
**NONHOLONOMIC
MOTION PLANNING**

**THE KLUWER INTERNATIONAL SERIES
IN ENGINEERING AND COMPUTER SCIENCE**

ROBOTICS: VISION, MANIPULATION AND SENSORS

Consulting Editor: Takeo Kanade

**KINEMATIC MODELING, IDENTIFICATION AND CONTROL OF ROBOT
MANIPULATORS**, H.W. Stone
ISBN: 0-89838-237-8

OBJECT RECOGNITION USING VISION AND TOUCH, P. Allen
ISBN: 0-89838-245-9

**INTEGRATION, COORDINATION AND CONTROL OF MULTI-SENSOR ROBOT
SYSTEMS**, H.F. Durrant-Whyte
ISBN: 0-89838-247-5

BAYESIAN MODELING OF UNCERTAINTY IN LOW-LEVEL VISION, R. Szeliski
ISBN: 0-7923-9039-3

VISION AND NAVIGATION: THE CMU NAVLAB, C. Thorpe (editor)
ISBN: 0-7923-9068-7

TASK-DIRECTED SENSOR FUSION AND PLANNING: A Computational Approach,
G. D. Hager
ISBN: 0-7923-9108-X

COMPUTER ANALYSIS OF VISUAL TEXTURES, F. Tomita and S. Tsuji
ISBN: 0-7923-9114-4

DATA FUSION FOR SENSORY INFORMATION PROCESSING SYSTEMS, J. Clark
and A. Yuille
ISBN: 0-7923-9120-9

**PARALLEL ARCHITECTURES AND PARALLEL ALGORITHMS FOR INTEGRATED
VISION SYSTEMS**, A.N. Choudhary, J. H. Patel
ISBN: 0-7923-9078-4

ROBOT MOTION PLANNING, J. Latombe
ISBN: 0-7923-9129-2

DYNAMIC ANALYSIS OF ROBOT MANIPULATORS: A Cartesian Tensor Approach,
C.A Balafoutis, R.V. Patel
ISBN: 07923-9145-4

PERTURBATION TECHNIQUES FOR FLEXIBLE MANIPULATORS: A. Fraser and
R. W. Daniel
ISBN: 0-7923-9162-4

COMPUTER AIDED MECHANICAL ASSEMBLY PLANNING: L. Homen de Mello and
S. Lee
ISBN: 0-7923-9205-1

INTELLIGENT ROBOTIC SYSTEMS FOR SPACE EXPLORATION: Alan A. Desrochers
ISBN: 0-7923-9197-7

MEASUREMENT OF IMAGE VELOCITY: David J. Fleet
ISBN: 0-7923-9198-5

DIRECTED SONAR SENSING FOR MOBILE ROBOT NAVIGATION: John J. Leonard,
Hugh F. Durrant-Whyte
ISBN: 0-7923-9242-6

A GENERAL MODEL OF LEGGED LOCOMOTION ON NATURAL TERRAIN: David
J. Mankos
ISBN: 0-7923-9247-7

INTELLIGENT ROBOTIC SYSTEMS: THEORY, DESIGN AND APPLICATIONS K.
Valavanis, G. Saridis
ISBN : 0-7923-9250-7

QUALITATIVE MOTION UNDERSTANDING: W. Burger, B. Bhanu
ISBN: 0-7923-9251-5

NONHOLONOMIC MOTION PLANNING

Edited by:

Zexiang Li

*Dept. of Electrical & Electronic Engineering
Hong Kong University of Science & Technology
Clear Water Bay
Kowloon, Hong Kong*

J.F. Canny

*Dept. of Electrical Engineering & Computer Sciences
University of California at Berkeley
Berkeley, CA 94720*



SPRINGER SCIENCE+BUSINESS MEDIA, LLC

Library of Congress Cataloging-in-Publication Data

Nonholonomic motion planning / edited by Zexiang Li, J.F. Canny.

p. cm. -- (The Kluwer international series in engineering and computer science. Robotics)

Includes bibliographical references and index.

ISBN 978-1-4613-6392-7 ISBN 978-1-4615-3176-0 (eBook)

DOI 10.1007/978-1-4615-3176-0

1. Robots--Motion. I. Li, Zexiang, 1961-. II. Canny, John.

III. Series.

TJ211.4.N66 1992

629.8'92--dc20

92-27560

CIP

Copyright © 1993 by Springer Science+Business Media New York
Originally published by Kluwer Academic Publishers in 1993
Softcover reprint of the hardcover 1st edition 1993

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form or by any means, mechanical, photo-copying, recording, or otherwise, without the prior written permission of the publisher, Springer Science+Business Media, LLC.

Printed on acid-free paper.

TABLE OF CONTENTS

Introduction	vii
1. Non-holonomic Kinematics and the Role of Elliptic Functions in Constructive Controllability - R.W. Brockett, Liyi Dai	1
2. Steering Nonholonomic Control Systems Using Sinusoids - Richard M. Murray, S. Shankar Sastry	23
3. Smooth Time-Periodic Feedback Solutions for Nonholonomic Motion Planning - L. Gurvits, Z.X. Li	53
4. Lie Bracket Extensions and Averaging: The Single-Bracket Case - Héctor J. Sussmann, Wensheng Liu	109
5. Singularities and Topological Aspects in Nonholonomic Motion Planning - Jean-Paul Laumond	149
6. Motion Planning for Nonholonomic Dynamic Systems- Mahmut Reyhanoglu, N. Harris McClamroch, Anthony M. Bloch . .	201
7. A Differential Geometric Approach to Motion Planning- Gerardo Lafferriere, Héctor J. Sussmann	235
8. Planning Smooth Paths for Mobile Robots- Paul Jacobs, John Canny	271
9. Nonholonomic Control and Gauge Theory- Richard Montgomery	343
10. Optimal Nonholonomic Motion Planning for a Falling Cat- C. Fernandes, L. Gurvits, Z.X. Li	379
11. Nonholonomic Behavior in Free-floating Space Manipulators and its Utilization - Evangelos G. Papadopoulos	423
Index	447

constraint limits only the freedom of motion. The case of parallel parking subject to rolling contact constraints serves as a perfect illustrative example. First, if cars were able to move sideways, then a careless mistake can cause a severe car accident. Second, given adequate knowledge about the constraint, one can actively explore or maneuver the constraint to move from one configuration to another. The motions of a falling cat serve as another illustrative example.

Research in traditional motion planning problem or the so-called Piano Movers' Problem has a long history, but research in Nonholonomic Motion Planning is very recent, and has been driven by studies of the following problems.

1. Mobile robots navigating in a cluttered environment. The kinematics of the driving mechanisms of robot carts results in constraints on the instantaneous velocities that they can achieve. For instance, a cart with two forward drive wheels and two back wheels cannot move sideways. This was first pointed out by Laumond in the context of motion planning for a mobile robot at Toulouse, France ([LS89]).
2. Multifingered hands rolling on the surface of a grasped object. If the object is twirled through a cyclic motion which returns it to its initial position and orientation, and the fingers roll without slipping on the surface of the object, then they would not return to their initial positions. This feature can be used to plan the regrasp of a poorly grasped object or to choose a good initial grasp. This application of nonholonomic motion planning was first pointed out by Li ([LC90] and [LCS89]). A formulation of nonholonomic motion planning in the framework of a nonlinear control system was given in ([LC90], [LCS89] and [MS90]).
3. Space Robots. Unanchored robots in space are difficult to control with either thrusters or internal motors since they conserve total angular momentum. In fact the motion of astronauts on space walks is of this ilk, so that the planning of the strategy required to reorient an astronaut is a problem

of this nature (also gymnasts and divers are good examples of this kind of motion planning). Application of nonholonomic motion planning in studying these problems can be found in ([VD87], [NM90], [LM90], [Kri90], [BMR90] and [WS91]).

4. Ultrasonic Motors. Based on the traveling wave method and nonholonomic constraints, Panasonic Inc. developed a class of ultrasonic motors whose torque-to-mass ratio could be orders of magnitude higher than that of traditional motors. Operating principles and controls of the ultrasonic motors were studied in [Bro88].

The link made between nonholonomic motion planning and constructive nonlinear control has sparked a series of research activities in this problem: (1) more applications of nonholonomic motion planning are being found in planning/control of systems such as redundant robots, space based robots and even a one-legged hopper; (2) classical and modern tools in nonlinear and optimal control, starting with the works of Hermann in 1963 and continuing with the works of Brockett, Krener, Sussmann and others (e.g., see [Bro81], [Sus83], [HK77] and [HH70]) are being applied to this problem; (3) differential geometric methods and classical mechanics are being utilized to characterize solutions of nonholonomic motion planning ([Mon90]); (4) new algorithms and approaches are being developed for computing collision-free paths of nonholonomic motion planning.

This book grew out of the Workshop on Nonholonomic Motion Planning that took place at the 1991 IEEE International Conference on Robotics and Automation. It consists of a collection of papers representing new development in this area. Contributors of the book include robotics engineers, nonlinear control experts, differential geometers and applied mathematicians.

The chapters of this book may be arranged into three groups:

1 Controllability

One of the key mathematical tools needed to study nonholonomic motion planning is nonlinear controllability. While nonholonomic constraints usually appear as differential form constraints, they can be easily dualized to give steering problems in the form of control systems. There is a long history of activity in control theory and optimal control for steering problems on \mathbb{R}^n as well as Lie Groups and other manifolds beginning with the work of Hermann in 1963 and continuing with the work of several researchers including Brockett, Krener, Sussmann and others([Bro81], [Sus83], [HK77], [HH70] for example). In the present collection we have four papers in the area of controllability:

1. “Nonholonomic Kinematics and the Role of Elliptic Functions in Constructive Controllability”, by Brockett and Dai. In this paper the authors continue with and extend the earlier work of Brockett which showed that the optimal control laws needed for steering drift free control systems with control Lie algebra being the Heisenberg algebra were sinusoids at integrally related frequencies. In the current paper, the authors show that for certain other control Lie algebras, which are nilpotent but involve two layers of brackets, the optimal steering inputs are elliptic functions. They apply the theory to the regrasping of a ball grasped between two parallel plates.
2. “Steering Nonholonomic Control Systems Using Sinusoids”, by Murray and Sastry. In this paper the authors extend the results of Brockett alluded to above to constructively steer several classes of nonholonomic systems using sinusoids and some Fourier analysis even when their control Lie algebras are not Heisenberg algebras. The classes of systems steerable are referred to as “chained form systems”.
3. “Smooth Time Periodic Feedback Solutions for Nonholonomic Motion Planning”, by Gurvits and Li. In this paper, the authors use averaging techniques to compute asymptotic

behaviors of nonholonomic systems under application of a class of highly oscillatory inputs and present an algorithm for computing time-periodic feedback solutions for nonholonomic motion planning. Analytic solutions of a class of nonholonomic systems using Fourier analysis are also discussed.

4. “Lie Bracket Extensions and Averaging: the Single Bracket Case”, by Sussmann and Liu. Here Sussmann and Liu show how highly oscillatory high amplitude sinusoids may be used to generate Lie bracket motion in a drift free control system. Their method is general (though only the “Single-Bracket” case is discussed here) and enables the generation of inputs required to steer the system along the basis directions of a Phillip Hall basis for the control Lie algebra. Consequently, algorithms for steering the control system along all possible directions are given.

2 Motion Planning for Mobile Robots

In recent years there has been a great deal of activity in the generation of efficient motion planning algorithms for robots. A tremendous variety of algorithms for particular motion planning problems have appeared, both heuristic and guaranteed. Most of this work has focussed on the “Piano Movers’ Problem” and generalizations, where the obstacle positions are known, and dynamical constraints are not considered. In this section on motion planning the papers are focussed on problems with non-holonomic velocity constraints as well as constraints on the generalized coordinates. These lead to the most difficult planning problems, since it is non-trivial to find a path even in the absence of obstacles. There are three papers in this area:

1. “Singularity and Topological Aspects in Nonholonomic Motion Planning”, by Laumond. In this survey paper, Laumond gives a comprehensive discussion of the aggregation

of tools from controllability and the sub-Riemannian geometry to solve the problem of steering a mobile robot with n trailers in a field of obstacles.

2. “Motion Planning for Nonholonomic Dynamic Systems”, by Reyhanoglu, McClamroch and Bloch. The authors develop a class of nonholonomic systems called Caplygin systems, whose constraints are cyclic in the variables. Methods for steering such systems are discussed and applied to a knife edge moving on a planar surface or a planar multi-body system.
3. “A Differential Geometric Approach to Motion Planning”, by Lafferriere and Sussmann. The authors develop methods for steering drift free systems using the basic definition of the Lie Bracket and feedback nilpotentization of the associated control Lie algebra.
4. “Planning Smooth Paths for Mobile Robots”, by Jacobs and Canny. The authors define a set of canonical trajectories that satisfy non-holonomic constraints and present a graph search algorithm to compute approximate paths for mobile robots.

3 Falling Cats, Space Robots and Gauge Theory

There are numerous connections to be made between symplectic geometry techniques for the study of holonomies in mechanics, gauge theory and control. In this section, using the back drop of examples drawn from space robots and falling cats reorienting themselves, we have three papers which aim to make these connections:

1. “Nonholonomic Control and Gauge Theory” by Montgomery. The author provides a dictionary between gauge theory and

control and shows how it is useful to study the control of deformable bodies (cats, gymnasts, astronauts). It also contains some useful results on stabilization.

2. “Optimal Nonholonomic Motion Planning for a Falling Cat”, by Fernandes, Gurvits and Li. Here the authors work out the details of how a falling cat might change her orientation in mid-air to land on her feet. First, they modeled a falling-cat as two rigid bodies coupled by a universal or a ball-in-socket joints. Then, they applied Ritz approximation theory in functional analysis to reduce the motion planning problem into a finite dimensional optimization problem. Finally, they developed a Basis Algorithm to solve the optimization problem.
3. “Nonholonomic Behavior in Free-floating Space Manipulators and its Utilization”, by Papadopoulos. The author discusses the reorientation of a spacecraft using internal motions of an on-board manipulator.

We invite the reader to plunge into this exciting fusion which is a representative but not complete sample of the fantastic amount of new literature in this field. Nonholonomies are around us everywhere: for example, walking and swimming are cyclic motions which produce a forward drift!

Finally, we would like to thank all contributors whose efforts have made this workshop and book possible. We would also like to thank J. Burdick of California Institute of Technology and J. Wen of Rensselaer Polytechnic Institute for co-organizing the workshop.

S.S. Sastry, Berkeley, California
Z.X. Li, New York, New York

References

- [BMR90] A. Bloch, N.H. McClamroch, and M. Reyhanoglu. Controllability and stabilizability properties of a nonholonomic control system. In *Conf. on Decision and Control*, pages 1312–1314, 1990.
- [Bro81] R. Brockett. Control theory and singular riemannian geometry. In P. Hilton and G. Young, editors, *New Directions in Applied Mathematics*, pages 11–27. Springer-Verlag, 1981.
- [Bro88] R. W. Brockett. On the rectification of vibratory motion. In *Proceedings of Microactuators and Micromechanism*, Salt Lake City, Utah, 1988.
- [HH70] G.W. Haynes and H. Hermes. Nonlinear controllability via lie theory. *SIAM J. Control*, 8(4):450–460, 1970.
- [HK77] R. Hermann and A.J. Krener. Nonlinear controllability and observability. *IEEE Trans. on Automatic Control*, 22:728–740, 1977.
- [Kri90] P.S. Krishnaprasad. Geometric phases, and optimal reconfiguration for multibody systems. Technical report, System research center, University of Maryland at College Park, 1990.
- [LC90] Z. Li and J. Canny. Motion of two rigid bodies with rolling constraint. *IEEE Trans. on Robotics and Automation*, RA2-06:62–72, 1990.
- [LCS89] Z.X. Li, J. F. Canny, and S.S. Sastry. On motion planning for dexterous manipulation. In *IEEE International conference on robotics and automation*, 1989.
- [LM90] Z. Li and R. Montgomery. Dynamics and optimal control of a legged robot in flight phase. In *Proceedings of IEEE Int. Conf. on Robotics and Automation*, 1990.

