_S \nabla_V V, \nabla_V^2 V >\Big|_a^b + \int_a^b (-V < \nabla_V S, \nabla_V^3 V > + < \nabla_V S, \nabla_V^4 V >\\
&\quad - < R(V, \nabla_V^3 V)V, S > + < R(\nabla_V V, \nabla_V^2 V)V, S >)\, dt\\
&\overset{4,5}{=} [< \nabla_V \nabla_S V + R(V,S)V, \nabla_V^2 V > - < \nabla_V S, \nabla_V^3 V >]_a^b\\
&\quad + \int_a^b (< \nabla_V S, \nabla_V^4 V > - < R(V, \nabla_V^3 V)V, S > + < R(\nabla_V V, \nabla_V^2 V)V, S >)\, dt\\
&\overset{2,3,6}{=} [< \nabla_V \nabla_V S, \nabla_V^2 V > + < R(V, \nabla_V^2 V)V, S > - < \nabla_V S, \nabla_V^3 V >]_a^b\\
&\quad + \int_a^b (V < S, \nabla_V^4 V > - < S, \nabla_V^5 V > + < -R(V, \nabla_V^3 V)V + R(\nabla_V V, \nabla_V^2 V)V, S >)\, dt\\
&\overset{4}{=} [< \nabla_V \nabla_V S, \nabla_V^2 V > + < R(V, \nabla_V^2 V)V, S > - < \nabla_V S, \nabla_V^3 V > + < S, \nabla_V^4 V >]_a^b\\
&\quad + \int_a^b (< [-\nabla_V^5 V - R(V, \nabla_V^3 V)V + R(\nabla_V V, \nabla_V^2 V)V], S >)\, dt
\end{aligned}$$

Since the initial and final positions, velocities and accelerations are fixed, S, $\nabla_S V = \nabla_V S$ and $\nabla_S \nabla_V V = \nabla_V \nabla_S V + R(V,S)V = \nabla_V \nabla_V S$ vanish at the endpoints. Thus the integral in the above equation must vanish for an arbitrary variation (that preserves the boundary conditions). But this is only possible if Equation (8.30) holds so the Theorem is proved.

Appendix C

Metric with screw motions as geodesics

In Section 7.2.1, we concluded that Equation (7.36):

$$\hat{L}_k(g_{ij}) = \frac{1}{2}\sum_l (C^l_{ki} g_{lj} + C^l_{kj} g_{li}), \tag{C.1}$$

must be satisfied by the metric if screw motions are geodesics. The coefficients C^k_{ij} are the structure constants of the Lie algebra $se(3)$ (see Appendix E). We evaluated this equation in Mathematica to obtain a system of 126 partial differential equations that have to be solved for the metric coefficients g_{ij}. In the equations, we use the abbreviation $\mathcal{G}^k_{ij} \stackrel{\text{def}}{=} \hat{L}_k(g_{ij})$.

$$\begin{array}{lll}
\mathcal{G}^1_{11} = 0 & \mathcal{G}^2_{11} = -g_{13} & \mathcal{G}^3_{11} = g_{12} \\
\mathcal{G}^4_{11} = 0 & \mathcal{G}^5_{11} = -g_{16} & \mathcal{G}^6_{11} = g_{15} \\
\mathcal{G}^1_{12} = \frac{1}{2} g_{13} & \mathcal{G}^2_{12} = -\frac{1}{2} g_{23} & \mathcal{G}^3_{12} = \frac{1}{2}(g_{22} - g_{11}) \\
\mathcal{G}^4_{12} = \frac{1}{2} g_{16} & \mathcal{G}^5_{12} = -\frac{1}{2} g_{26} & \mathcal{G}^6_{12} = \frac{1}{2}(g_{25} - g_{14}) \\
\mathcal{G}^1_{13} = -\frac{1}{2} g_{12} & \mathcal{G}^2_{13} = \frac{1}{2}(g_{11} - g_{33}) & \mathcal{G}^3_{13} = \frac{1}{2} g_{23} \\
\mathcal{G}^4_{13} = -\frac{1}{2} g_{15} & \mathcal{G}^5_{13} = \frac{1}{2}(g_{14} - g_{36}) & \mathcal{G}^6_{13} = \frac{1}{2} g_{35} \\
\mathcal{G}^1_{14} = 0 & \mathcal{G}^2_{14} = \frac{1}{2}(-g_{34} - g_{16}) & \mathcal{G}^3_{14} = \frac{1}{2}(g_{24} + g_{15}) \\
\mathcal{G}^4_{14} = 0 & \mathcal{G}^5_{14} = -\frac{1}{2} g_{46} & \mathcal{G}^6_{14} = \frac{1}{2} g_{45} \\
\mathcal{G}^1_{15} = \frac{1}{2} g_{16} & \mathcal{G}^2_{15} = -\frac{1}{2} g_{35} & \mathcal{G}^3_{15} = \frac{1}{2}(g_{25} - g_{14}) \\
\mathcal{G}^4_{15} = 0 & \mathcal{G}^5_{15} = -\frac{1}{2} g_{56} & \mathcal{G}^6_{15} = \frac{1}{2} g_{55} \\
\mathcal{G}^1_{16} = -\frac{1}{2} g_{15} & \mathcal{G}^2_{16} = \frac{1}{2}(g_{14} - g_{36}) & \mathcal{G}^3_{16} = \frac{1}{2} g_{26} \\
\mathcal{G}^4_{16} = 0 & \mathcal{G}^5_{16} = -\frac{1}{2} g_{66} & \mathcal{G}^6_{16} = \frac{1}{2} g_{56} \\
\mathcal{G}^1_{22} = g_{23} & \mathcal{G}^2_{22} = 0 & \mathcal{G}^3_{22} = -g_{12} \\
\mathcal{G}^4_{22} = g_{26} & \mathcal{G}^5_{22} = 0 & \mathcal{G}^6_{22} = -g_{24}
\end{array}$$