- [LS89] J.P. Laumond and T. Simeon. Motion planning for a two degrees of freedom mobile robot with towing. Technical report, LAAS/CNRS, Report 89148, Toulouse, France, 1989.
- [Mon90] R. Montgomery. Isoholonomic problems and some applications. *Communication in Mathematical Physics*, pages 128:565–592, 1990.
- [MS90] R. Murray and S. Sastry. Grasping and manipulation using multifingered robot hands. In R.W. Brockett, editor, *Robotics: Proceedings of Symposia in Applied Mathematics*, volume 41, pages 91–128. American Mathematical Society, 1990.
- [NM90] Y. Nakamura and R. Mukherjee. Nonholonomic path planning of space robotics via bidirectional approach. In *Proceedings of IEEE Int. Conf. on Robotics and Automation*, 1990.
- [Sus83] H. J. Sussmann. Lie brackets, real analyticity and geometric control. In R. Millman R.W. Brockett and H.J. Sussman, editors, *Differential Geometric Control Theory*. Birkhauser, Boston, 1983.
- [VD87] Z. Vafa and S. Dubowsky. On the dynamics of manipulators in space using the virtual manipulator approach. In *Proceedings of IEEE Int. Conf. on Robotics and Automation*, 1987.
- [WS91] G. Walsh and S. Sastry. On reorienting linked rigid bodies using internal motions. In *IEEE Conf. on Decision and Control*, 1991.

1

Non-holonomic Kinematics and the Role of Elliptic Functions in Constructive Controllability

R. W. Brockett and Liyi Dai

Harvard University

Division of Applied Sciences

Cambridge, MA 02138

1 Introduction

Beginning around 1963 with the paper of Hermann [1] there began to be developed a theory of controllability that included nonlinear effects in a mathematically natural way. By the early 1970's this work was in full swing, with several centers of activity throughout the world. In addition to the study of thermodynamic paths, which were the inspiration for the earlier work of Caratheodory and Chow, a number of other physical problems began to be studied using these techniques. Among these were a set of questions concerning the control of rigid bodies. Some of these are mentioned in reference [2] and several others, including the development of necessary and sufficient conditions for controllability of the Euler equations in body

*This work was supported in part by the National Science Foundation under Engineering Research Center Program, NSF D CDR-8803012 and by the US Army Research Office under grant DAAL03-86-K-0171

fixed, angular velocity space, were discussed in [3] [4] [5]. One could characterize the bulk of this work as being nonconstructive in the sense that the authors gave sufficient conditions for the existence of a control which steered the state from one point to another but did not give a procedure for computing such a control.

Over the last five years there has been a new push to explore these ideas in the context of robotics. Questions of interest range from steering multitrailer vehicles to the manipulation of rigid objects that interact by rolling on each other without slipping. The literature includes Li, Canny and Sastry [6], Murray and Sastry [7] and the work of Montana [8] which summarizes the development of the differential geometry of rolling contact contained in his thesis [9]. Constructive aspects are addressed in [7] and in the recent work of Lafferriere and Sussmann [10]. A related set of ideas is discussed in [11] where the role of nonholonomic kinematics in the design of high force actuators is explored.

Here we consider a kinematic problem involving rigid bodies but mainly we develop a theoretical idea having to do with the role of elliptic functions in the optimization of paths for nonholonomic maneuvers. On one hand this might be viewed as a continuation of earlier work [12] [13] in which the problems are solved in terms of trigonometric functions but it also has much in common with certain examples involving elliptic functions treated by Griffiths [14]. In [12] we discuss in some detail the control problem

$$\begin{aligned}\dot{x} &= u \\ \dot{y} &= v \\ \dot{z} &= xv - yu\end{aligned}$$

Insofar as this example is concerned, the main points established there are:

1. This set of equations is, in a certain sense, generic for the local behavior of driftless, controllable systems with two controls and a three dimensional state space.

2. If we wish to steer this system between $(x(0), y(0), z(0))$ and $(x(1), y(1), z(1))$ while minimizing

$$\eta = \int_0^1 u^2 + v^2 dt$$

then the optimal trajectories can be computed explicitly in terms of trigonometric functions.

3. This system is part of a hierarchy with other members of the hierarchy being obtained by considering systems with more controls and/or a higher dimensional state space with higher order nonlinear effects.

Because the example to be discussed here requires it, we enlarge on this last point. There is a generalization of the $\dot{x} = u$; $\dot{y} = v$; $\dot{z} = xv - yu$ system in which the number of controls is n rather than two. It takes the form

$$\dot{x} = u$$

$$\dot{Z} = xu^T - ux^T$$

with x and u being m -vectors and Z being an m by m skew symmetric matrix. This problem was also discussed in reference [12]. There is a second, and in all respects more intricate extension that involves a higher dimensional state space with controllability being achieved by virtue of higher order Lie brackets. We display this hierarchy for the case of two inputs and indicate, in less detail, the m -input situation.

The number of linearly independent binary forms (i.e. forms involving two variables) that are homogeneous of degree p is, of course, $p + 1$. Thus there are $2(p + 1)$ linearly independent expressions of the form

$$\eta = \phi(x, y)\dot{x} + \psi(x, y)\dot{y}$$

with ϕ and ψ being homogeneous of degree p . Since there are $p + 2$ linearly independent homogeneous forms of degree $p + 1$ in x and y

we see that in the $2(p+1)$ -dimensional linear family of expressions of the form $\phi(x, y)\dot{x} + \psi(x, y)\dot{y}$ there is a $(p+2)$ -dimensional subspace whose elements are of the form $d\gamma/dt$ for some form γ , homogeneous of degree $p+1$. We call these **exact differentials**. Since $2(p+1) - (p+2) = p$ there is a complementary p -dimensional linear family of η 's whose nonzero elements are not integrable. For example, if p is 2 we have the four dimensional integrable family consisting of the linear combinations of the terms

$$\{x^2\dot{x}, 2xy\dot{x} + x^2\dot{y}, 2xy\dot{y} + y^2\dot{x}, y^2\dot{y}\}$$

and we have a complementary two dimensional nonintegrable family which we may think of as consisting of linear combinations of the two terms

$$\{x^2\dot{y}, y^2\dot{x}\}$$

Of course there are other choices of basis for the space of exact differentials and other choices for the complementary space.

Using this as a guide it is not too hard to conclude that for each positive integer p there exists a two input controllable system whose state space is of dimension $(p^2 + p + 4)/2$ and which takes the form

$$\begin{array}{ll} \text{level 0} & \left\{ \begin{array}{lcl} \dot{x} & = & u \\ \dot{y} & = & v \end{array} \right. \\ \text{level 1} & \left\{ \begin{array}{lcl} \dot{z} & = & xv - yu \end{array} \right. \\ \text{level 2} & \left\{ \begin{array}{lcl} \dot{m} & = & x^2v \\ \dot{n} & = & y^2u \end{array} \right. \\ \vdots & \\ \text{level } p & \left\{ \begin{array}{lcl} \dot{a}_{p1} & = & \phi_{p1}u + \psi_{p1}v \\ \dot{a}_{p2} & = & \phi_{p2}u + \psi_{p2}v \\ \vdots & & \\ \dot{a}_{pp} & = & \phi_{pp}u + \psi_{pp}v \end{array} \right. \end{array}$$

Of course, for fixed i , the right-hand sides of the equations for \dot{a}_{ij} are chosen so as to form a basis for a complement of the space of exact forms of degree i . We will consider certain examples of this hierarchy occurring in nonholonomic mechanics. A more general study of integrability of forms, including the type represented by η as a special case is investigated (for different reasons) in [15].

Remark 1: For the sake of completeness we observe that if there are m controls then because there are

$$d_{mp} = \binom{m+p-1}{p}; \text{ (binomial coefficient)}$$

linearly independent homogeneous forms of degree p in m variables, it follows that the level p subsystem is of dimension

$$m d_{mp} - d_{m,p+1} = p \binom{m+p-1}{p+1}$$

Examples are given later.

Remark 2: The m input system for $m = 2, 3, \dots$ has $md_{mp} - d_{m,p+1}$ scalar equations at the p th level and is what one might think of as being the **complete canonical form** for systems of the special form

$$\begin{aligned}\dot{x}_u &= \sum_{i=1}^m g_i(x_u) u_i \\ \dot{x}_l &= \sum_{i=1}^m \tilde{g}_i(x_u) u_i\end{aligned}$$

with x_u being m -dimensional. (That is, the vector fields depend on a set of m variables, x_u , and do not depend on any of the remaining ones). By taking a subset of the given terms at any level one can get a controllable system. In fact, any subsystem of the complete p -level system obtained by omitting equations which appear above the zeroth level is controllable. What makes the set of equations that are given here unique is that one can not add further equations of the given type at any level and still have controllability. Sussmann

has emphasized the role of the Phillip Hall basis in treating the general case (in which the g 's are permitted to depend on x in an arbitrary way). The Phillip Hall basis has cardinality larger than $md_{mp} - d_{m,pH}$ when p is four or more.

We complete this introduction by mentioning a variation on our basic theme that plays a natural role in kinematics. Consider the equations for a control system on the three dimensional orthogonal group taking the form $\dot{A} = (\Omega_1 u_1 + \Omega_2 u_2)A$ where A is orthogonal and

$$\Omega_1 u_1 + \Omega_2 u_2 = \begin{bmatrix} 0 & 0 & -u_2 \\ 0 & 0 & u_1 \\ u_2 & -u_1 & 0 \end{bmatrix}$$

In his thesis [16] John Baillieul carried out a detailed investigation of a certain optimal control problem associated with this system. For our present purposes, however we want to point out that in a coordinate chart near the identity matrix $A = I$ these equations are equivalent to

$$\begin{aligned} \dot{z} &= -u_2 a_{32} + u_1 a_{31} \\ \dot{a}_{13} &= -u_2 a_{33} \\ \dot{a}_{23} &= u_1 a_{33} \end{aligned}$$

where $z = a_{21} + a_{12}$. If we restrict our attention to a neighborhood of the identity then these equations are, to first order in the a_{ij} simply

$$\begin{aligned} \dot{z} &= -u_2 a_{23} + u_1 a_{13} \\ \dot{a}_{13} &= -u_2 \\ \dot{a}_{23} &= -u_1 \end{aligned}$$

In the thesis of Gunther [17] there is a systematic investigation of the symmetries associated with the $\dot{A} = (\Omega_1 u_1 + \Omega_2 u_2)A$ system and related systems on the n -dimensional orthogonal group.

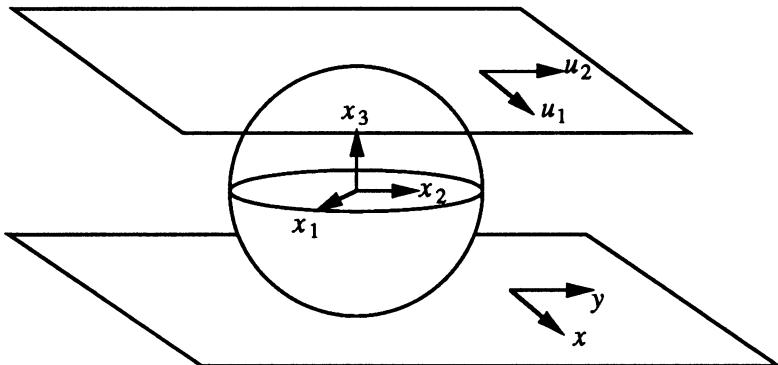


Figure 1: The Plate-Ball Example.

2 The Plate-Ball Problem

We now turn our attention to a specific nonholonomic kinematic problem. It consists of two plates and a ball. The ball rolls without slipping between the plates; the plates are assumed to be flat and horizontal, separated by a distance equal to the diameter of the ball. We imagine that one of the plates is fixed in space and that the two components of the velocity of the other plate in the horizontal plane are the controls. To simplify matters we let the radius of the ball be one.

We let A be the orthogonal matrix whose columns are the coordinates of an orthonormal frame fixed in the ball relative to an orthonormal frame fixed in space and let x and y be the coordinates of the center of the ball. Consider the homogeneous coordinate matrix

$$H = \begin{bmatrix} a_{11} & a_{12} & a_{13} & x \\ a_{21} & a_{22} & a_{23} & y \\ a_{31} & a_{32} & a_{33} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

It satisfies the kinematic relation

$$2\dot{H} = H(\Omega_1 u_1 + \Omega_2 u_2) + (E_1 u_1 + E_2 u_2)H$$

with u_1 being the velocity of the top plate along one axis and u_2 being the velocity of the top plate along the other, where

$$\Omega_1 u_1 + \Omega_2 u_2 = \begin{bmatrix} 0 & 0 & -u_1 & 0 \\ 0 & 0 & -u_2 & 0 \\ u_1 & u_2 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$E_1 u_1 + E_2 u_2 = \begin{bmatrix} 0 & 0 & 0 & u_1 \\ 0 & 0 & 0 & u_2 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Clearly the center of the ball moves at one half the velocity of the top plate. this yields the relation between (\dot{x}, \dot{y}) and (u_1, u_2) . Moreover, in terms of a coordinate system aligned with the x -axis, y -axis, and the vertical, the angular velocity about the x axis is u_1 radians/sec, whereas that about the y axis is u_2 radians/sec.

For our purposes it is more convenient to rewrite the kinematics of this system as

$$\begin{bmatrix} \dot{a}_{11} & \dot{a}_{12} & \dot{a}_{13} \\ \dot{a}_{21} & \dot{a}_{22} & \dot{a}_{23} \\ \dot{a}_{31} & \dot{a}_{32} & \dot{a}_{33} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} 0 & 0 & -u \\ 0 & 0 & -v \\ u & v & 0 \end{bmatrix}$$

$$\begin{aligned} \dot{x} &= u \\ \dot{y} &= v \end{aligned}$$

If $A(0)$ is orthogonal then A remains orthogonal and so this system can be thought of as evolving on a five dimensional manifold.

If we introduce the variables $m = x - a_{13}$ and $n = y - a_{23}$ then from the given equations it follows that

$$\dot{m} = \left(1 - \sqrt{1 - a_{13}^2 - d_{23}^2}\right) u$$

$$\dot{n} = \left(1 - \sqrt{1 - a_{13}^2 - a_{23}^2}\right) v$$

or

$$\begin{aligned}\dot{m} &\approx (a_{13}^2 + a_{23}^2)u \\ \dot{n} &\approx (a_{13}^2 + a_{23}^2)v \\ \dot{a}_{13} &= u \\ \dot{a}_{23} &= v \\ \dot{z} &= ua_{23} - va_{13}\end{aligned}$$

This set is equivalent to the standard form with $p = 2$ introduced in Section 1.

3 Optimization

In this section we confine our discussion to an optimization problem for the system

$$\begin{aligned}\dot{x} &= u \\ \dot{y} &= v \\ \dot{z} &= xv - yu \\ \dot{m} &= x^2v \\ \dot{n} &= y^2u\end{aligned}\tag{1}$$

This system is equivalent to of the type one obtains by keeping the leading terms in the plate-ball example. More specifically, we impose the boundary conditions

$$\begin{aligned}x(0) &= 0; y(0) = 0; z(0) = 0; m(0) = 0; n(0) = 0 \\ x(T) &= x_T; y(T) = y_T; z(T) = z_T; m(T) = m_T; n(T) = n_T\end{aligned}$$

and consider the cost functional

$$\eta = \int_0^T (u^2 + v^2) dt\tag{2}$$

Our goal is to find the controls u and v that steer the system to the desired fixed state and minimize the cost functional (2).

We may rephrase this problem in geometrical terms. Find the shortest curve in the (x, y) -plane joining the origin to $(x(T), y(T))$ and having the following three additional properties:

$$\begin{aligned} z(T) &= \int_0^T (x\dot{y} - y\dot{x}) dt \\ m(T) &= \int_0^T x^2 \dot{y} dt \\ n(T) &= \int_0^T y^2 \dot{x} dt \end{aligned}$$

It may not be immediately clear that the performance measure is equivalent to minimizing arc length but we will see that it is. Notice that in the special case where $x(T) = x(0)$ and $y(T) = y(0)$ the curve in (x, y) -space is closed and $z(t)$ is twice the (signed) area of the interior of the curve. In this case m and n are the x and y -moments associated with the area.

In order to find necessary conditions for optimality we introduce Lagrange multipliers $\lambda_1, \lambda_2, \lambda_3$ and consider minimizing

$$\eta = \int_0^T (\dot{x}^2 + \dot{y}^2 + \lambda_1(x\dot{y} - y\dot{x}) + \lambda_2 x^2 \dot{y} + \lambda_3 y^2 \dot{x}) dt$$

In this case the Euler-Lagrange system is

$$\begin{aligned} \ddot{x} &= (\lambda_1 + \lambda_2 x - \lambda_3 y)\dot{y} \\ \ddot{y} &= -(\lambda_1 + \lambda_2 x - \lambda_3 y)\dot{x} \end{aligned} \tag{3}$$

and the problem becomes that of picking $\dot{x}(0), \dot{y}(0), \lambda_1, \lambda_2, \lambda_3$ in such a way as to meet the five final conditions. Multiplying the first of these by \dot{x} , the second by \dot{y} and adding gives

$$2\dot{x}\ddot{x} + 2\dot{y}\ddot{y} = 0$$

which is equivalent to

$$\dot{x}^2 + \dot{y}^2 = \text{constant}$$

That is the optimal controls satisfy $u^2 + v^2 = \text{constant}$. This explains why minimizing $u^2 + v^2$ is equivalent to minimizing the integral of $(u^2 + v^2)^{1/2}$. Clearly we can write

$$\begin{aligned}\dot{x} &= r \cos(\alpha(t)) \\ \dot{y} &= r \sin(\alpha(t))\end{aligned}\tag{4}$$

where $r \geq 0$ is a constant. By substituting these two equations into (3) and differentiating we see that there exists constants a and ϕ such that $\alpha(t)$ satisfies the following equation

$$\ddot{\alpha}(t) = -r\lambda_2 \cos(\alpha(t)) + r\lambda_3 \sin(\alpha(t)) = a \sin(\alpha(t) - \phi)\tag{5}$$

By setting $\beta(t) = \alpha(t) - \phi$ we get

$$\ddot{\beta}(t) = a \sin(\beta(t))\tag{6}$$

Note that we have five free parameters $r, \phi, a, \beta(0), \dot{\beta}(0)$ which can be chosen to meet the final value condition $(x(T), y(T), z(T), m(T), n(T))$.

It is interesting to study the analytic form of optimal control solution. The following remarks are a guide.

Remark 3: For systems without the $m(t)$ and $n(t)$ terms, it is easy to check that the optimal solution also takes the form of (4) except that in this case $\alpha(t)$ is determined by

$$\dot{\alpha}(t) = \lambda_1\tag{7}$$

or

$$\alpha(t) = \lambda_1 t + \alpha(0)\tag{8}$$

This corresponds to the special case for which $a = 0$ and $\phi = 0$.

Remark 4: The solution of equation (6) can be given in terms of elliptic integrals. Note that by multiplying both sides by $2\dot{\beta}$ the equation for $\ddot{\beta}$ becomes

$$2\dot{\beta}(t)\ddot{\beta}(t) = 2a \sin(\beta(t))\dot{\beta}(t) \quad (9)$$

Integrating this we get

$$(\dot{\beta}(t))^2 = b - 2a \cos(\beta(t)) \quad (10)$$

where b is the constant of integration. If $b - 2|a|$ is positive this can be written as

$$\begin{aligned} \dot{\beta}(t) &= \pm \sqrt{b - 2a \cos(\beta(t))} \\ &= c \sqrt{1 - k \sin^2(\beta(t)/2)} \end{aligned}$$

where c and k are constants. Thus

$$\int_0^{\frac{\beta(t)}{2}} \frac{d\sigma}{\sqrt{1 - k \sin^2(\sigma)}} = \frac{c}{2}t + d \quad (11)$$

where d is a free parameter. The left-hand side of equation (12) is of course an elliptic integral. (See chapters 20 and 22 of [18].)

Remark 5: There is a very general argument, applying to any system of the type discussed in the introduction, showing that along trajectories that minimize the integral square value of the control, the instantaneous value of the magnitude of the control vector is constant. This parallels an argument used in the study of geodesics on a Riemannian manifold whereby one shows that minimizing

$$\eta = \int_0^1 \sqrt{\sum g_{ij}(x) \dot{x}_i \dot{x}_j} dt$$

leads to the same trajectories as minimizing the corresponding integral without the square root.

Remark 6: The trajectories satisfying the Euler-Lagrange equation will be periodic of period T if $x(0) = x(T), y(0) = y(T)$ and $\dot{x}(0) = \dot{x}(T), \dot{y}(0) = \dot{y}(T)$. These latter two conditions can be thought of as fixing two of the λ 's and so we can expect to have a one parameter family of periodic solutions corresponding to generic points in the (x, y) -plane. If $x(0) = 0, y(0) = 1, x(T) = 0, y(T) = 1$ which problems lead to periodic solutions? It may be observed that an $(x(\cdot), y(\cdot))$ path, traversed backwards in time, (i.e. the path $x(T - \cdot), y(T - \cdot)$) generates the negative of the area and the negative of the moments. The path $(-x(T - \cdot), y(T - \cdot))$ generates the negative of the area but only reverses the signs on the y -moment. Thus if $x(0) = 0 = x(T), y(0) = y(T), z(0) = z(T) = 0$ and $m(T) = m(0) = 0$ the trajectory will be periodic. By varying $n(T) - n(0)$ we can generate a one parameter family of periodic solutions passing through $(0, y(0))$. In fact, one can display this family rather explicitly by noting that if $\lambda_1 = \lambda_2 = 0$ then the Euler-Lagrange equation reduce to

$$\ddot{y} = \lambda_3 y \sqrt{c - \lambda_3 y^2}$$

Multiplying by \dot{y} and integrating gives

$$3\dot{y}^2 = -2\sqrt{(c - \lambda_3 y^2)^3} + d$$

which can then be solved for \dot{y} in the standard way.

In order to display some optimal trajectories we consider a particular case. Let $T = 1, k = 1, \phi = 0$, and let $x(0) = x(1) = 0, y(0) = y(1) = 0$. In this case, by solving (10) and (11) we may get the optimal trajectory

$$\begin{aligned} x &= rt + \frac{4r}{b} \left(\frac{1}{1+a^2 e^{2bt}} - \frac{1}{1+a^2} \right) \\ y &= \frac{4ar}{b} \left(\frac{e^{bt}}{1+a^2 e^{2bt}} - \frac{1}{1+a^2} \right) \end{aligned}$$

where $b = -3.83, a = \pm e^{(b/2)}$ are chosen to meet $x(1) = y(1) = 0$, r is a free parameter. Optimal trajectories corresponding to different choices of a and r are shown in Figure 2.

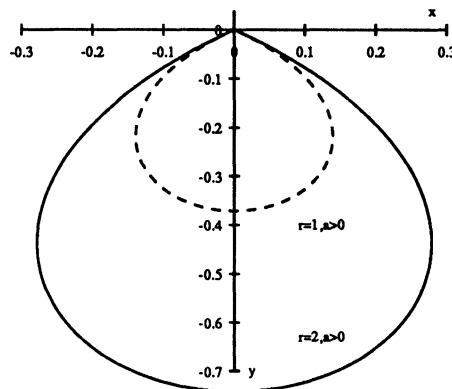


Figure 2: The Optimal trajectory for an Example

In figure 3 we show typical trajectories generated numerically, starting at $x(0) = 0, y(0) = 0$.

4 Symmetries

Because the optimization problem being considered relates to the arc length of a curve in euclidean space and because arc length is invariant under a euclidean transformation, one can use this group of transformations to simplify the investigation of optimal trajectories. We illustrate how this works leaving most of the details and generalization to the reader.

Let (x', y') be related to (x, y) by

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} a \\ b \end{bmatrix}$$

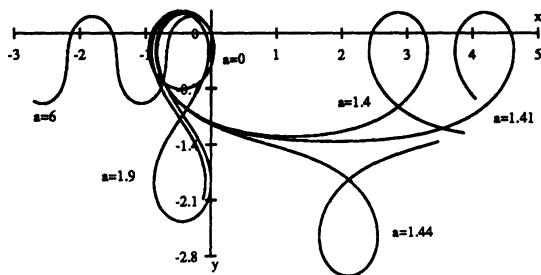


Figure 3: Solutions of equations (4)-(6);
 $x(0)=0; y(0)=0; \beta(0)=\dot{\beta}(0)=2; \phi=0.$

How can we define z', m', n' such that the vector (x', y', z', m', n') satisfies

$$\begin{aligned} \dot{x}' &= u \\ \dot{y}' &= v \\ \dot{z}' &= x'v - y'u \\ \dot{m}' &= (x')^2v \\ \dot{n}' &= (y')^2u \end{aligned}$$

Consider the case $\theta = 0$. An easy calculation shows that if we let

$$\begin{aligned} x' &= x + a \\ y' &= y + b \\ z' &= z + ay - bx \\ m' &= m + a^2y + az + axy \\ n' &= n + b^2x - bz + bxy \end{aligned}$$

then the equation for primed quantities is the standard form displayed above. The effect of the rotational part can also be worked

out. In this case one wants to replace u and v by

$$\begin{bmatrix} u' \\ v' \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix}$$

which does not, of course, change the length of the control.

5 Generalizations

In this section, we will consider some generalizations of the optimal control results discussed above. First we consider the an example that includes the third order terms in x and y . The complete system takes the form:

$$\begin{aligned} \dot{x} &= u \\ \dot{y} &= v \\ \dot{z} &= xv - yu \\ \dot{m} &= x^2v \\ \dot{n} &= y^2u \\ \dot{f} &= x^3v \\ \dot{g} &= y^3u \\ \dot{h} &= x^2yu + xy^2v \end{aligned} \tag{12}$$

This problem may be treated in the same way as we treated the examples of section 3. The Euler Lagrange equations are

$$\begin{aligned} 2\ddot{x} + (-2\lambda_1 - 2\lambda_2x + 2\lambda_3y - 3\lambda_4x^2 + 3\lambda_5y^2 + \lambda_6(x^2 - y^2))\dot{y} &= 0 \\ 2\ddot{y} - (-2\lambda_1 - 2\lambda_2x + 2\lambda_3y - 3\lambda_4x^2 + 3\lambda_5y^2 - \lambda_6(x^2 - y^2))\dot{x} &= 0 \end{aligned} \tag{13}$$

As before, we can show directly that $u^2 + v^2$ is constant. Thus the optimal controls can again be expressed by equation (4). In this case the function $\alpha(t)$ is determined by the following more complicated equation

$$\begin{aligned} 2\dot{\alpha}(t) &= -2\lambda_1 - 2\lambda_2 \int_0^t r \cos(\alpha(t))dt + 2\lambda_3 \int_0^t r \sin(\alpha(t))dt \\ &\quad + (\lambda_6 - 3\lambda_4 [\int_0^t r \cos(\alpha(t))dt]^2 - (\lambda_6 - 3\lambda_5 [\int_0^t r \sin(\alpha(t))dt]^2) \end{aligned}$$

or equivalently,

$$2\ddot{\alpha}(t) = \eta_1 \sin(\alpha(t) + \phi) + \eta_2 \sin(\alpha(t)) \int_0^t \sin(\alpha(\sigma)) d\sigma + \eta_3 \cos(\alpha(t)) \int_0^t \cos(\alpha(\sigma)) d\sigma$$

where $\eta_1, \eta_2, \eta_3, \phi$ are constants.

Again in this case elliptic functions arise. For example, if $\lambda_2 = \lambda_4 = \lambda_6 = 0$ in equation (14), then we can solve $\dot{x} = c + \lambda_1 y - \frac{1}{2}\lambda_3 y^2 - \frac{1}{2}\lambda_5 y^3$ to obtain

$$2\ddot{y} + (2\lambda_1 - 2\lambda_3 y - 3\lambda_5 y^2)(c + \lambda_1 y - \frac{1}{2}\lambda_3 y^2 - \frac{1}{2}\lambda_5 y^3) = 0$$

this can be solved in terms of hyper-elliptic functions.

Finally consider the generation to 3 dimensional inputs. According to our earlier development the complete level 2 system is

$$\begin{aligned} \dot{x}_1 &= u_1 \\ \dot{x}_2 &= u_2 \\ \dot{x}_3 &= u_3 \\ \dot{y}_1 &= x_1 u_2 - x_2 u_1 \\ \dot{y}_2 &= x_1 u_3 - x_3 u_1 \\ \dot{y}_3 &= x_2 u_3 - x_3 u_2 \\ \dot{z}_1 &= x_1^2 u_2 \\ \dot{z}_2 &= x_1^2 u_3 \\ \dot{z}_3 &= x_2^2 u_1 \\ \dot{z}_4 &= x_2^2 u_3 \\ \dot{z}_5 &= x_3^2 u_1 \\ \dot{z}_6 &= x_3^2 u_2 \\ \dot{z}_7 &= x_1 x_2 u_3 \\ \dot{z}_8 &= x_2 x_3 u_1 \end{aligned} \tag{14}$$

where, for the sake of convenience, we have adopted a new notation. In this system, u_1, u_2, u_3 are controls. The problem is that of finding u_1, u_2, u_3 such that the cost functional

$$\eta = \int_0^T (u_1^2 + u_2^2 + u_3^2) dt \quad (15)$$

is as small as possible given that $(x_1, x_2, x_3, y_1, y_2, y_3, z_1, \dots, z_8)$ is initially zero and is to take on a specific value at $t = T$.