$$
\begin{array}{lll}
\mathcal{G}^1_{23} = \frac{1}{2}(g_{33} - g_{22}) & \mathcal{G}^2_{23} = \frac{1}{2}g_{12} & \mathcal{G}^3_{23} = -\frac{1}{2}g_{13} \\
\mathcal{G}^4_{23} = \frac{1}{2}(g_{36} - g_{25}) & \mathcal{G}^5_{23} = \frac{1}{2}g_{24} & \mathcal{G}^6_{23} = -\frac{1}{2}g_{34} \\
\mathcal{G}^1_{24} = \frac{1}{2}g_{34} & \mathcal{G}^2_{24} = -\frac{1}{2}g_{26} & \mathcal{G}^3_{24} = \frac{1}{2}(g_{25} - g_{14}) \\
\mathcal{G}^4_{24} = \frac{1}{2}g_{46} & \mathcal{G}^5_{24} = 0 & \mathcal{G}^6_{24} = -\frac{1}{2}g_{44} \\
\mathcal{G}^1_{25} = \frac{1}{2}(g_{35} + g_{26}) & \mathcal{G}^2_{25} = 0 & \mathcal{G}^3_{25} = \frac{1}{2}(\text{-}g_{15} - g_{24}) \\
\mathcal{G}^4_{25} = \frac{1}{2}g_{56} & \mathcal{G}^5_{25} = 0 & \mathcal{G}^6_{25} = -\frac{1}{2}g_{45} \\
\mathcal{G}^1_{26} = \frac{1}{2}(g_{36} - g_{25}) & \mathcal{G}^2_{26} = \frac{1}{2}g_{24} & \mathcal{G}^3_{26} = -\frac{1}{2}g_{16} \\
\mathcal{G}^4_{26} = \frac{1}{2}g_{66} & \mathcal{G}^5_{26} = 0 & \mathcal{G}^6_{26} = -\frac{1}{2}g_{46} \\
\mathcal{G}^1_{33} = -g_{23} & \mathcal{G}^2_{33} = g_{13} & \mathcal{G}^3_{33} = 0 \\
\mathcal{G}^4_{33} = -g_{35} & \mathcal{G}^5_{33} = g_{34} & \mathcal{G}^6_{33} = 0 \\
\mathcal{G}^1_{34} = -\frac{1}{2}g_{24} & \mathcal{G}^2_{34} = \frac{1}{2}(g_{14} - g_{36}) & \mathcal{G}^3_{34} = \frac{1}{2}g_{35} \\
\mathcal{G}^4_{34} = -\frac{1}{2}g_{45} & \mathcal{G}^5_{34} = \frac{1}{2}g_{44} & \mathcal{G}^6_{34} = 0 \\
\mathcal{G}^1_{35} = \frac{1}{2}(g_{36} - g_{25}) & \mathcal{G}^2_{35} = \frac{1}{2}g_{15} & \mathcal{G}^3_{35} = -\frac{1}{2}g_{34} \\
\mathcal{G}^4_{35} = -\frac{1}{2}g_{55} & \mathcal{G}^5_{35} = \frac{1}{2}g_{45} & \mathcal{G}^6_{35} = 0 \\
\mathcal{G}^1_{36} = \frac{1}{2}(\text{-}g_{26} - g_{35}) & \mathcal{G}^2_{36} = \frac{1}{2}(g_{16} + g_{34}) & \mathcal{G}^3_{36} = 0 \\
\mathcal{G}^4_{36} = -\frac{1}{2}g_{56} & \mathcal{G}^5_{36} = \frac{1}{2}g_{46} & \mathcal{G}^6_{36} = 0 \\
\mathcal{G}^1_{44} = 0 & \mathcal{G}^2_{44} = -g_{46} & \mathcal{G}^3_{44} = g_{45} \\
\mathcal{G}^4_{44} = 0 & \mathcal{G}^5_{44} = 0 & \mathcal{G}^6_{44} = 0 \\
\mathcal{G}^1_{45} = \frac{1}{2}g_{46} & \mathcal{G}^2_{45} = -\frac{1}{2}g_{56} & \mathcal{G}^3_{45} = \frac{1}{2}(g_{55} - g_{44}) \\
\mathcal{G}^4_{45} = 0 & \mathcal{G}^5_{45} = 0 & \mathcal{G}^6_{45} = 0 \\
\mathcal{G}^1_{46} = -\frac{1}{2}g_{45} & \mathcal{G}^2_{46} = \frac{1}{2}(g_{44} - g_{66}) & \mathcal{G}^3_{46} = \frac{1}{2}g_{56} \\
\mathcal{G}^4_{46} = 0 & \mathcal{G}^5_{46} = 0 & \mathcal{G}^6_{46} = 0 \\
\mathcal{G}^1_{55} = g_{56} & \mathcal{G}^2_{55} = 0 & \mathcal{G}^3_{55} = -g_{45} \\
\mathcal{G}^4_{55} = 0 & \mathcal{G}^5_{55} = 0 & \mathcal{G}^6_{55} = 0 \\
\mathcal{G}^1_{56} = \frac{1}{2}(g_{66} - g_{55}) & \mathcal{G}^2_{56} = \frac{1}{2}g_{45} & \mathcal{G}^3_{56} = -\frac{1}{2}g_{46} \\
\mathcal{G}^4_{56} = 0 & \mathcal{G}^5_{56} = 0 & \mathcal{G}^6_{56} = 0 \\
\mathcal{G}^1_{66} = -g_{56} & \mathcal{G}^2_{66} = g_{46} & \mathcal{G}^3_{66} = 0 \\
\mathcal{G}^4_{66} = 0 & \mathcal{G}^5_{66} = 0 & \mathcal{G}^6_{66} = 0
\end{array}
\tag{C.2}
$$

Appendix D

Metric compatible with the acceleration connection

In Section 7.3, we concluded that a metric compatible with the acceleration connection must satisfy Equation (7.65):

$$\hat{L}_k(g_{ij}) = \sum_l (\Gamma^l_{ik} g_{lj} + \Gamma^l_{jk} g_{li}). \tag{D.1}$$

The Christoffel symbols Γ^k_{ij} specify the acceleration connection and are listed in (7.63). For $k > 3$, the Christoffel symbols Γ^j_{ik} are 0, and therefore $\hat{L}_k(g_{ij}) = 0$. For this reason, we only list equations for $k \leq 3$. The equations were generated in Mathematica and are listed below. In the equations, $\mathcal{G}^k_{ij}$ stands for $\hat{L}_k(g_{ij})$.

$$\begin{array}{lll}
\mathcal{G}^1_{11} = 0 & \mathcal{G}^2_{11} = -g_{13} & \mathcal{G}^3_{11} = g_{12} \\
\mathcal{G}^1_{12} = \frac{1}{2} g_{13} & \mathcal{G}^2_{12} = -\frac{1}{2} g_{23} & \mathcal{G}^3_{12} = \frac{1}{2} g_{22} - \frac{1}{2} g_{11} \\
\mathcal{G}^1_{13} = -\frac{1}{2} g_{12} & \mathcal{G}^2_{13} = \frac{1}{2} g_{11} - \frac{1}{2} g_{33} & \mathcal{G}^3_{13} = \frac{1}{2} g_{23} \\
\mathcal{G}^1_{14} = 0 & \mathcal{G}^2_{14} = -\frac{1}{2} g_{34} - g_{16} & \mathcal{G}^3_{14} = \frac{1}{2} g_{24} + g_{15} \\
\mathcal{G}^1_{15} = g_{16} & \mathcal{G}^2_{15} = -\frac{1}{2} g_{35} & \mathcal{G}^3_{15} = \frac{1}{2} g_{25} - g_{14} \\
\mathcal{G}^1_{16} = -g_{15} & \mathcal{G}^2_{16} = g_{14} - \frac{1}{2} g_{36} & \mathcal{G}^3_{16} = \frac{1}{2} g_{26} \\
\mathcal{G}^1_{22} = g_{23} & \mathcal{G}^2_{22} = 0 & \mathcal{G}^3_{22} = -g_{12} \\
\mathcal{G}^1_{23} = \frac{1}{2} g_{33} - \frac{1}{2} g_{22} & \mathcal{G}^2_{23} = \frac{1}{2} g_{12} & \mathcal{G}^3_{23} = -\frac{1}{2} g_{13} \\
\mathcal{G}^1_{24} = \frac{1}{2} g_{34} & \mathcal{G}^2_{24} = -g_{26} & \mathcal{G}^3_{24} = g_{25} - \frac{1}{2} g_{14} \\
\mathcal{G}^1_{25} = \frac{1}{2} g_{35} + g_{26} & \mathcal{G}^2_{25} = 0 & \mathcal{G}^3_{25} = -\frac{1}{2} g_{15} - g_{24} \\
\mathcal{G}^1_{26} = \frac{1}{2} g_{36} - g_{25} & \mathcal{G}^2_{26} = g_{24} & \mathcal{G}^3_{26} = -\frac{1}{2} g_{16} \\
\mathcal{G}^1_{33} = -g_{23} & \mathcal{G}^2_{33} = g_{13} & \mathcal{G}^3_{33} = 0 \\
\mathcal{G}^1_{34} = -\frac{1}{2} g_{24} & \mathcal{G}^2_{34} = \frac{1}{2} g_{14} - g_{36} & \mathcal{G}^3_{34} = g_{35} \\
\mathcal{G}^1_{35} = g_{36} - \frac{1}{2} g_{25} & \mathcal{G}^2_{35} = \frac{1}{2} g_{15} & \mathcal{G}^3_{35} = -g_{34}
\end{array} \tag{D.2}$$

$$\mathcal{G}^1_{36} = -\tfrac{1}{2}g_{26} - g_{35} \qquad \mathcal{G}^2_{36} = \tfrac{1}{2}g_{16} + g_{34} \qquad \mathcal{G}^3_{36} = 0$$
$$\mathcal{G}^1_{44} = 0 \qquad \mathcal{G}^2_{44} = -2g_{46} \qquad \mathcal{G}^3_{44} = 2g_{45}$$
$$\mathcal{G}^1_{45} = g_{46} \qquad \mathcal{G}^2_{45} = -g_{56} \qquad \mathcal{G}^3_{45} = g_{55} - g_{44}$$
$$\mathcal{G}^1_{46} = -g_{45} \qquad \mathcal{G}^2_{46} = g_{44} - g_{66} \qquad \mathcal{G}^3_{46} = g_{56}$$
$$\mathcal{G}^1_{55} = 2g_{56} \qquad \mathcal{G}^2_{55} = 0 \qquad \mathcal{G}^3_{55} = -2g_{45}$$
$$\mathcal{G}^1_{56} = g_{66} - g_{55} \qquad \mathcal{G}^2_{56} = g_{45} \qquad \mathcal{G}^3_{56} = -g_{46}$$
$$\mathcal{G}^1_{66} = -2g_{56} \qquad \mathcal{G}^2_{66} = 2g_{46} \qquad \mathcal{G}^3_{66} = 0$$

Appendix E

Lie brackets for $se(3)$

In our derivations we need to evaluate Lie brackets of the basis vector fields $\hat{L}_i$. According to Equation (7.9), since the vector fields $\hat{L}_i$ are left invariant, it suffices to evaluate the brackets on $se(3)$. From Equation (7.5), we obtain:

$$
\begin{array}{lll}
[L_1, L_1] = 0 & [L_1, L_2] = L_3 & [L_1, L_3] = -L_2 \\
[L_1, L_4] = 0 & [L_1, L_5] = L_6 & [L_1, L_6] = -L_5 \\
[L_2, L_2] = 0 & [L_2, L_3] = L_1 & [L_2, L_4] = -L_6 \\
[L_2, L_5] = 0 & [L_2, L_6] = L_4 & [L_3, L_3] = 0 \\
[L_3, L_4] = L_5 & [L_3, L_5] = -L_4 & [L_3, L_6] = 0 \\
[L_4, L_4] = 0 & [L_4, L_5] = 0 & [L_4, L_6] = 0 \\
[L_5, L_5] = 0 & [L_5, L_6] = 0 & [L_6, L_6] = 0
\end{array}
\tag{E.1}
$$

Appendix F

The logarithm function on $SO(3)$ and $SE(3)$

The function $\log : SO(3) \to so(3)$ is defined by [87, 92]:

$$\log R = \theta \Xi, \qquad 2\cos\theta + 1 = \text{Trace}(R), \qquad \Xi = \frac{1}{2\sin\theta}(R - R^T) \tag{F.1}$$

where Ξ is a 3×3 skew-symmetric matrix such that the corresponding 3×1 vector, ξ, has unit Euclidean norm, and θ is a real number. Geometrically, ξ corresponds to the unit vector along the axis of rotation while θ is the angle of rotation. Note that the log function on $SO(3)$ is multi-valued.