The Euler Lagrange system is

$$\begin{aligned} 2\ddot{x}_1 - & (2\lambda_1 + 2\eta_1 x_1 - 2\eta_3 x_2 - \eta_8 x_3)\dot{x}_2 - \\ & (2\lambda_2 + 2\eta_2 x_1 - 2\eta_5 x_3 - (\eta_8 - \eta_7)x_2)\dot{x}_3 = 0 \\ 2\ddot{x}_2 + & (2\lambda_1 + 2\eta_1 x_1 - 2\eta_3 x_2 - \eta_8 x_3)\dot{x}_1 - \\ & (2\lambda_3 + 2\eta_4 x_2 - 2\eta_6 x_3 + \eta_7 x_1)\dot{x}_3 = 0 \\ 2\ddot{x}_3 + & (2\lambda_2 + 2\eta_2 x_1 - 2\eta_5 x_3 - (\eta_8 - \eta_7)x_2)\dot{x}_1 + \\ & (2\lambda_3 + 2\eta_4 x_2 - 2\eta_6 x_3 + \eta_7 x_1)\dot{x}_2 = 0 \end{aligned} \quad (16)$$

where λ_i, η_i are Lagrange multipliers. From equation (16) it is easy to see that x_1, x_2, x_3 satisfy the following equation:

$$2\dot{x}_1\ddot{x}_1 + 2\dot{x}_2\ddot{x}_2 + 2\dot{x}_3\ddot{x}_3 = 0$$

or

$$\dot{x}_1^2 + \dot{x}_2^2 + \dot{x}_3^2 = R$$

with $R \geq 0$ being a constant. Therefore, we can express $u_i = \dot{x}_i$ in the form

$$\begin{aligned} u_1 &= R \cos(\alpha_1(t)) \cos(\alpha_2(t)) \\ u_2 &= R \sin(\alpha_1(t)) \cos(\alpha_2(t)) \\ u_3 &= R \sin(\alpha_2(t)) \end{aligned} \quad (17)$$

By substituting equation (17) into (16), we know that $\alpha_1(t)$ and $\alpha_2(t)$ are determined by the the following equation

$$\begin{aligned} \dot{\alpha}_1(t) = & -\frac{1}{2}(2\lambda_1 + 2\eta_1 x_1 - 2\eta_3 x_2 - \eta_8 x_3) \\ & -\frac{1}{2}(2\lambda_2 + 2\eta_2 x_1 - 2\eta_5 x_3 - (\eta_8 - \eta_7)x_2) \tan(\alpha_2(t)) \sin(\alpha_1(t)) \\ & +\frac{1}{2}(2\lambda_3 + 2\eta_4 x_2 - 2\eta_6 x_3 + \eta_7 x_1) \tan(\alpha_2(t)) \cos(\alpha_1(t)) \\ \dot{\alpha}_2(t) = & -\frac{1}{2}(2\lambda_2 + 2\eta_2 x_1 - 2\eta_5 x_3 - (\eta_8 - \eta_7)x_2) \cos(\alpha_1(t)) \\ & -\frac{1}{2}(2\lambda_3 + 2\eta_4 x_2 - 2\eta_6 x_3 + \eta_7 x_1) \sin(\alpha_1(t)) \end{aligned} \quad (18)$$

where

$$x_i = \int_0^t u_i dt$$

Again, elliptic functions can arise. To see this point we repeat the method used before: setting $\lambda_1 = \eta_1 = \eta_2 = \eta_3 = \eta_4 = \eta_7 = \eta_8 = 0$ in (18) we obtain

$$\begin{aligned}\ddot{x}_1 - (\lambda_2 - \eta_5 x_3) \dot{x}_3 &= 0 \\ \ddot{x}_2 - (\lambda_3 - \eta_6 x_3) \dot{x}_3 &= 0 \\ \ddot{x}_3 + (\lambda_2 - \eta_5 x_3) \dot{x}_1 + (\lambda_3 - \eta_6 x_3) \dot{x}_2 &= 0\end{aligned}$$

from which we can solve

$$\begin{aligned}\dot{x}_1 &= c_1 + \lambda_2 x_3 - \frac{1}{2} \eta_5 x_3^2 \\ \dot{x}_2 &= c_2 + \lambda_3 x_3 - \frac{1}{2} \eta_6 x_3^2\end{aligned}$$

and

$$\ddot{x}_3 + (\lambda_2 - \eta_5 x_3)(c_1 + \lambda_2 x_3 - \frac{1}{2} \eta_5 x_3^2) + (\lambda_3 - \eta_6 x_3)(c_2 + \lambda_3 x_3 - \frac{1}{2} \eta_6 x_3^2) = 0$$

which can be solved in terms of elliptic functions.

6 Acknowledgement

The first author would like to thank Shankar Sastry for suggesting the possibility of applying this kind of analysis in robotics and Hector Sussmann for pointing out an error in our original statement of Remark 2. We would like to thank Ann Stokes for her help with the Plate-Ball Problem.

References

- [1] Hermann, Robert, "On the Accessibility Problem in Control Theory" in *Nonlinear Differential Equations and Nonlinear Mechanics*, J. P. Lasalle and S. Lefschetzy Eds., Academic Press, New York, 1963.

- [2] Brockett, R. W., "System Theory on Group Manifolds and Coset Spaces," *SIAM J. on Control*, Vol. 10 pp.265–284 1972.
- [3] Brockett, R. W., "Nonlinear Systems and Differential Geometry," *Proceedings of the IEEE*, Vol. 64 pp. 61–72 1976..
- [4] Baillieul, J., "The Geometry of Homogeneous Polynormal Dynamical Systems", *Nonlinear Analysis, Theory, Methods and Applications*, Vol. 4, No. 5, pp. 879–900, 1980.
- [5] Crouch, P.E., "Spacecraft Attitude Control and Stabilization", *IEEE Trans. on Automatic Control*, AC29, pp. 321–331 1984.
- [6] Li, Z., Canny, J. F., Sastry, S. S., "Motion Planning for Dextrous Manipulation Part 1: The Problem Formulation" *Proceedings of the 1989 IEEE International Conference on Robotics and Automation*, pp. 775–781.
- [7] Murray, R. M. and Sastry, S. S., "Steering Nonholonomic Systems using Sinusoids", *Procd. 29th IEEE Conf. on Decision and Control*, Honolulu, Hawaii
- [8] Montana, D., "The Kinematics of Grasp", *International J. of Robotics*, Vol. 7, No. 3, pp. 17–31 1988.
- [9] Montana, D., *Tactile Sensing and the Kinematics of Contact*, Ph.D. Thesis, Harvard University, 1986.
- [10] Lafferriere, G. and Sussmann, H. J., "Motion Planning for Controllable Systems without Drift: A preliminary Report", Rutgers Center for Systems and Control, Report, June 1990.
- [11] Brockett, R. W., "On the Rectification of Vibratory Motion," *Sensors and Actuators*, Vol. 20, pp. 91–96 1989.
- [12] Brockett, R. W., "Control Theory and Singular Riemannian Geometry," in *New Directions in Applied Mathematics* (P. Hilton and G. Young, eds.). New York: Springer-Verlag, pp. 11–27 1981.

- [13] Brockett, R. W., "Nonlinear Control Theory and Differential Geometry," *Proceedings of the International Congress of Mathematicians*, Polish Scientific Publishers, Warszawa, pp. 1357–1367 1984.
- [14] Phillip A. Griffiths, *Exterior Differential Systems and the Calculus of Variations*, Birkhauser, Boston, 1983.
- [15] Brockett, R. W., "Optimal Linear Systems with Polynomial Performance Measures," *1977 Joint Automatic Control Conference*, IEEE, New York, 1977
- [16] Baillieul, J., *Some Optimization Problems in Geometric Control Theory*, Ph.D. Thesis, Harvard University, 1975.
- [17] Gunther, N., *Hamiltonian Mechanics and Optimal Control*, Ph.D. Thesis, Harvard University, 1982.
- [18] Whittaker, E. T. and Watson, G.N., *Modern Analysis*, Cambridge University Press, Cambridge England, 4th Edition, 1927.

2

Steering Nonholonomic Control Systems Using Sinusoids*

Richard M. Murray[†]

Department of Mechanical Engineering
California Institute of Technology
Pasadena, CA 91125
(818) 356-6460
E-mail: murray@design.caltech.edu

S. Shankar Sastry

Electronics Research Laboratory
Department of Electrical Engineering
and Computer Sciences
University of California
Berkeley, CA 94720

October 21, 1991

Abstract

This paper revises and extends our earlier work in using sinusoids to steer systems with nonholonomic constraints. We show that simple sinusoidal input trajec-

*Research supported in part by the National Science Foundation under grant IRI-90-14490

[†]This research was performed while the author was at the University of California, Berkeley

tories are not easily applied to some classes of nonholonomic systems. This leads to the definition of a form of systems which *can* be steered using our earlier methods. We describe this form in detail and present preliminary efforts towards understanding when systems can be converted into this form.

1 Introduction

We consider systems of the form

$$\dot{x} = g_1(x)u_1 + \cdots + g_m(x)u_m \quad x \in \mathbb{R}^n, \quad u \in \mathbb{R}^m \quad (1)$$

with $\{g_i\}$ a set of smooth, linearly independent vector fields in some neighborhood of the origin. Given an initial point x_0 and a final point x_1 , we wish to find $u(t)$, $t \in [0, 1]$ which *steers* the system from x_0 to x_1 . That is, if $x(0) = x_0$, then under application of u , $x(1) = x_1$.

The problem of steering such systems occurs in several contexts in the robotics literature. Our own involvement arose through the study of dynamic regrasping using multi-fingered robot hands [MS90a, LC90]. Other applications can be found in the areas of mobile robots and space manipulators (see [Mur90] for a recent bibliography). The general formulation of the problem includes a specification of obstacles which are to be avoided, in addition to moving from start to goal.

The conditions for the existence of a path between two configurations is given by Chow's theorem. We let $[f, g]$ be the Lie bracket between two vector fields,

$$[f, g] = \frac{\partial g}{\partial x} f - \frac{\partial f}{\partial x} g,$$

and define the involutive closure of a distribution Δ as the closure of Δ under Lie bracketing. Briefly, Chow's theorem states that if the involutive closure of the distribution associated with equation (1) spans \mathbb{R}^n at each configuration, the system can be steered

between any two configurations. It is not apparent how the path can be explicitly constructed; in this paper we propose techniques for generating such paths.

1.1 Classification

We briefly develop some concepts which allow us to classify non-holonomic systems. A more complete treatment can be found in the work of Vershik [GV88, VG88]. Basic facts concerning Lie algebras are taken from Varadarajan [Var84]. Let $\Delta = \text{span}\{g_1, \dots, g_m\}$ be the distribution associated with the control system (1). Define $G_1 = \Delta$ and

$$G_i = G_{i-1} + [G_1, G_{i-1}]$$

where

$$[G_1, G_{i-1}] = \text{span}\{[g, h] : g \in G_1, h \in G_{i-1}\}$$

The set of all G 's defines the *filtration* associated with a distribution. Each G_i is spanned by the input vector fields plus the vector fields formed by taking up to $i - 1$ Lie brackets. The Jacobi identity implies $[G_i, G_j] \subset [G_1, G_{i+j-1}] \subset G_{i+j}$.

A filtration is *regular* in a neighborhood U of x_0 if

$$\text{rank } G_i(x) = \text{rank } G_i(x_0) \quad \forall x \in U$$

We say a system is regular if the corresponding filtration is regular. If a filtration is regular, then at each step of its construction, G_i either gains dimension or the construction terminates. If $\text{rank } G_{i+1} = \text{rank } G_i$ then G_i is involutive and hence $G_{i+j} = G_i$ for all $j \geq 0$. Clearly $\text{rank } G_i \leq n$ and hence if a filtration is regular, then there exists an integer $p < n$ such that $G_i = G_{p+1}$ for all $i \geq p + 1$. We refer to p as the *degree of nonholonomy* of the distribution.

For a regular system, Chow's theorem is particularly easy to prove.

Theorem 1 *For a regular system, a path exists between two arbitrary points in an open set $U \subset M$ if and only if $G_p(x) = T_x M \approx \mathbb{R}^n$ for all $x \in U$.*

A system (or distribution) satisfying the conditions of the theorem is said to be *maximally nonholonomic*. This version of Chow's theorem is considerably weaker than other versions, which hold for non-regular systems. If a regular system is not maximally nonholonomic, then by Frobenius' theorem we can restrict ourselves to a manifold on which the system is maximally nonholonomic.

It is also useful to record the dimension of each G_i . For a regular system, we define the *growth vector* $r \in \mathbb{Z}^{p+1}$ as

$$r_i = \text{rank } G_i$$

We define the *relative growth vector* $\sigma \in \mathbb{Z}^{p+1}$ as $\sigma_i = r_i - r_{i-1}$ and $r_0 := 0$. The growth vector for a system is a convenient way to represent information about the associated control Lie algebra.

For a distribution with finite rank, the growth vector is bounded from above at each step. To properly determine this bound, we must determine the rank of G_i taking into account skew-symmetry and the Jacobi identity. A careful calculation [Ser65] gives

$$\bar{\sigma}_i = \frac{1}{i} \left((\bar{\sigma}_1)^i - \sum_{j|i, j < i} j \bar{\sigma}_j \right) \quad i > 1 \quad (2)$$

where $\bar{\sigma}_i$ is the maximum relative growth at the i^{th} stage and $j|i$ means all integers j such that j divides i . If $\sigma_i = \bar{\sigma}_i$ for all i , we say Δ has *maximum growth*.

2 Sinusoidal steering

In this section we review the results presented in [MS90b]. We motivate the use of sinusoidal inputs as a method for steering non-holonomic systems through the use of several examples.

2.1 First degree systems

Control systems in which the first level of brackets together with the input vector fields span the tangent space at each configuration arise

in many areas. In classical mechanics, systems with growth vector $r = (n-1, n)$ are called contact structures [Arn89]. A version of the Darboux theorem asserts that for these systems the corresponding constraint can be written as

$$dx_3 = x_2 dx_1$$

(using the notation of exterior differential forms). In \mathbf{R}^3 and using control system form, this becomes

$$\begin{aligned}\dot{x}_1 &= u_1 \\ \dot{x}_2 &= u_2 \\ \dot{x}_3 &= x_2 u_1\end{aligned}\tag{3}$$

Brockett considered a more general version of this system [Bro81]; we review his results here. Consider a control system as in equation (1) that is maximally nonholonomic with growth vector $(m, n) = (m, \frac{m(m+1)}{2})$. We would like to find an input $u(t)$ on the interval 0 to 1 which steers the system between an arbitrary initial and final configuration and minimizes

$$\int_0^1 |u|^2 dt$$

This problem is related to finding the geodesics associated with a singular Riemannian metric (Carnot-Caratheodory metric).

To solve the problem, Brockett considers a class of systems which have a special canonical form. An equivalent form, which is more useful for our purposes, is

$$\begin{aligned}\dot{x}_i &= u_i & i = 1, \dots, m \\ \dot{x}_{ij} &= x_i u_j & i < j\end{aligned}\tag{4}$$

We see that if $m = 2$, this is exactly the contact system (3). It can be shown that the input vector fields and their pairwise brackets span \mathbf{R}^n and hence the system is controllable with degree of nonholonomy equal to 1.

To find the optimal input between two points, we construct the Lagrangian

$$L(x, \dot{x}) = \sum_{i=1}^m \dot{x}_i^2 + \sum_{i,j} \lambda_{ij} (\dot{x}_{ij} - x_i u_j)\tag{5}$$

Here we have used the fact that $u_i = \dot{x}_i$. The λ_{ij} 's are the Lagrangian multipliers associated with the constraint imposed by the control system. Substituting equation (5) into the Euler-Lagrange equation

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{x}} - \frac{\partial L}{\partial x} = 0$$

it can be shown that the input must satisfy

$$u = e^{\Lambda t} u_0$$

where Λ is a constant skew-symmetric matrix. Thus the inputs are sinusoids at various frequencies. Unfortunately, even for very simple problems, determining Λ and u_0 given an initial and final configuration is very difficult.

A great deal of simplification occurs if we consider moving between configurations where $x_i(1) = x_i(0)$. In this instance the eigenvalues of Λ must be multiples of 2π and Brockett showed that the optimal inputs are sinusoids at integrally related frequencies, namely $2\pi, 2 \cdot 2\pi, \dots, \frac{m}{2} \cdot 2\pi$. This simplifies the problem tremendously and for many examples reduces the search to that of finding u_0 . We use this result to propose the following algorithm for steering systems of this type:

Algorithm 1

1. Steer the x_i 's to their desired values using any input and ignoring the evolution of the x_{ij} 's.
2. Using sinusoids at integrally related frequencies, find u_0 such that the input steers the x_{ij} 's to their desired values. By the choice of input, the x_i 's are unchanged.

The resulting trajectories are suboptimal but easily computable and have several nice properties which we will explore.

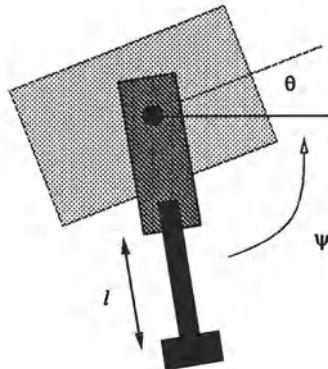


Figure 1: A simple hopping robot. The robot consists of a leg which can both rotate and extend. The configuration of the mechanism is given by the angle of the body and the angle and length (extension) of the leg.

Example 1

We consider as an example a kinematic hopping robot, as shown in Figure 1. This example has been studied by Li, Montgomery and Raibert [LMR89] using holonomy methods. We wish to reorient the body of robot while in midair and bring the leg rotation and extension to a desired final value. The kinematic equations of the robot (in center of mass coordinates) can be written as

$$\begin{aligned}\dot{\psi} &= u_1 \\ \dot{l} &= u_2 \\ \dot{\theta} &= -\frac{m_l(l+1)^2}{1+m_l(l+1)^2} u_1\end{aligned}$$

where we have used units such that the mass of the body is 1 and the length of the leg at zero extension is also 1. The last equation is a consequence of conservation of angular momentum. Expanding the equation using a Taylor series about $l = 0$

$$\dot{\theta} = -\frac{m_l}{1+m_l} \dot{\psi} - \frac{2m_l}{(1+m_l)^2} lu_1 + o(l)u_1$$

This suggests a change of coordinates, $\alpha = \theta + \frac{m_l}{1+m_l}\psi$ to put the equations in the form

$$\begin{aligned}\dot{\psi} &= u_1 \\ \dot{l} &= u_2 \\ \dot{\alpha} &= \frac{2m_l}{(1+m_l)^2}lu_1 + o(l)u_1 = f(l)u_1\end{aligned}$$

This equation has the same form locally as the canonical system in equation (4).

Using this as justification, we attempt to use our proposed algorithm to steer the full nonlinear system. Since we control the ψ and l states directly, we first steer them to their desired values. Then using sinusoids in the ψ and l inputs,

$$\begin{aligned}u_1 &= a_1 \sin \omega t \\ u_2 &= a_2 \cos \omega t\end{aligned}$$

we steer θ to its desired value. By construction, this last motion does not affect the final values of ψ and l . To include the effect of nonlinearity in the first vector field, harmonic analysis can be used. Since l is periodic, we expand f using its Fourier series,

$$f\left(\frac{a_2}{\omega} \sin \omega t\right) = \beta_1 \sin \omega t + \beta_2 \sin 2\omega t + \dots$$

Integrating $\dot{\alpha}$ over one period, only the first term in the expansion contributes to the net motion

$$\begin{aligned}\alpha\left(\frac{2\pi}{\omega}\right) &= \alpha(0) + \int_0^{\frac{2\pi}{\omega}} \left(\frac{a_1 \beta_1}{\omega} \sin^2 \omega t + \frac{a_1 \beta_2}{\omega} \sin \omega t \sin 2\omega t + \dots \right) dt \\ &= \alpha(0) + \frac{\pi a_1 \beta_1}{\omega^2}\end{aligned}$$

Figure 2 shows the trajectory for the last motion segment; ψ and l return to their initial values but α (and hence θ) experiences a net change. To compute the required input amplitudes, we plot β_1 as a function of a_2 and choose a_2 such that $\frac{\pi a_1 \beta_1}{\pi \omega} = \theta_1 - \theta_0$. Using this procedure, we can (locally) steer between any two configurations.

2.2 Second degree systems

We next consider systems in which the first level of bracketing is not enough to span \mathbf{R}^n . We begin by trying to extend the previous

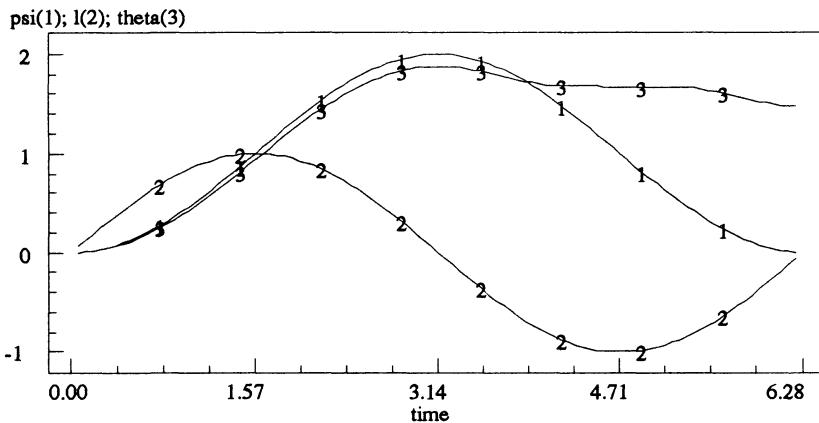


Figure 2: Nonholonomic motion for a hopping robot. Using sinusoidal inputs, the leg angle and extension return to their starting values but the body angle goes a net rotation.

canonical form to the next higher level of bracketing. Consider a system which can be expressed as

$$\begin{aligned}\dot{x}_i &= u_i \quad i = 1, \dots, m \\ \dot{x}_{ij} &= x_i u_j \quad i < j \\ \dot{x}_{ijk} &= x_{ij} u_k \quad (\text{mod Jacobi identity})\end{aligned}\tag{6}$$

Because Jacobi's identity imposes relations between certain brackets, not all x_{ijk} combinations are possible. This is analogous to limiting the x_{ij} 's to those for which $i < j$, reflecting skew-symmetry of the Lie bracket.¹ Using the calculation in equation (2) shows that this system has relative growth vector $(m, \frac{m(m-1)}{2}, \frac{(m+1)m(m-1)}{3})$. Constructing the Lagrangian (with the same integral cost function) and substituting into the Euler-Lagrange equations does not result in a constant set of Lagrange multipliers. As a consequence, we cannot solve the optimal control problem in closed form.

We can however extend and apply our previous algorithm as follows:

Algorithm 2

¹This is made more explicit in Section 3.

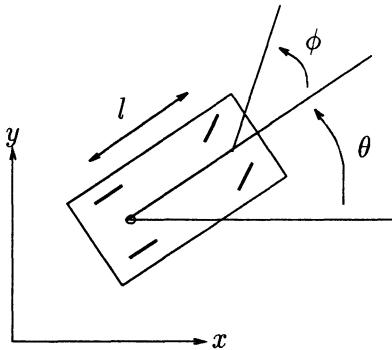


Figure 3: Front wheel drive cart. The configuration of the cart is determined by the Cartesian location of the back wheels, the angle the car makes with the horizontal and the steering wheel angle relative to the car body. The two inputs are the velocity of the front wheels (in the direction the wheels are pointing) and the steering velocity.

1. Steer the x_i 's to their desired values. This causes drift in all other states.
2. Steer the x_{ij} 's to their desired values using integrally related sinusoidal inputs. If the i^{th} input has frequency w_i then x_{ij} will have frequency components at $w_i \pm w_j$. By choosing inputs such that we get frequency components at zero, we can generate motion in the desired states.
3. Use sinusoidal inputs a second time to move all previously steered states in a closed loop and generate motion only in the x_{ijk} directions. This requires careful choice of the input frequencies so that $w_i \pm w_j \neq 0$ but $w_i \pm w_j \pm w_k$ has zero frequency components.

Example 2

To illustrate the algorithm, we consider the motion of a front wheel drive car as shown in Figure 3. The kinematics of this mechanism

can be written as

$$\begin{aligned}\dot{x} &= \cos \theta u_1 \\ \dot{y} &= \sin \theta u_1 \\ \dot{\phi} &= u_2 \\ \dot{\theta} &= \frac{1}{l} \tan \phi u_1\end{aligned}\tag{7}$$

In this form, u_1 does not control any state directly. We use a change of coordinates and a change of input to put the equations in the form

$$\begin{aligned}\dot{x} &= v_1 & v_1 &= \cos \theta u_1 \\ \dot{\phi} &= v_2 & v_2 &= u_2 \\ \dot{\alpha} &= \tan \phi v_1 & \alpha &= \sin \theta \\ \dot{y} &= \frac{\alpha}{\sqrt{1-\alpha^2}} v_1\end{aligned}$$

As before, the linear portion of the nonlinearities matches the canonical system and we can include the effects of the nonlinearities using Fourier series techniques.

An example of the algorithm applied to the car is given in Figure 4. The first portion of the path, labeled A, drives the x and ϕ states to their desired values using a constant input. The second portion, labeled B, uses a periodic input to drive θ while bringing the other two states back to their desired values. The last step brings y to its desired value and returns the other three states to their correct values. The Lissajous figures that are obtained from the phase portraits of the different variables are quite instructive. Consider the portion of the curve labeled C. The upper left plot contains the Lissajous figure for x, ϕ (two loops); the lower left plot is the corresponding figure for x, θ (one loop) and the open curve in x, y shows the increment in the y variable. The very powerful implication here is that the Lie bracket directions correspond to rectification of harmonic periodic motions of the driving vector fields and the harmonic relations are determined by the degree of the Lie bracket corresponding to the desired direction of motion. This point has also been made rather elegantly by Brockett [Bro89] in the context of the rectification of mechanical motion.

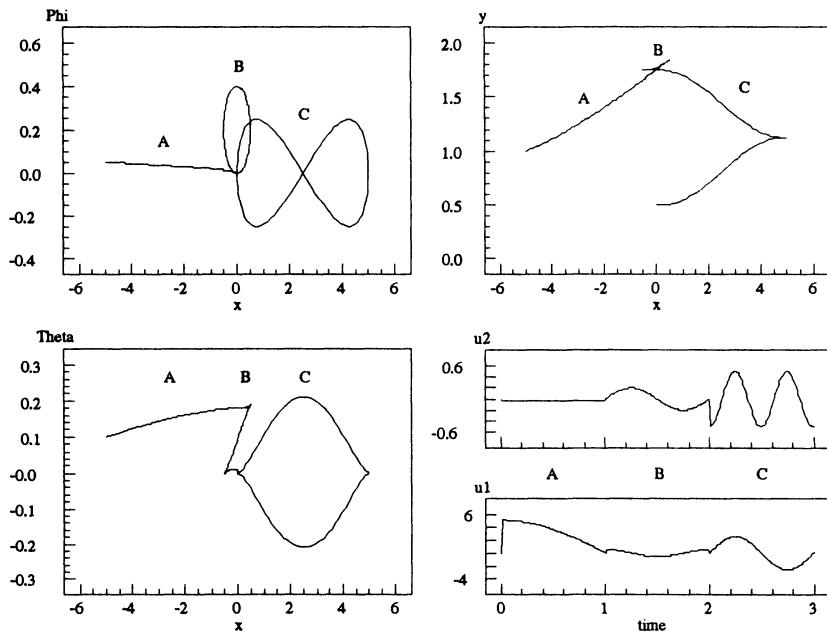


Figure 4: Sample trajectories for a car. The trajectory shown is a three stage path which moves the unicycle from $x = -5$, $y = 1$, $\theta = 0.05$, $\phi = 1$) to $(0, 0.5, 0, 0)$. The first three figures show the states versus x ; the bottom right figures show the inputs as functions of time.

3 Higher order canonical systems

We now study more general examples of nonholonomic systems and investigate the use of sinusoids for steering such systems. As before, we begin by studying a class of canonical examples and then attempt to extend the analysis to non-canonical systems.

To construct systems with a given number of inputs and degree of nonholonomy, it is necessary to introduce some additional machinery. As mentioned earlier, in constructing canonical systems we must observe the fundamental restrictions imposed by the Lie bracket: skew-symmetry and the Jacobi identity. Our search for a set of vector fields which have a given degree of nonholonomy is equivalent to searching for a basis for an abstract, finite-dimensional Lie algebra. One such basis is P. Hall basis [Hal59, Ser65] (see

also [LS90] for a more detailed description).

3.1 P. Hall Bases

Given a set of generators $\{X_1, \dots, X_m\}$, we define the length of a Lie product recursively as

$$\begin{aligned} l(X_i) &= 1 & i = 1, \dots, m \\ l([A, B]) &= l(A) + l(B) \end{aligned}$$

where A and B are themselves Lie products. This length function induces a partial ordering on a set of Lie products. A Lie algebra is *nilpotent* if there exists an integer k such that all Lie products of length greater than k are zero. k is called the degree of nilpotency. A nilpotent Lie algebra is finite-dimensional. A *P. Hall basis* is an ordered set of Lie products $H = \{B_i\}$ satisfying

- (PH1) $X_i \in H$, $i = 1, \dots, m$
- (PH2) If $l(B_i) < l(B_j)$ then $B_i < B_j$
- (PH3) $[B_i, B_j] \in H$ if and only if
 - (a) $B_i, B_j \in H$ and $B_i < B_j$ and
 - (b) either $B_j = X_k$ for some k or
 $B_j = [B_l, B_r]$ with $B_l, B_r \in H$ and $B_l \leq B_i$

The proof that a P. Hall basis is a basis for the free Lie algebra generated by $\{X_1, \dots, X_m\}$ can be found in [Hal59, Ser65].

A P. Hall basis with degree of nilpotency k can be constructed from a set of generators using the definition. The simplest approach is to construct all possible Lie products with length less than k and use the definition to eliminate elements which fail to satisfy one of the properties. In practice, the basis can be built in such a way that only (PH3) need be checked. A Mathematica program which performs this calculation is given in [Mur90].

Examples

A basis for the nilpotent Lie algebra of degree 3 generated by $\{X, Y, Z\}$ is

$$\begin{array}{ccc} X & Y & Z \\ [X, Y] & [X, Z] & [Y, Z] \\ [X, [X, Y]] & [X, [X, Z]] & [Y, [X, Y]] \quad [Y, [X, Z]] \\ [Y, [Y, Z]] & [Z, [X, Y]] & [Z, [X, Z]] \quad [Z, [Y, Z]] \end{array}$$

Note that $[X, [Y, Z]]$ does not appear since

$$[X, [Y, Z]] + [Y, [Z, X]] + [Z, [X, Y]] = 0$$

and two of the three terms are already present.

A larger example, which we will use in the sequel, is a basis for a Lie algebra of degree 5 with 2 generators:

$$\begin{aligned} B_1 - B_2 : \quad & X \quad Y \\ B_3 : \quad & [X, Y] \\ B_4 - B_5 : \quad & [X, [X, Y]] \quad [Y, [X, Y]] \\ B_6 - B_8 : \quad & [X, [X, [X, Y]]] \quad [Y, [X, [X, Y]]] \quad [Y, [Y, [X, Y]]] \\ B_9 - B_{14} : \quad & [X, [X, [X, [X, Y]]]] \quad [Y, [X, [X, [X, Y]]]] \quad [Y, [Y, [X, [X, Y]]]] \\ & [Y, [Y, [Y, [X, Y]]]] \quad [[X, Y], [X, [X, Y]]] \quad [[X, Y], [Y, [X, Y]]] \end{aligned}$$

Note that B_{13} and B_{14} have the form $[B_3, B_4]$ and $[B_3, B_5]$, requiring careful checking of the condition (PH3).