The log function on $SE(3), \log : SE(3) \to se(3)$ is defined by [92]:

$$\log \begin{bmatrix} R & d \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} \Omega & v \\ 0 & 0 \end{bmatrix}, \quad \Omega = \log(R),$$

and

$$v = \left(I - \frac{1}{2}\Omega + \frac{2\sin|\omega| - |\omega|(1 + \cos|\omega|)}{2\omega \cdot \omega \sin|\omega|}\Omega\Omega \right)^{-1} d.$$

where in the last expression, ω is the vector corresponding to the skew-symmetric matrix Ω.

Bibliography

[1] A. Ailon and G. Langholz. On the existence of time-optimal control of mechanical manipulators. *Journal of Optimization Theory and Applications*, 46(1):1–21, May 1985.

[2] C. G. Atkeson and J. M. Hollerbach. Kinematic features of unrestrained vertical arm movements. *The Journal of Neuroscience*, 5(9):2318–2330, 1985.

[3] F. Avnaim, J. D. Boissonnat, and B. Faverjon. A practical exact motion planning algorithm for polygonal objects amidst polygonal obstacles. Technical Report 890, INRIA, Sophia-Antipolis,France, 1988.

[4] R. S. Ball. *The theory of screws*. Hodges & Foster, Dublin, 1876.

[5] R. Bellman. *Dynamic Programming*. Princeton University Press, Princeton, NJ, 1957.

[6] L. D. Berkovitz. Variational methods in problems of control and programming. *Journal of Mathematical Analysis and Applications*, 3:145–169, 1961.

[7] D. P. Bertsekas. *Constrained optimization and Lagrange multiplier methods*. Academic Press, New York, 1982.

[8] G. Bessonnet and J. P. Lallemand. Planning of optimal free paths of robotic manipulators with bounds on dynamic forces. In *Proceedings of 1993 International Conference on Robotics and Automation*, pages 270–275, Atlanta, GA, 1993.

[9] R. L. Bishop and S. I. Goldberg. *Tensor Analysis and Manifolds*. Dover Publications, New York, 1980.

[10] E. Bizzi, N. Accornero, W. Chapple, and N. Hogan. Posture control and trajectory formation during arm movement. *The Journal of Neuroscience*, 4(11):2738–2744, 1984.

[11] G. A. Bliss. *Lectures on the calculus of variations*. University of Chicago Press, Chicago, 1st phoenix edition, 1961.

[12] A. M. Bloch, P. S. Krishnaprasad, J. E. Marsden, and R. M. Murray. Nonholonomic mechanical systems and symmetry. Technical Report CDS 94-013, California Institute of Technology, Pasadena, CA, 1994.

[13] J.E. Bobrow, S. Dubowsky, and J.S. Gibson. Time-optimal control of robotic manipulators along specified paths. *Int. J. Robotic Research*, 4(3):3–17, 1985.

[14] J.-D. Boissonnat, O. Devillers, and S. Lazard. Motion planning of legged robots. In K. Goldberg, D. Halperin, J.-C. Latombe, and R. Willson, editors, *Algorithmic Foundations of Robotics*, pages 49–67. A K Peters, Ltd., Wellesley, MA, 1995.

[15] M. Brady, J. M. Hollerbach, T. L. Johnson, T. Lozano-Pérez, and M. T. Mason. *Robot motion : Planning and control*. MIT Press, Cambridge, MA, 1982.

[16] M. S. Branicky, V. S. Borkar, and S. K. Mitter. A unified framework for hybrid control. In *Proceedings of the 33rd IEEE Conference on Decision and Control*, pages 4228–4234, Lake Buena Vista, FL, 1994.

[17] M. S. Branicky and S. K. Mitter. Algorithms for optimal hybrid control. In *Proceedings of the 34th IEEE Conference on Decision and Control*, pages 2661–2666, New Orleans, LA, 1995.

[18] R. W. Brockett. Control theory and singular Riemannian geometry. In P. Hilton and G. Young, editors, *New directions in Applied Mathematics*, pages 11–27. Springer-Verlag, New York, 1981.

[19] R. W. Brockett. Robotic manipulators and the product of exponentials formula. In *Proc. Symp. Math. Theory Networks and Systems*, pages 120–129, Beer Sheba, Israel, 1983.

[20] R. W. Brockett. Hybrid models for motion control systems. In H. L. Trentelman and J. C. Willems, editors, *Essays in Control: Perspectives in the Theory and its Applications*, pages 29–53. Birkhäuser, Boston, 1993.

[21] R. A. Brooks. Solving the find-path problem by good representation of free space. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-13(3):190–197, 1983.

[22] A. E. Bryson and Y.-C. Ho. *Applied Optimal Control*. Hemisphere Publishing Co., New York, 1975.

[23] C. E. Buckley. *The application of continuum methods to path planning*. PhD thesis, Stanford University, Stanford, CA, 1985.

[24] D. Bullock and S. Grossberg. Neural dynamics of planned arm movements: Emergent invariants and speed-accuracy properties during trajectory formation. *Psychological Review*, 95(1):49–90, 1988.

[25] M. Camarinha, F. Silva Leite, and P. Crouch. Splines of class c^k on non-Euclidean spaces. *IMA J. Math. Control Inform.*, 12(4):399–410, 1995.

[26] J. F. Canny. *The complexity of robot motion planning*. MIT Press, Cambridge, MA, 1988.