3.2 Maximum growth canonical systems

Using a P. Hall basis, it is possible to construct vector fields which have maximum growth; at each level of bracketing the dimension of the filtration grows by the maximum possible amount. More specifically, we wish construct a set of vector fields $\{X_i\}$ such that when the vector fields are substituted into the expressions for the P. Hall basis elements, the resulting set of vector fields is linearly independent. The method of construction used here is due to Grayson and Grossmann [GG87]. We present only the 2-input case since this is of the most interest to us.

An important property of a P. Hall Basis is that each basis element has a unique representation as a set of nested Lie products

$$B_i = [B_{i_1}, [B_{i_2}, \dots [B_{i_l}, X_j] \dots]] \quad (8)$$

Given a P. Hall basis element $B = [B_i, B_j]$, we convert it into this form by recursively expanding B_j . We associate with each such basis element a vector $\alpha_i \in \mathbf{Z}^n$ which indicates the number of times each basis element occurs in the expansion (8). Thus $\alpha_i(k)$ is the number of times B_k appears in the expansion for B_i . From the properties of a P. Hall basis, it is clear that $\alpha_i(k) = 0$ if $k \geq i$.

Given a P. Hall basis $H = \{B_1, \dots, B_n\}$ we construct a vector field on \mathbf{R}^n using coordinates $h \in \mathbf{R}^n$. Assume $B_i = X_i$ for $i = 1, \dots, m$. Given α_i associated with B_i , $i > m$, we define

$$\begin{aligned} h^{\alpha_i} &= \prod_j h_j^{\alpha_i(j)} \\ \alpha_i! &= \prod_j \alpha_i(j)! \end{aligned}$$

Theorem 2 (Grayson and Grossman) Fix $k \geq 1$ and let n be the rank of the free, nilpotent Lie algebra of order k with 2 generators. Then

$$X_1 = \frac{\partial}{\partial h_1} \quad X_2 = \frac{\partial}{\partial h_2} + \sum_{i=3}^n \frac{h^{\alpha_i}}{\alpha_i!} \frac{\partial}{\partial h_i}$$

generate a free, nilpotent Lie algebra (of vector fields) of order k at the origin.

The vector fields generated by this theorem are extensions of the canonical forms we have seen for degree of nonholonomy 1 and 2. The degree of nonholonomy for these vector fields is identical to the order of nilpotency. One way to interpret and gain insight into this formula is to note that a Lie product

$$[B_{i_1}, [B_{i_2}, \dots [B_{i_k}, X_2]]]$$

corresponds to a vector field obtained by taking the derivative of the components of X_2 with respect to $h_{i_1}, h_{i_2}, \dots, h_{i_k}$. The coefficients of X_2 are chosen such that taking this derivative leaves 1 in the $\frac{\partial}{\partial h_i}$ term.

Example

Consider the two input example given previously, but with order of nilpotency 4 instead of 5. The system generated by Theorem 2 is

$$\begin{aligned}
 \dot{h}_1 &= u_1 & X \\
 \dot{h}_2 &= u_2 & Y \\
 \dot{h}_3 &= h_1 u_2 & [X, Y] \\
 \dot{h}_4 &= \frac{1}{2} h_1^2 u_2 & [X, [X, Y]] \\
 \dot{h}_5 &= h_1 h_2 u_2 & [Y, [X, Y]] \\
 \dot{h}_6 &= \frac{1}{6} h_1^3 u_2 & [X, [X, [X, Y]]] \\
 \dot{h}_7 &= \frac{1}{2} h_1^2 h_2 u_2 & [Y, [X, [X, Y]]] \\
 \dot{h}_8 &= \frac{1}{2} h_1 h_2^2 u_2 & [Y, [Y, [X, Y]]]
 \end{aligned}$$

We can now ask ourselves if it is possible to steer these canonical systems using simple sinusoids. Although the form of the system is different from that we used in Section 2.2, the same approach can be used to steer h_1 through h_5 . That is, sinusoids at the same frequency and proper phase give motion in h_3 and sinusoids at frequency 1 and 2 give motion in h_4 and h_5 (switching the input frequency switches between h_4 and h_5). This can be verified by direct calculation.

Steering in the $h_6 - h_8$ directions is more difficult. Consider the effect of using two simple sinusoids as inputs, $u_1 = a \cos \omega_1 t$ and $u_2 = b \sin \omega_2 t$. In order to prevent motion in lower level brackets, we must have $\omega_1 \neq \pm \omega_2$, $\omega_1 \neq \pm 2\omega_2$, $\omega_2 \neq \pm 2\omega_1$. Assuming these relationships hold, we get the following frequency components in the derivatives of the dynamic system:

$$\begin{aligned}
 \dot{h}_6 : \quad &\omega_1 \pm \omega_2 \quad 3\omega_1 \pm \omega_2 \\
 \dot{h}_7 : \quad &\omega_1 \quad 2\omega_1 \quad 2\omega_2 \quad 2\omega_1 \pm 2\omega_2 \\
 \dot{h}_8 : \quad &\omega_2 \quad \omega_1 \pm \omega_2 \quad \omega_1 \pm 3\omega_2
 \end{aligned}$$

By choosing frequencies such that the derivative has a term at frequency 0, we get motion in that coordinate. Thus $\omega_2 = 3\omega_1$ gives motion in h_6 (only) and $\omega_1 = 3\omega_2$ gives motion in h_8 (only).

Based on these calculations, it would appear that choosing $2\omega_1 = 2\omega_2$ would give motion in h_7 . This is in fact the case, but we also

get motion in the h_3 direction. It is not possible to get motion *only* in the h_7 direction using simple sinusoids. A direct calculation verifies that adjusting the phasing of the inputs does not resolve this dilemma. It may still be possible to steer the system using combinations of sinusoids at different frequencies for each input or using more complicated periodic functions (such as elliptic functions).

Rather than explore the use of more complicated inputs for steering nonholonomic systems, we consider instead a simpler class of systems. The justification for changing the class of systems is simple—most of the systems encountered as examples do not have the complicated structure of our canonical example. Thus there may be a simpler class of systems which is both steerable using simple sinusoids and representative of systems in which we are interested.

3.3 Chained systems

Consider a two input system of the following form:

$$\begin{array}{rcl} \dot{x}_0 & = & u_x \\ \dot{x}_1 & = & y_0 u_x \\ \dot{x}_2 & = & x_1 u_x \\ \dot{x}_3 & = & x_2 u_x \\ & \vdots & \\ \dot{x}_{n_x} & = & x_{n_x-1} u_x \end{array} \qquad \begin{array}{rcl} \dot{y}_0 & = & u_y \\ (y_1 & := & -x_1) \\ \dot{y}_2 & = & y_1 u_y \\ \dot{y}_3 & = & y_2 u_y \\ & \vdots & \\ \dot{y}_{n_y} & = & y_{n_y-1} u_y \end{array} \quad (9)$$

or more compactly

$$\begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} = X u_x + Y u_y \qquad \begin{array}{l} X = \frac{\partial}{\partial x_0} + y_0 \frac{\partial}{\partial x_1} + \sum_{i=2}^n x_{i-1} \frac{\partial}{\partial x_i} \\ Y = \frac{\partial}{\partial y_0} + \sum_{j=2}^n y_{j-1} \frac{\partial}{\partial y_j} \end{array}$$

where $y_1 := -x_1$ to account for skew-symmetry of the Lie bracket. Denote iterated Lie products as $\text{ad}_X^k Y$:

$$\text{ad}_X Y = [X, Y] \qquad \text{ad}_X^k Y = [X, \text{ad}_X^{k-1} Y] = [X, [X, \dots, [X, Y] \dots]]$$

Lemma 3 For the vector fields in equation (9)

$$\begin{aligned}\text{ad}_X^k Y &= (-1)^k \frac{\partial}{\partial x_k} \\ \text{ad}_Y^k X &= (-1)^k \frac{\partial}{\partial y_k}\end{aligned}\quad k > 1$$

Proof. By induction. Since the first level of brackets is irregular, we begin by expanding $[X, Y]$ and $[X, [X, Y]]$.

$$\begin{aligned}[X, Y] &= \left(\frac{\partial}{\partial x_0} + y_0 \frac{\partial}{\partial x_1} + \sum x_{i-1} \frac{\partial}{\partial x_i} \right) \left(\frac{\partial}{\partial y_0} + \sum y_{j-1} \frac{\partial}{\partial y_j} \right) - \\ &\quad \left(\frac{\partial}{\partial y_0} + \sum y_{j-1} \frac{\partial}{\partial y_j} \right) \left(\frac{\partial}{\partial x_0} + y_0 \frac{\partial}{\partial x_1} + \sum x_{i-1} \frac{\partial}{\partial x_i} \right) \\ &= 0 - \frac{\partial}{\partial x_1} \\ [X, [X, Y]] &= X \left(-\frac{\partial}{\partial x_1} \right) + \frac{\partial}{\partial x_1} (X) = 0 + \frac{\partial}{\partial x_2}\end{aligned}$$

Now assume that $\text{ad}_X^k Y = (-1)^k \frac{\partial}{\partial x_k}$. Then

$$\text{ad}_X^{k+1} Y = [X, \text{ad}_X^k Y] = (-1)^k \left(X \left(\frac{\partial}{\partial x_k} \right) - \frac{\partial}{\partial x_k} (X) \right) = (-1)^{k+1} \frac{\partial}{\partial x_{k+1}}$$

The proof for $\text{ad}_Y^k X$ is identical using the facts $[Y, X] = -[X, Y]$ and $y_1 := -x_1$. \square

Proposition 4 System (9) is maximally nonholonomic (controllable).

Proof. There are $2n - 1$ coordinates in (9) and the $2n - 1$ Lie products

$$\{X, Y, \text{ad}_X^i Y, \text{ad}_Y^j X\} \quad i \geq 1, \quad j \geq 2$$

are independent using Lemma 3. We require $j \geq 2$ since $\text{ad}_Y X = -\text{ad}_X Y$ and hence those Lie products can never be independent. \square

To steer this system, we use sinusoids at integrally related frequencies. Roughly speaking, if we use $u_x = \sin t$ and $u_y = \cos kt$ then \dot{x}_1 will have components at frequency $k - 1$, \dot{x}_2 at frequency $k - 2$, etc. \dot{x}_k will have a component at frequency zero and when integrated we get motion in x_k while all previous variables return to their starting values. In the y variables, all frequency components will be of the form $m \cdot k \pm 1$ and hence we get no motion for $k > 1$. (For $k = 1$, y_1 and x_1 are the same variable). We make this precise with the following algorithm.

Algorithm 3

1. Steer x_0 and y_0 to their desired values.
2. For each x_k , $k \geq 1$, steer x_k to its final value using $u_x = a \sin t$, $u_y = b \cos kt$, where a and b satisfy

$$x_k(2\pi) - x_k(0) = \frac{(a/2)^k b}{k!} \cdot 2\pi$$

3. For each y_k , $k \geq 2$, steer y_k to its final value using $u_x = b \cos kt$, $u_y = a \sin t$, where a and b satisfy

$$y_k(2\pi) - y_k(0) = \frac{(a/2)^k b}{k!} \cdot 2\pi$$

Proposition 5 *Algorithm 3 can steer (9) to an arbitrary configuration.*

Proof. The proof is constructive. It suffices to consider only step 2 since step 3 can be proved by switching x and y in what follows. We must show 2 things:

1. moving x_k does not affect x_j , $j < k$
2. moving x_k does not affect y_j , $j = 1, \dots, n_y$

To verify that using $u_x = a \sin t$, $u_y = b \cos kt$ produces motion only in x_k , we integrate the x states. If x_{k-1} has terms at frequency ω_i , then x_k has corresponding terms at $\omega_i \pm 1$ (by expanding products of sinusoids as sums of sinusoids). Since the only way to have $x_i(2\pi) \neq x_i(0)$ is to have x_i have a component at frequency zero, it suffices to keep track only of the lowest frequency component in each variable; higher components will integrate to zero. Direct

computation starting from the origin yields

$$\begin{aligned}
 x_0 &= a(1 - \cos t) \\
 x_1 &= \int \frac{1}{2} \frac{ab}{k} \sin kt \sin t = \frac{ab}{k(k-1)} \sin(k-1)t + \frac{1}{2} \frac{ab \sin(k+1)t}{k(k+1)} \\
 x_2 &= \frac{1}{2^k} \frac{a^2 b}{k(k-1)(k-2)} \sin(k-2)t + \dots \\
 &\vdots \\
 x_k &= \int \left(\frac{a^k b}{2^{k-1} k!} \sin^2 t + \dots \right) dt = \frac{a^k b}{2^{k-1} k!} \frac{t}{2} + \dots
 \end{aligned}$$

$x_k(2\pi) = x_k(0) + \frac{(a/2)^k b}{k!} \pi$ and all earlier x_i 's are periodic and hence $x_i(2\pi) = x_i(0)$, $i < k$. If the system does not start at the origin, the initial conditions generate extra terms of the form $x_{i-1}(0)u_2$ in the i^{th} derivative and this integrates to zero, giving no net contribution.

To show that we get no motion in the y variables, we show that all frequency components in the y 's have the form $mk \pm 1$ where m is some integer. This is true for $y_1 := x_1$ from the calculation above. Assume it is true for y_i :

$$\begin{aligned}
 \dot{y}_{i+1} &= y_i u_2 \\
 &= \sum_m \alpha(m) \sin(mk \pm 1)t \cdot \cos kt \\
 &= \sum_m \frac{\alpha(m)}{2} (\sin((m+1)k \pm 1)t + \sin((m-1)k \pm 1)t)
 \end{aligned}$$

Hence y_{i+1} has components at frequency $m'k \pm 1$ and therefore $y_i(2\pi) = y_i(0)$. \square

To include systems with more than two inputs, we replicate the structure of (9) for each additional input. Let h_{ij}^k represent the motion corresponding to the Lie product $\text{ad}_{X_i}^k X_j$. In the two input case, $x_k = h_{21}^k$ and $y_k = h_{12}^k$. Consider the following system on \mathbb{R}^n :

$$\begin{aligned}
 \dot{h}_j^0 &= u_j & j = 1, \dots, m \\
 \dot{h}_{ij}^1 &= h_{ij}^0 u_j & i > j \text{ and } h_{ji}^1 := -h_{ij}^1 \\
 \dot{h}_{ij}^k &= h_{ij}^{k-1} u_j
 \end{aligned} \tag{10}$$

Proposition 6 *The system (10) is maximally nonholonomic and can be steered using sinusoids.*

Proof. The system (10) can be rewritten

$$\dot{h} = X_1 u_1 + \cdots + X_m u_m$$

with

$$X_j = \frac{\partial}{\partial h_j^0} + \sum_{\substack{i=1 \\ i < j}}^m h_i^0 \frac{\partial}{\partial h_{ij}^1} + \sum_k \sum_i h_{ij}^{k-1} \frac{\partial}{\partial h_{ij}^k}$$

Given any two X_i, X_j , their Lie product expansions only involve terms of the form h_{ij}^k for some k . But this is precisely the vector fields from Lemma 3 and hence

$$\text{ad}_{X_i}^k X_j = (-1)^k \frac{\partial}{\partial h_{ij}^k}$$

Taking these terms for all possible i, j, k we get a set of independent Lie products just as in the proof of Theorem 4.