[27] J. Cheeger and D. G. Ebin. *Comparison Theorems in Riemannian Geometry*. North-Holland Publishing Company, Amsterdam, 1975.

[28] C. H. Chen and V. Kumar. Motion planning of walking robots in environments with uncertainty. In *Proceedings of 1996 International Conference on Robotics and Automation*, Minneapolis, MN, 1996.

[29] I.-M. Chen and J. W. Burdick. A qualitative test for N-finger force-closure grasps on planar objects with applications to manipulation and finger gaits. In *Proceedings of 1993 International Conference on Robotics and Automation*, pages 814–820, Atlanta, GA, 1993.

[30] Y. Chen and A. A. Desrochers. A proof of the structure of the minimum-time control law for robotic manipulators using a hamiltonian formulation. *IEEE Transactions on Robotics and Automation*, 6(3):388–393, June 1990.

[31] P. Crouch and F. Silva Leite. The dynamic interpolation problem: on Riemannian manifolds, Lie groups, and symmetric spaces. *J. Dynam. Control Systems*, 1(2):177–202, 1995.

[32] O. Dahl. Path constrained motion optimization for rigid and flexible joint robots. In *Proceedings of 1993 International Conference on Robotics and Automation*, pages 223–229, Atlanta, GA, 1993.

[33] A. Divelbiss and J. T. Wen. Nonholonomic path planning with inequality constraints. In *Proceedings of the 32nd IEEE Conference on Decision and Control*, pages 2712–2717, San Antonio, TX, 1993.

[34] M. P. do Carmo. *Riemannian geometry*. Birkhauser, Boston, 1992.

[35] B. R. Donald, J. Jennings, and D. Rus. Information invariants for distributed manipulation. In K. Goldberg, D. Halperin, J.-C. Latombe, and R. Willson, editors, *Algorithmic Foundations of Robotics*, pages 431–457. A K Peters, Ltd., Wellesley, MA, 1995.

[36] M. Erdmann and M. T. Mason. An exploration of sensorless manipulation. In *Proceedings of 1986 International Conference on Robotics and Automation*, pages 1569–1574, San Francisco, CA, April 1986.

[37] R. Featherstone. *Robot Dynamics Algorithms*. Kluwer Academic Publishers, 1987.

[38] A. G. Feldman. Functional tuning of the nervous system with control of movement or maintenance of a steady posture ii. controllable parameters of the muscles. *Biofizika*, 11(3):498–508, 1966.

[39] A. G. Feldman. Change in the length of the muscle as a consequence of a shift in equilibrium in the muscle-load system. *Biofizika*, 19:534–538, 1974.

[40] C. Fernandes, L. Gurvits, and Z. X. Li. Optimal nonholonomic motion planning for a falling cat. In Z. Li and J. F. Canny, editors, *Nonholonomic motion planning*, pages 235–270. Kluwer Academic Publishers, Boston, 1993.

[41] P. M. Fitts. The information capacity of human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology*, 47:381–391, 1954.

[42] T. Flash and N. Hogan. The coordination of arm movements: An experimentally confirmed mathematical model. *The Journal of Neuroscience*, 5(7):1688–1703, 1985.

[43] W. H. Fleming. *Functions of several variables.* Springer-Verlag, New York, 2nd edition, 1977.

[44] G. Garvin. Trajectory formulation in human movement: An extension of existing models for single arm motions to coupled motions of two arms. Master's thesis, University of Pennsylvania, Philadelphia, PA, 1994.

[45] G. J. Garvin, M. Žefran, E. A. Henis, and V. Kumar. Two-arm trajectory planning in a manipulation task. Submitted to Biological Cybernetics, 1995.

[46] Q. J. Ge and B. Ravani. Computer aided geometric design of motion interpolants. *ASME Journal of Mechanical Design*, 116:756–762, 1994.

[47] K. Goldberg. Orienting polygonal parts without sensors. *Algorithmica*, 10(2):201–225, 1993. Special Issue on Computational Robotics.

[48] H. Goldstein. *Classical Mechanics.* Addison-Wesley, Cambridge, MA, 1950.

[49] J. Gregory and C. Lin. *Constrained optimization in the calculus of variations and optimal control theory.* Van Nostrand Reinhold, New York, 1992.

[50] R. L. Grossman, A. Nerode, A. P. Ravn, and H. Rischel (Eds.). *Hybrid systems*, volume 736 of *Lecture notes in computer science.* Springer-Verlag, New York, 1993.

[51] J. M. Herve. Intrinsic formulation of problems of geometry and kinematics of mechanisms. *Mechanism and Machine Theory*, 17(3):179–184, 1982.

[52] M. R. Hestenes. A general problem in the calculus of variations with applications to paths of least time. Research Memorandum RM-100, The RAND Corporation, 1949.

[53] M. R. Hestenes. *Calculus of variations and optimal control theory.* John Wiley & Sons, INc., New York, 1966.

[54] M. R. Hestenes. Multiplier and gradient methods. *Journal of optimization theory and applications*, 4:303–320, 1969.

[55] E. C. Hildreth and J. M. Hollerbach. The computational approach to vision and motor control. Technical report, MIT AI Laboratory, August 1985.

[56] M. W. Hirsch and S. Smale. *Differential equations, dynamical systems, and linear algebra.* Academic Press, New York, 1974.

[57] N. Hogan. An organizing principle for a class of voluntary movements. *Journal of Neuroscience*, 4(11):2745–2754, 1984.

[58] J. Hoschek and D. Lasser. *Fundamentals of Computer Aided Geometric Design.* AK Peters, 1993.

[59] H.-P. Huang and N. H. McClamroch. Time-optimal control for a robot contour following problem. *IEEE Transactions on Robotics and Automation*, 4(2):140–149, 1988.

[60] D. H. Jacobson. Differential dynamic programming methods for solving bang-bang control problems. *IEEE Transactions on Automatic Control*, AC-13(6):661–675, 1968.

[61] D. H. Jacobson and M. M. Lele. A transformation technique for optimal control problems with a state variable inequality constraint. *IEEE Transactions on Automatic Control*, AC-14(5):457–464, 1969.