To show that the system can be steered using sinusoids, pick any $i, j \in \{1, \dots, m\}$, $i < j$. Fix $u_l = 0$ for all $l \neq i, j$. The resulting system is identical to (9) can be steered using algorithm 3. By choosing all possible combinations of i and j , we can move to any position. \square

We define systems having the form of equation (10) as *chained systems*. For a fixed i, j , we refer to the coordinates h_{ij}^k as a *chain*. Each chain in the system can be steered using sinusoidal inputs in the appropriate input channels.

3.4 Non-canonical chained systems

We would like to extend the class of systems which we can steer by including systems which have similar structure to equation (9), but with additional nonlinearities. The following example illustrates the limitations of using sinusoidal inputs for this purpose. Consider the system

$$\begin{aligned}\dot{x}_0 &= u_1 \\ \dot{y}_0 &= u_2 \\ \dot{x}_1 &= (y_0 + \epsilon y_0^2)u_1 \\ \dot{x}_2 &= (x_1 + \epsilon x_1^2)u_1 \\ \dot{x}_3 &= x_2 u_1\end{aligned}$$

For $\epsilon = 0$ this is a chained system with a single chain: $\text{ad}_{g_1}^k g_2$, $k = 0, 1, 2, 3$ together with g_2 forms a basis for \mathbb{R}^5 . Controllability around the origin is guaranteed even if $\epsilon \neq 0$. We would like to show that simple sinusoidal inputs can be used to steer this system.

If we apply inputs $u_1 = \sin t$ and $u_2 = \cos 3t$, we get the following motion, starting from $x = 0$

$$\begin{aligned}x_0(2\pi) &= 0 \\y_0(2\pi) &= 0 \\x_1(2\pi) &= 0 \\x_2(2\pi) &= -\frac{7}{1440}\epsilon^2 \\x_3(2\pi) &= \frac{\pi}{24} + 2.5 \times 10^5 \epsilon^2\end{aligned}$$

The reason for this perturbation in x_2 is that the (small) nonlinear terms cause zero frequency components to appear in \dot{x}_2 . Hence we cannot use simple sinusoids to steer this system as before.

Nonetheless, there are many special instances where sinusoids are an important tool. For example, we were able to steer the automobile with sinusoids, despite the nonlinearities. Since the automobile had degree of nonholonomy 2, the problems present in the previous example do not occur. Another example is a system which has the chained form until the last coordinate. In this case, harmonic analysis is needed when finding the motion at the last step of the algorithm and zero frequency terms do not appear in any previous coordinates.

It may also be possible to use feedback transformation to convert certain systems into chained form. This is similar to the technique used in nonlinear control to convert a nonlinear system into a linear one by using a change of coordinates and state feedback. Similar efforts have been used by Lafferriere and Sussmann [LS90] to convert systems into nilpotent form for use with their planning algorithm. It is interesting to note that in several of their examples, the converted systems were also in chained form.

4 Converting systems to chained form

In this section we consider the problem of converting a nonholonomic system into chained form. Given a nonholonomic system, we ask if there is a *feedback transformation*

$$\begin{aligned}\xi &= \phi(x) \\ u &= \alpha(x) + \beta(x)v\end{aligned}$$

such that the resulting system

$$\dot{\xi} = \tilde{g}(\xi)v$$

is in chained form. If such a transformation exists, the system can be steered using sinusoids in v and the resulting path can be realized using a precompensator $u = \alpha(x) + \beta(x)v$. This ideal has also been utilized by Lafferriere and Sussmann in the context of steering nilpotent systems.

Consider first the case of a system with 2 inputs in which controllability is achieved using the distribution

$$\{g_1, g_2, \text{ad}_{g_1}g_2, \dots, \text{ad}_{g_1}^{p-1}g_2\}$$

Suppose further that there exists a function h which satisfies

$$\begin{aligned}dh \cdot \{g_2, \text{ad}_{g_1}g_2, \dots, \text{ad}_{g_1}^{p-1}g_2\} &= 0 \\ dh \cdot \text{ad}_{g_1}^p g_2 &\neq 0\end{aligned}\tag{11}$$

and that g_1 has the special form

$$g_1 = \frac{\partial}{\partial x_1} + \sum g_1^i \frac{\partial}{\partial x_i}$$

In this case we can construct a new set of coordinates using h as an “output function.” This procedure is analogous to that used in input/output linearization of nonlinear systems [Isi89]. We illustrate this procedure by example; details are the subject of a forthcoming paper [MS91].

Examples

As an example of a nonholonomic control system, we again consider the kinematic model of an automobile. The kinematics are given by equation(7), reproduced here:

$$\begin{aligned}\dot{x} &= \cos \theta u_1 \\ \dot{y} &= \sin \theta u_1 \\ \dot{\phi} &= u_2 \\ \dot{\theta} &= \frac{1}{l} \tan \phi u_1\end{aligned}\tag{12}$$

To check controllability, we construct the involutive closure of $\{g_1, g_2\}$:

$$\begin{aligned}g_1 &= \cos \theta \frac{\partial}{\partial x} + \sin \theta \frac{\partial}{\partial y} + \frac{1}{l} \tan \phi \frac{\partial}{\partial \theta} \\ g_2 &= \frac{\partial}{\partial \phi} \\ g_3 = [g_1, g_2] &= \frac{-1}{l \cos^2 \phi} \frac{\partial}{\partial \theta} \\ g_4 = [g_1, g_3] &= \frac{-\sin \theta}{l \cos^2 \phi} \frac{\partial}{\partial x} + \frac{\cos \theta}{l \cos^2 \phi} \frac{\partial}{\partial y} \\ g_5 = [g_2, g_3] &= \frac{-2 \tan \phi}{l \cos^2 \phi} \frac{\partial}{\partial \theta}\end{aligned}$$

$\{g_1, g_2, g_3, g_4\}$ are linearly independent when $\phi \neq \pm \pi$ and hence the system is completely controllable away from those points.

We now convert the system to chained form. First scale the inputs so that u_1 enters \dot{x} directly. Reusing the symbol u_1 , the kinematics become:

$$\begin{aligned}\dot{x} &= u_1 \\ \dot{y} &= \tan \theta u_1 \\ \dot{\phi} &= u_2 \\ \dot{\theta} &= \frac{1}{l} \sec \theta \tan \phi u_1\end{aligned}$$

Choose the y position of the car as the function h ; it is easy to verify that this function satisfies the conditions in equation (11). The resulting change of coordinates is

$$\begin{aligned}\xi_1 &= x & u_1 &= v_1 \\ \xi_2 &= L_{g_1}^2 h = \frac{1}{l} \sec^3 \theta \tan \phi & u_2 &= -\frac{3}{l} \sin^2 \phi \tan \theta \sec \theta v_1 \\ \xi_3 &= L_{g_1} h = \tan \theta & &+ l \cos^2 \phi \cos^3 \theta v_2 \\ \xi_4 &= h = y & &\end{aligned}$$

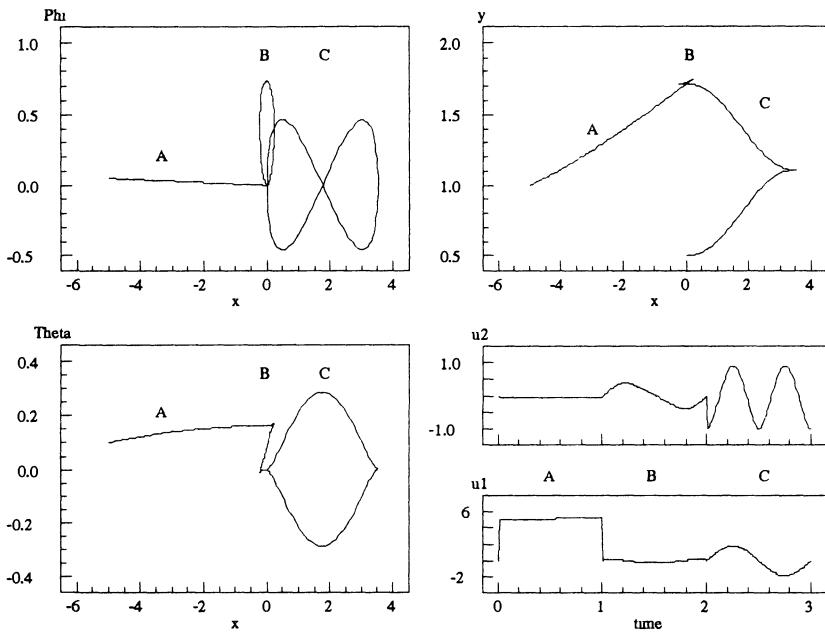


Figure 5: Sample trajectories for a car using chained form. These trajectories are qualitatively similar to those in Figure 4, but do not require the calculation of Fourier coefficients for determining open loop trajectories.

And the transformed system has the form:

$$\begin{aligned}\dot{\xi}_1 &= v_1 \\ \dot{\xi}_2 &= v_2 \\ \dot{\xi}_3 &= \xi_2 v_1 \\ \dot{\xi}_4 &= \xi_3 v_1\end{aligned}$$

This system can now be steered using the sinusoidal algorithm of the previous section or another method, such as Lafferriere and Sussmann's algorithm for generating motions for nilpotent systems. The motion is implemented as a feedback pre-compensator which converts the v inputs into the actual system inputs, u . This feedback transformation agrees with that used in Lafferriere and Sussmann to nilpotentize the kinematic car example. Their formulation of the feedback transformation was not presented, although it seems clear that a similar approach must have been used.

A second example is the case of a car pulling a trailer. The equations of motion are identical to those of the car, with an additional equation specifying the motion of the attached trailer:

$$\dot{\psi} = \sin(\theta - \psi)u_1$$

(see [MS90b, LS90]). By solving the partial differential equations in the statement of the proposition above, it can be shown that the function

$$h(y, \psi) = y - \log\left(\frac{1 + \sin \psi}{\cos \psi}\right)$$

generates a chained set of coordinates. Again we can locally steer the trailer using sinusoidal inputs or other methods.

5 Discussion

There are many open questions in the use of sinusoids for generating trajectories for nonholonomic questions. The full class of systems to which the techniques presented here can be applied is not known. In particular, the use of feedback transformations and the necessary conditions for converting a system into chained form is of great interest. This appears to be related to the problem of nilpotentization of a set of vector fields, which has been by Hermes, Lundell, and Sullivan [HLS84].

Another fundamental problem of practical importance is obstacle avoidance. In many situations, such as parallel parking a car, obstacles play an important role in choosing a good trajectory. The use of sinusoidal inputs allows some freedom in shaping a trajectory based on the presence of obstacles. By varying the amplitude, phase, and number of cycles of the input sinusoids, different trajectories can be generated which result in the same net motion. Figure 6 shows three possible trajectories for parallel parking a car.

Finally, we note that all of the trajectories we have studied have been fundamentally open loop. Errors in the initial conditions, model mismatch, and sensor noise will all degrade performance. A fundamental property of feedback control is robustness

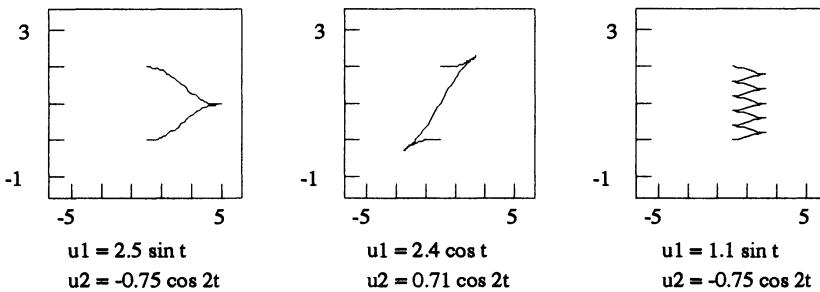


Figure 6: Alternative trajectories for parallel parking

with respect to these disturbances. We are currently investigating methods for generating controllers for nonholonomic systems which respect the fundamental limitations mentioned in the introduction. An effective strategy may be to design controllers which perform some type of trajectory tracking rather than stabilization to a point. Related work in this area includes that of Bloch and McClamroch [BM89, BM90] and Samson [Sam90].

References

- [Arn89] V. I. Arnold. *Mathematical Methods of Classical Mechanics*. Springer-Verlag, second edition, 1989.
- [BM89] A. M. Bloch and N. H. McClamroch. Control of mechanical systems with classical nonholonomic constraints. In *IEEE Control and Decision Conference*, pages 201–205, 1989.
- [BM90] A. M. Bloch and N. H. McClamroch. Controllability and stabilizability properties of a nonholonomic control system. In *IEEE Control and Decision Conference*, 1990.
- [Bro81] R. W. Brockett. Control theory and singular Riemannian geometry. In *New Directions in Applied Mathematics*, pages 11–27. Springer-Verlag, New York, 1981.
- [Bro89] R. W. Brockett. On the rectification of vibratory motion. *Sensors and Actuators*, 20(1–2):91–96, 1989.

- [GG87] M. Grayson and R. Grossman. Models for free nilpotent Lie algebras. Technical Memo PAM-397, Center for Pure and Applied Mathematics, University of California, Berkeley, 1987. (to appear in *J. Algebra*).
- [GV88] V. Gershkovich and A. Vershik. Nonholonomic manifolds and nilpotent analysis. *Journal of Geometry and Physics*, 5(3):407–452, 1988.
- [Hal59] M. Hall. *The Theory of Groups*. Macmillan, 1959.
- [HLS84] H. Hermes, A. Lundell, and D. Sullivan. Nilpotent bases for distributions and control systems. *Journal of Differential Equations*, 55:385–400, 1984.
- [Isi89] A. Isidori. *Nonlinear Control Systems*. Springer-Verlag, 2nd edition, 1989.
- [LC90] Z. Li and J. Canny. Motion of two rigid bodies with rolling constraint. *IEEE Transactions on Robotics and Automation*, 6(1):62–71, 1990.
- [LMR89] Z. Li, R. Montgomery, and M. Raibert. Dynamics and optimal control of a legged robot in flight phase. In *IEEE International Conference on Robotics and Automation*, pages 1816–1821, 1989.
- [LS90] G. Lafferriere and H. J. Sussmann. Motion planning for controllable systems without drift: a preliminary report. Technical Report SYCON-90-04, Rutgers Center for Systems and Control, 1990.
- [LS91] G. Lafferriere and H. J. Sussmann. Motion planning for controllable systems without drift. In *IEEE International Conference on Robotics and Automation*, pages 1148–1153, 1991.
- [MS90a] R. M. Murray and S. S. Sastry. Grasping and manipulation using multifingered robot hands. In R. W. Brockett, editor, *Robotics: Proceedings of Symposia in Applied Mathematics, Volume 41*, pages 91–128. American Mathematical Society, 1990.

- [MS90b] R. M. Murray and S. S. Sastry. Steering nonholonomic systems using sinusoids. In *IEEE Control and Decision Conference*, 1990.
- [MS91] R. M. Murray and S. S. Sastry. Nonholonomic motion planning: Steering using sinusoids. Technical Report UCB/ERL M91/45, Electronics Research Laboratory, University of California at Berkeley, 1991.
- [Mur90] R. M. Murray. *Robotic Control and Nonholonomic Motion Planning*. PhD thesis, University of California at Berkeley, 1990.
- [Sam90] C. Samson. Velocity and torque feedback control of a nonholonomic cart. In *International Workshop in Adaptive and Nonlinear Control: Issues in Robotics*, 1990.
- [Ser65] J-P. Serre. *Lie Algebras and Lie groups*. W. A. Benjamin, New York, 1965.
- [Var84] V. S. Varadarajan. *Lie Groups, Lie Algebras, and Their Representations*. Springer-Verlag, 1984.
- [VG88] A. M. Vershik and V. Ya. Gershkovich. Nonholonomic problems and the theory of distributions. *Acta Applicandae Mathematicae*, 12:181–209, 1988.