[62] D. H. Jacobson, M. M. Lele, and J. L. Speyer. New necessary conditions of optimality for control problems with state-variable inequality constraints. *Journal of mathematical analysis and applications*, 35:255–284, 1971.

[63] M. E. Kahn and B. Roth. The near-minimum-time control of open-loop articulated kinematic chains. *ASME Journal of Dynamic Systems, Measurement, and Control*, 93:164–172, September 1971.

[64] A. Karger and J. Novak. *Space Kinematics and Lie Groups*. Gordon and Breach Science Publishers, 1985.

[65] M. Kawato. Optimization and learning in neural networks for formation and control of coordinated movement. Technical Report TR-A-0086, ATR Auditory and Visual Perception Research Laboratories, Kyoto, Japan, 1990.

[66] S. W. Keele. Behavioral analysis of movement. In V. B. Brooks, editor, *Handbook of Physiology*, pages 1391–1414. American Physiological Society, Bethesda, MD, 1981.

[67] O. Khatib. Real-time obstacle avoidance for manipulators and mobile robots. *International Journal of Robotics Research*, 5(1):90–98, 1986.

[68] K.H.Hunt. *Kinematic Geometry of Mechanisms*. Clarendon Press, Oxford, 1978.

[69] C. A. Klein and C.-H. Huang. Review of pseudoinverse control for use with kinematically redundant manipulators. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-13:245–250, 1983.

[70] D. E. Koditschek. Exact robot navigation by means of potential functions: Some topological considerations. In *Proceedings of 1987 International Conference on Robotics and Automation*, pages 1–6, Raleigh, NC, 1987.

[71] W. Kohn, A. Nerode, J. B. Remmel, and Xiolin Ge. Multiple agent hybrid control: carrier manifolds and chattering approximations to optimal control. In *Proceedings of the 33rd IEEE Conference on Decision and Control*, pages 4221–4227, Lake Buena Vista, FL, 1994.

[72] G. Lafferriere and H. J. Sussmann. A differential geometric approach to motion planning. In Z. Li and J. F. Canny, editors, *Nonholonomic motion planning*, pages 235–270. Kluwer Academic Publishers, Boston, 1993.

[73] L. S. Lasdon, S. K. Mitter, and A. D. Warren. The conjugate gradient method for optimal control problems. *IEEE Transactions on Automatic Control*, AC-12(2):132–138, 1967.

[74] J.-C. Latombe. *Robot motion planning*. Kluwer Academic Publishers, Boston, 1991.

[75] J.-P. Laumond, P. E. Jacobs, M. Taix, and R. M. Murray. A motion planner for nonholonomic mobile robots. *IEEE Transactions on Robotics and Automation*, 10(5):577–593, 1994.

[76] S. Leveroni and K. Salisbury. Reorienting objects with a robot hand using grasp gaits. In *International Symposium on Robotics Research*, pages 49–63, Munich, Germany, 1995.

[77] Z. Li and J. F. Canny. *Nonholonomic motion planning*. Kluwer Academic Publishers, Boston, 1993.

[78] Z. Li, J. F. Canny, and S. S. Sastry. On motion planning for dextrous manipulation, Part 1: The problem formulation. In *Proceedings of 1989 International Conference on Robotics and Automation*, pages 775–780, 1989.

[79] H. Lipkin and J. Duffy. Hybrid twist and wrench control. *ASME J. of Mechanisms, Transmissions and Automation in Design*, September 1987.

[80] J. Loncaric. Normal forms of stiffness and compliance matrices. *IEEE Journal of Robotics and Automation*, RA-3(6):567–572, 1987.

[81] T. Lozano-Pérez and M. A. Wesley. An algorithm for planning collision-free paths among polyhedral obstacles. *Communications of the ACM*, 22(10):560–570, 1979.

[82] K. M. Lynch and M. T. Mason. Stable pushing: Mechanics, controllability and planning. In K. Goldberg, D. Halperin, J.-C. Latombe, and R. Willson, editors, *Algorithmic Foundations of Robotics*, pages 239–262. A K Peters, Ltd., Wellesley, MA, 1995.

[83] B. Ma and W. S. Levine. An algorithm for solving control constrained optimal control problems. In *Proceedings of the 32nd IEEE Conference on Decision and Control*, pages 3784–3790, San Antonio, TX, 1993.

[84] D. P. Martin, J. Baillieul, and J. M. Hollerbach. Resolution of kinematic redundancy using optimization techniques. *IEEE Transactions on Robotics and Automation*, 5, 1989.

[85] M. T. Mason. Mechanics and planning of manipulator pushing operations. *International Journal of Robotics Research*, 5(3), 1986.

[86] D. Q. Mayne and E. Polak. An exact penalty function algorithm for control problems with state and control constraints. *IEEE Transactions on Automatic Control*, AC-32(5):380–387, 1987.

[87] J. M. McCarthy. *An Introduction to Theoretical Kinematics*. MIT Press, 1990.

[88] R. B. McGhee and D. E. Orin. A mathematical programming approach to control of joint positions and torques in legged locomotion systems. In *Proceedings of ROMANSY-76 Symposium*, 1976.

[89] A. Miele and T. Wang. Primal-Dual properties of sequential gradient-restoration algorithms for optimal control problems, Part 2, General problem. *Journal of mathematical analysis and applications*, 119:21–54, 1986.

[90] P. Morasso. Spatial control of arm movements. *Experimental Brain Research*, 42:223–227, 1981.

[91] R. M. Murray. Nonlinear control of mechanical systems: A lagrangian perspective. In *IFAC Symposium on Nonlinear Control Systems Design*, 1995.

[92] R. M. Murray, Z. Li, and S. S. Sastry. *A Mathematical Introduction to Robotic Manipulation*. CRC Press, 1994.

[93] R. M. Murray and S. S. Sastry. Nonholonomic motion planning: Steering using sinusoids. *IEEE Transactions on Automatic Control*, 38(5):700–716, 1993.

[94] E. Muybridge. *Animals in motion*. Dover Publications, New York, 1957. First published 1887.

[95] E. Muybridge. *The human figure in motion*. Dover Publications, New York, 1957. First published 1887.

[96] Y. Nakamura and H. Hanafusa. Optimal redundancy control of robot manipulators. *International Journal of Robotics Research*, 6, 1987.

[97] W. L. Nelson. Physical principles for economies of skilled movements. *Biological Cybernetics*, 46:135–147, 1983.

[98] A. Nerode and W. Kohn. Models for hybrid systems: Automata, topologies, stability. In R. L. Grossman, A. Nerode, A. P. Ravn, and H. Rischel, editors, *Hybrid systems*, pages 317–356. Springer-Verlag, New York, 1993.

[99] C. P. Neuman and A. Sen. A suboptimal control algorithm for constrained problems using cubic splines. *Automatica*, 9:601–613, 1973.

[100] N. J. Nilsson. A mobile automaton: An application of artificial intelligence techniques. In *Proceedings of the 1st IJCAI*, pages 509–520, Washington, D.C., 1969.

[101] L. Noakes, G. Heinzinger, and B. Paden. Cubic splines on curved spaces. *IMA J. of Math. Control & Information*, 6:465–473, 1989.

[102] C. Ó'Dúnlaing, M. Sharir, and C. K. Yap. Retraction: A new approach to motion planning. In *Proceedings of the 15th ACM Symposium on the Theory of Computing*, pages 207–220, Boston, 1983.

[103] J. P. Ostrowski. *The Mechanics and Control of Undulatory Locomotion*. PhD thesis, California Institute of Technology, Pasadena, CA, 1995.

[104] P. R. Pagilla and M. Tomizuka. Control of mechanical systems subject to unilateral constraints. In *Proceedings of the 34th IEEE Conference on Decision and Control*, New Orleans, LA, 1995.

[105] F. C. Park. Distance metrics on the rigid-body motions with applications to mechanism design. *ASME Journal of Mechanical Design*, 117(1):48–54, 1995.

[106] F. C. Park and R. W. Brockett. Kinematic dexterity of robotic mechanisms. *International Journal of Robotics Research*, 13(1):1–15, 1994.

[107] F. C. Park and B. Ravani. Bezier curves on Riemannian manifolds and Lie groups with kinematics applications. *ASME Journal of Mechanical Design*, 117(1):36–40, 1995.

[108] R. P. Paul. *Robot Manipulators, Mathematics, Programming and Control*. The MIT Press, Cambridge, 1981.

[109] F. Pfeiffer and R. Johanni. A concept for manipulator trajectory planning. *IEEE Journal of Robotics and Automation*, RA-3(2):115–123, April 1987.

[110] D. L. Pieper. *The kinematics of manipulators under computer control*. PhD thesis, Stanford University, 1968.

[111] E. Polak. *Computational methods in optimization*. Academic Press, New York, 1971.

[112] E. Polak. On the use of consistent approximations in the solution of semi-infinite optimization and optimal control problems. *Mathematical programming*, 62:385–414, 1993.

[113] L. S. Pontryagin, V. G. Boltyanskii, R. V. Gamkrelidze, and E. F. Mishchenko. *The mathematical theory of optimal processes*. Interscience Publishers, New York, 1962.

[114] M. J. D. Powell. A method for nonlinear constraints in minimization problems. In R. Fletcher, editor, *Optimization*, pages 283–298. Academic Press, New York, 1969.

[115] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in C*. Cambridge University Press, Cambridge, 1988.

[116] R. Pytlak and R. B. Vinter. Second-order method for optimal control problems with state constraints and piecewise-constant controls. In *Proceedings of the 34th IEEE Conference on Decision and Control*, New Orleans, LA, 1995.

[117] E. Rimon and D. E. Koditschek. Exact robot navigation using artificial potential functions. *IEEE Transactions on Robotics and Automation*, 8(5):501–518, 1992.

[118] R. M. Rosenberg. *Analytical Dynamics of Discrete Systems*. Plenum Press, New York, 1977.

[119] H. Sagan. *Introduction to the calculus of variations*. McGraw-Hill, New York, 1969.

[120] C. Samson and K. Ait-Abderrahim. Feedback control of a nonholonomic wheeled cart in cartesian space. In *Proceedings of 1991 International Conference on Robotics and Automation*, pages 1136–1141, Sacramento, CA, April 1991.

[121] D.H. Sattinger and O.L. Weaver. *Lie groups and algebras with applications to physics.* Springer-Verlag, New York, 1986.

[122] B. F. Schutz. *Geometrical Methods of Mathematical Physics.* Cambridge University Press, Cambridge, 1980.

[123] A. L. Schwartz. *Theory and implementation of numerical methods based on Runge-Kutta integration for solving optimal control problems.* PhD thesis, University of California at Berkeley, Berkeley, CA, 1996.

[124] J. T. Schwartz and M. Sharir. On the 'piano movers' problem: 1. The case of two-dimensional rigid polygonal body moving amidst polygonal barriers. *Communications on pure and applied mathematics*, 36:345–398, 1983.

[125] J. T. Schwartz and M. Sharir. On the 'piano movers' problem: 2. General techniques for computing topological properties of real algebraic manifolds. *Advances in applied mathematics*, 4:298–351, 1983.

[126] Z. Shiller. On singular points and arcs in path constrained time optimal motions. *ASME, Dyn. Syst. Contr. Division DSC-42*, 42:141–147, 1992.

[127] Z. Shiller. Time-energy optimal control of articulated systems with geometric path constraints. In *Proceedings of 1994 International Conference on Robotics and Automation*, San Diego, CA, 1994.

[128] Z. Shiller and S. Dubowsky. On computing the global time-optimal motions of robotic manipulators in the presence of obstacles. *IEEE Transactions on Robotics and Automation*, 7(6):785–797, December 1991.

[129] K. G. Shin and N. D. McKay. Minimum-time control of robotic manipulators with geometric constraints. *IEEE Transactions on Automatic Control*, AC-30(6):531–541, 1985.

[130] K. Shoemake. Animating rotation with quaternion curves. *ACM Siggraph*, 19(3):245–254, 1985.

[131] S. Singh and M. C. Leu. Optimal trajectory generation for robotic manipulators using dynamic programming. *ASME Journal of Dynamic Systems, Measurement, and Control*, 109, 1989.

[132] J.-J. E. Slotine and H. S. Yang. Improving the efficiency of time-optimal path-following algorithms. *IEEE Transactions on Robotics and Automation*, 5(1):118–124, February 1989.

[133] J. F. Soechting and F. Lacquaniti. Invariant characteristics of a pointing movement in man. *The Journal of Neuroscience*, 1(7):710–720, 1981.

[134] E. D. Sontag and Y. Lin. Gradient techniques for systems with no drift. In *Proceedings of Conference on Signals and Systems*, Princeton, NJ, 1992.

[135] E. D. Sontag and H. J. Sussmann. Time-optimal control of manipulators. In *Proceedings of 1986 International Conference on Robotics and Automation*, pages 1692–1697, San Francisco, CA, April 1986.

[136] R. B. Stein. What muscle variable(s) does the nervous system control in limb movements? *The Behavioral and Brain Sciences*, 5:535–577, 1982.

[137] K. Sugimoto and J. Duffy. Application of linear algebra to screw systems. *Mechanism and Machine Theory*, 17(1):73–83, 1982.

[138] K. C. Suh and J. M. Hollerbach. Local versus global torque optimization of redundant manipulators. In *Proceedings of 1987 International Conference on Robotics and Automation*, Raleigh, NC, 1987.

[139] H. J. Sussmann. Lie brackets, real analyticity and geometric control. In R. Millman, R. W. Brockett, and H. J. Sussmann, editors, *Differential Geometric Control Theory*, pages 1–116. Birkhauser, Boston, 1983.

[140] H. J. Sussmann and W. Liu. Lie bracket extensions and averaging: The single-bracket case. In Z. Li and J. F. Canny, editors, *Nonholonomic motion planning*, pages 109–148. Kluwer Academic Publishers, Boston, 1993.

[141] K. L. Teo, C. J. Goh, and K. H. Wong. *A unified computational approach to optimal control problems*. Longman Scientific & Technical, Burnt Mill, Harlow, UK, 1991.

[142] D. M. Tilbury. *Exterior differential systems and nonholonomic motion planning*. PhD thesis, University of California at Berkeley, Berkeley, CA, 1994.

[143] Y. Uno, M. Kawato, and R. Suzuki. Formation and control of optimal trajectory in human multijoint arm movement. *Biological Cybernetics*, 61:89–101, 1989.

[144] F. A. Valentine. The problem of lagrange with differential inequalities as added side conditions. In *Contributions to the Calculus of Variations*, pages 407–448. Chicago University Press, Chicago, IL, 1937.

[145] P. Viviani and C. Terzuolo. Trajectory determines movement dynamics. *Neuroscience*, 7:431–437, 1982.

[146] M. Vukobratović and M. Kirćanski. A method for optimal synthesis of manipulation robot trajectories. *ASME Journal of Dynamic Systems, Measurement, and Control*, 104, 1982.

[147] K. J. Waldron. Geometrically based manipulator rate control algorithms. *Mechanism and Machine Theory*, 17(6):379–385, 1982.

[148] G. Walsh, D.Tilbury, R. Murray, and J. P. Laumond. Stabilization of trajectories for systems with nonholonomic constraints. *IEEE Transactions on Automatic Control*, 39(1):216–222, 1994.

[149] C.-C. Wang. *Local and Repeatable Coordination Schemes for Redundant Manipulators*. PhD thesis, University of Pennsylvania, Philadelphia, PA, 1995.

[150] J. Wei and E. Norman. On global representations of the solutions of linear differential equations as a product of exponentials. *Proc. American Mathematical Society*, pages 327–334, April 1964.

[151] D. E. Whitney. Resolved motion rate control of manipulators and human prostheses. *IEEE Transactions on Man-Machine Systems*, MMS-10:47–53, 1969.

[152] D. E. Whitney. The mathematics of coordinated control of prosthetic arms and manipulators. *ASME Journal of Dynamic Systems, Measurement, and Control*, 94:303–309, 1972.

[153] David A. Winter. *Biomechanics and motor control of human movement*. John Wiley & Sons, New York, second edition, 1990.

[154] M. Žefran. Numerical methods for computing optimal trajectories of robots. Master's thesis, University of Pennsylvania, Philadelphia, PA, 1995.

[155] M. Žefran, J. Desai, and V. Kumar. Continuous motion plans for robotic systems with changing dynamic behavior. Proceedings of 2nd Int. Workshop on Algorithmic Foundations of Robotics, 1996.

[156] M. Žefran and V. Kumar. Optimal control of systems with unilateral constraints. In *Proceedings of 1995 International Conference on Robotics and Automation*, pages 2695–2700, Nagoya, Japan, 1995.

[157] M. Žefran and V. Kumar. Coordinate-free formulation of the cartesian stiffness matrix. In *Proceedings of the 5th Int. Symposium on Advances in Robot Kinematics*, Portorož, Slovenia, 1996.

[158] M. Žefran and V. Kumar. Planning of smooth motions on se(3). In *Proceedings of 1996 International Conference on Robotics and Automation*, pages 121–126, Minneapolis, MN, April 1996.

[159] M. Žefran, V. Kumar, and C. Croke. On the generation of smooth three-dimensional rigid body motions. Submitted to IEEE Transactions on Robotics and Automation, 1995.

[160] M. Žefran, V. Kumar, and C. Croke. Choice of Riemannian metrics for rigid body kinematics. In *Proceedings of the ASME 24th Biennial Mechanisms Conference*, Irvine, CA, 1996.

[161] M. Žefran, V. Kumar, J. Desai, and E. Henis. Two-arm manipulation: What can we learn by studying humans? In *Proceedings IROS'95*, Pittsburgh, PA, August 1995.

[162] M. Žefran, V. Kumar, and X. Yun. Optimal trajectories and force distribution for cooperating arms. In *Proceedings of 1994 International Conference on Robotics and Automation*, pages 874–879, San Diego, 1994.