3

Smooth Time-Periodic Feedback Solutions for Nonholonomic Motion Planning

L. Gurvits and Z.X. Li*

Robotics Research Laboratory
Courant Institute of Mathematical Sciences
New York University, NY., NY 10012

Abstract

In this paper, we present an algorithm for computing time-periodic feedback solutions for nonholonomic motion planning with collision-avoidance. For a first-order Lie bracket system, we begin by computing a holonomic collision-free path using the potential field method. Then, we compute a nonholonomic path approximating the collision-free path within a predetermined bound. For this we first solve for extended inputs of an extended system using Lie bracket completion vectors. We then use averaging techniques to calculate the asymptotic trajectory of the nonholonomic system under application of a family of highly-oscillatory inputs. Comparing the limiting trajectories with the extended system we obtain a system of nonlinear equations from which the desired admissible control inputs can be solved. For higher-order Lie bracket systems we use multi-scale averaging and apply recursively the algorithm for first-order Lie bracket systems. Based on averaging techniques we also provide error bounds between a nonholonomic system and its averaged system.

1 Introduction

Nonholonomic Motion Planning (NMP) is the study of path planning for a robotic system subject to either nonholonomic constraints, e.g., rolling constraint, or nonintegrable conservation laws, e.g., conservation of angular momentum. NMP arises in dexterous manipulation with a multifingered robotic hand ([FGL91b], [Li89] and [MS90]), path planning for mobile robots([Lau92] and [JC89]), control of nonholonomic actuators ([Bro88]), and reconfiguration of a space structure using internal motion ([FGL91b], [Kri90], [NM90] and [PD90]).

The problem of nonholonomic motion planning can be stated as follows:

Problem 1.1 (Nonholonomic Motion Planning Problem:)
Let Q be a n -dimensional configuration space of a robotic system, x_0 and x_f two given initial and final configurations, $B(x) \in \mathbb{R}^{n \times m}$ a m -dimensional completely nonholonomic distribution¹ (or constraints) which translates into a system of differential equations of the form

$$\dot{x} = b_1(x)u_1 + \cdots + b_m(x)u_m = B(x)u \quad (1)$$

and

$$C_i(x) \leq 0, i = 1, \dots l \quad (2)$$

a set of position constraints for collision avoidance. Compute a set of input $u(t) \in \mathbb{R}^m$, $t \in [0, T]$, perhaps of optimal cost, such that the resulting trajectory $x(t) \in Q, t \in [0, T]$, is collision-free and links x_f to x_0 .

Research in nonholonomic motion planning has expanded recently. For example, Brockett ([Bro81] and [BD91]) and Murray and Sastry ([MS90] and [MS92]) studied analytic solutions of a family of canonical or chained systems using Fourier analysis and optimal control. They applied feedback transformation and linearization techniques to reduce a general system to the canonical

¹This is equivalent to the nonlinear control system (1) being controllable.

or chained form. Lafferriere and Sussmann ([LS90]) studied analytic solutions of a class of nilpotent or nilpotenizable systems. Their iterative algorithm can also produce approximate solutions for non-nilpotent or non-nilpotenizable systems. In ([Mon89], [Mon92], [BMR90] and [RMB92]), optimal solutions and feedback control for coupled rigid-body systems or nonholonomic systems with drift were studied. Based on Ritz approximation theory, Fernandes, Gurvits and Li ([FGL91b] and [FGL91a]) gave a numerical algorithm for NMP along with some interesting simulation results. In [SL91], Sussmann and Liu presented a new approach to nonholonomic motion planning with collision-avoidance.

It is clear that the approaches taken by ([Bro81], [BD91], [MS90], [MS92], [LS90] and [FGL91b]) can be classified as open-loop. While there are certain advantages associated with each of these open-loop approaches, an open-loop approach is less likely to produce solutions which are robust with respect to modeling uncertainties and sensoring errors, as compared with a feedback approach. However, as Brockett pointed out ([Bro83]) that smooth, time-independent feedbacks do not exist in general for a nonholonomic motion planning system.

In view of Brockett's result and the fact that a typical robotic task demands real-time generation of nonholonomic paths, we will present in this paper a new algorithm for computing collision-free, smooth time-periodic feedback solutions for NMP systems. The algorithm is efficient and requires only ordinary computational tools for its implementation. In addition, through the use of averaging techniques we will offer a geometrical interpretation to the approach used in ([SL91]). Here, the different time-scales of averaging arise because motion in the directions of the control vector fields is much faster than that in the directions of the Lie-bracket vector fields. Furthermore, if a system under consideration is of higher Lie-bracket order, then multi-scale averaging results and the feedback algorithm becomes recursive.

The paper is organized as follows: In Section 2.1, we present the algorithm for a first-order Lie bracket system along with a simple example. In Section 2.2, we generalize the algorithm to

higher-order Lie bracket systems. In Section 2.3, we apply averaging techniques to prove two important theorems used in the feedback algorithm and give estimates for error bounds. In Section 3, we provide motivations for the control inputs used in the feedback algorithm by studying the control of a class of nonholonomic systems. In Section 3.2, we present some controllability results of a linearized nonholonomic system around a nominal trajectory. In Appendix A, we compare some of our results with that of Coron's and establish a negative result on the existence of smooth time-periodic feedbacks for exponential stability. In Appendix B, we study analytic solutions for generalized Brockett system.

2 Averaging and the Feedback Algorithm

In this section, we describe an algorithm that computes time-periodic feedback solutions for NMP. In contrast to the Basis Algorithm ([FGL91b]), this algorithm will give solutions which are

- collision-free and
- robust with respect to modeling uncertainties and sensoring errors.

Furthermore, only ordinary computational tools will be needed for its implementation.

We will first consider the case of first-order Lie bracket systems and then generalize the results to higher-order Lie bracket systems.

2.1 The First-Order Lie Bracket Case

Consider the following controllable nonholonomic system

$$\dot{x} = b_1(x)u_1 + b_2(x)u_2 + \cdots + b_m(x)u_m = B(x)u, \quad x \in \mathfrak{X}^n \quad (3)$$

with initial and final configurations $x_0, x_f \in \Re^n$, ($n > m$). The system is said to be a first-order Lie bracket system if only first-order Lie bracket operations are necessary to generate the controllability Lie algebra. For example, let $l \triangleq m(m - 1)/2$ then the rank of the following Lie algebra completion matrix

$$\{b_1(x), b_2(x), \dots, b_m(x), c_1(x), \dots, c_l(x)\} \triangleq C(x)$$

is n , $\forall x \in \Re^n$, where

$$c_i(x) = [b_j(x), b_k(x)] = \frac{\partial b_k(x)}{\partial x} b_j(x) - \frac{\partial b_j(x)}{\partial x} b_k(x)$$

denotes the Lie bracket vector field of b_j and b_k .

The algorithm that computes time-periodic feedback solutions for system (3) has four steps and is described below. We leave the proof of correctness to Section 2.3.

Algorithm 2.1 (Feedback Algorithm for 1st-Order Systems)

Step 1 Computing Collision-free Holonomic Path: Use holonomic motion planning techniques (see, e.g., [SS86], [Can88], [Yap87], [Kod87] and [Rim90]) to construct a **collision-free** (holonomic) dynamical system

$$\dot{y} = g(y), \quad y(0) = x_0 \in \Re^n \quad (4)$$

such that the goal configuration, x_f , subject to some topological considerations, is essentially the global attractor. For example, if we follow the potential field method as in ([Kod87] and [Rim90]) then $g(y)$ is the gradient vector field of some navigation function.

Other than practical considerations dynamical system (4) can be chosen arbitrary. Our objective is to find for system (3) time-periodic feedbacks of the form

$$u = u(x, t, \epsilon)$$

where ϵ is a small parameter, such that the C -norm of the trajectory error between the holonomic system (4) and the nonholonomic system (3) goes to zero as $\epsilon \rightarrow 0$,

$$\lim_{\epsilon \rightarrow 0} \|x - y\|_{C[0,T]} \rightarrow 0$$

Here, T is fixed and the C -norm of a continuous function $y(t) \in \mathbb{R}^n, t \in [0, T]$, is defined by

$$\|y\|_{C[0,T]} = \max_{t \in [0,T]} |y(t)|$$

Step 2 Compute extended inputs $v = (v_1, v_2, \dots, v_{m+l})^T$ from the following system of linear equations

$$(b_1, \dots, b_m, c_1, \dots, c_l) v = C(y)v = g(y)$$

Since the Lie bracket completion matrix $C(y)$ has rank n , the solution is given by

$$v = C^T(CC^T)^{-1}g(y) \triangleq C^\#g(y)$$

Thus, the collision-free holonomic path satisfies the following system of differential equations

$$\dot{y} = b_1v_1 + b_2v_2 + \dots + b_mv_m + c_1v_{m+1} + \dots + c_lv_{m+l} \quad (5)$$

with initial condition $y(0) = x_0$.

Step 3 Let the control input of the nonholonomic system (3) be of the form

$$u_i^\epsilon = \alpha_i(x) + \sqrt{\frac{2}{\epsilon}} \sum_{j=1}^l \beta_i^j(x) \sin(jt/\epsilon) + \gamma_i^j(x) \cos(jt/\epsilon) \quad (6)$$

where ϵ is a small parameter.

Clearly, u_i^ϵ is time-periodic feedback. It becomes highly oscillatory as $\epsilon \rightarrow 0$. The asymptotic behavior of system (3) under application of control input (6) can be computed using

averaging techniques ([Arn78]). Following the results of Section 2.3 (see also [Gur91]) the asymptotic trajectory satisfies a differential equation of the form

$$\dot{y} = \sum_{i=1}^m \delta_i(y) b_i + \sum_{i < j} \eta_{ij} [b_i, b_j] \quad (7)$$

where

$$\delta_i = \alpha_i + \sum_{j=1}^m \sum_{k=1}^l \left(\langle \nabla \beta_i^k, b_j \rangle \gamma_j^k - \langle \nabla \gamma_i^k, b_j \rangle \beta_j^k \right) \operatorname{sgn}(j-i) \frac{1}{k}$$

$$\operatorname{Skew}(\eta_{ij}) \triangleq \begin{pmatrix} 0 & \eta_{12} & \eta_{13} & \cdot & \cdot & \cdot \\ -\eta_{12} & 0 & \eta_{23} & \cdot & \cdot & \cdot \\ -\eta_{13} & -\eta_{23} & 0 & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \end{pmatrix} = \sum_{k=1}^l (\beta^k \wedge \gamma^k) \frac{1}{k}$$

$$\beta^k = (\beta_1^k, \dots, \beta_m^k)^T, \quad \gamma^k = (\gamma_1^k, \dots, \gamma_m^k)^T$$

and the \wedge (wedge) product of two vectors $x, y \in \Re^n$ gives a $n \times n$ skew-symmetric matrix

$$x \wedge y = xy^T - yx^T.$$

∇f denotes the gradient vector field of the function f .

Step 4 Solving for the Desirable Inputs: Equating the coefficients of system (5) with that of (7) yield a system of nonlinear equations

$$\sum_{k=1}^l (\beta^k \wedge \gamma^k) \frac{1}{k} = \operatorname{Skew}(\eta_{ij}) \quad (8)$$

$$\alpha_i + \sum_{j=1}^m \sum_{k=1}^l \left(\langle \nabla \beta_i^k, b_j \rangle \gamma_j^k - \langle \nabla \gamma_i^k, b_j \rangle \beta_j^k \right) \operatorname{sgn}(j-i) \frac{1}{k} = v_i, \quad (9)$$

where $i = 1, \dots, m$. Solving for (α, β, γ) from Eqs. (8) and (9) gives the desirable inputs to the nonholonomic system.

Remark 2.1 (1) Estimates of the trajectory errors between the holonomic system and the nonholonomic system can be found in Section 2.3. (2) The algorithm also applies when both systems are non-autonomous, i.e.,

$$\begin{aligned}\dot{x} &= B(x, t)u, \\ \dot{y} &= g(y, t)\end{aligned}$$

In this case, the Lie bracket and gradient operations are performed for fixed time t , and the control input has the form

$$u_i = \alpha_i(x, t) + \sqrt{\frac{2}{\epsilon}} \sum_{j=1}^l \beta_i^j(x, t) \sin\left(\frac{jt}{\epsilon}\right) + \gamma_i^j(x, t) \cos\left(\frac{jt}{\epsilon}\right).$$

□

Example 2.1 We use the *rolling disk* example to illustrate the *feedback algorithm*. Let (x, y) be the contact location, ϕ the angle and (u_1, u_2) the steering and rolling velocity, respectively, of a rolling disk. Then, the equations describing the constraint are given by (see [FGL91b])

$$\begin{bmatrix} \dot{\phi} \\ \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} u_1 + \begin{bmatrix} 0 \\ \cos \phi \\ \sin \phi \end{bmatrix} u_2 \triangleq b_1 u_1 + b_2 u_2 \quad (10)$$

It is clear that this is a first-order Lie bracket system as the completion matrix

$$C = (b_1, b_2, [b_1, b_2]) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi & \cos \phi \end{pmatrix}$$

is nonsingular over the configuration space.

Assuming that the holonomic system is given by the following exponentially stable system

$$\dot{z} = -z \quad (11)$$

We want to construct inputs for system (10) whose asymptotic behavior as $\epsilon \rightarrow 0$ is given by that of (11).

From Step 2, the extended inputs are found to be

$$\begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = C^T(CC^T)^{-1} \begin{bmatrix} -\phi \\ -x \\ -y \end{bmatrix} = \begin{bmatrix} -\phi \\ -(x \cos \phi + y \sin \phi) \\ x \sin \phi - y \cos \phi \end{bmatrix}$$

Choose the following forms of the control inputs

$$\begin{aligned} u_1^\epsilon &= \alpha_1 + \sqrt{\frac{2}{\epsilon}} \beta_1 \cos \frac{t}{\epsilon} \\ u_2^\epsilon &= \alpha_2 + \sqrt{\frac{2}{\epsilon}} \beta_2 \sin \frac{t}{\epsilon} \end{aligned}$$

where we have set γ_i 's to zero for simplicity.

Thus, the system of nonlinear equations from Steps 3 and 4 have the form

$$\begin{aligned} \alpha_1 - \beta_2 \langle \nabla \beta_1, b_2 \rangle &= -\phi \\ \alpha_2 + \beta_1 \langle \nabla \beta_2, b_1 \rangle &= -(x \cos \phi + y \sin \phi) \\ \beta_1 \beta_2 &= x \sin \phi - y \cos \phi \end{aligned} \quad (12)$$

It is not difficult to check that

$$\begin{aligned} \beta_1 &= 1, \alpha_1 = -\phi \\ \beta_2 &= x \sin \phi - y \cos \phi \\ \alpha_2 &= -2(x \cos \phi + y \sin \phi) \end{aligned}$$

constitute a set of solution to Eq. (12). Thus, the time-periodic feedback solutions are

$$\begin{aligned} u_1 &= -\phi + \sqrt{\frac{2}{\epsilon}} \cos \left(\frac{t}{\epsilon} \right) \triangleq u_1(\phi, t, \epsilon) \\ u_2 &= -2(x \cos \phi + y \sin \phi) + \sqrt{\frac{2}{\epsilon}} (x \sin \phi - y \cos \phi) \sin \left(\frac{t}{\epsilon} \right) \\ &\triangleq u_2(\phi, x, y, t, \epsilon) \end{aligned}$$

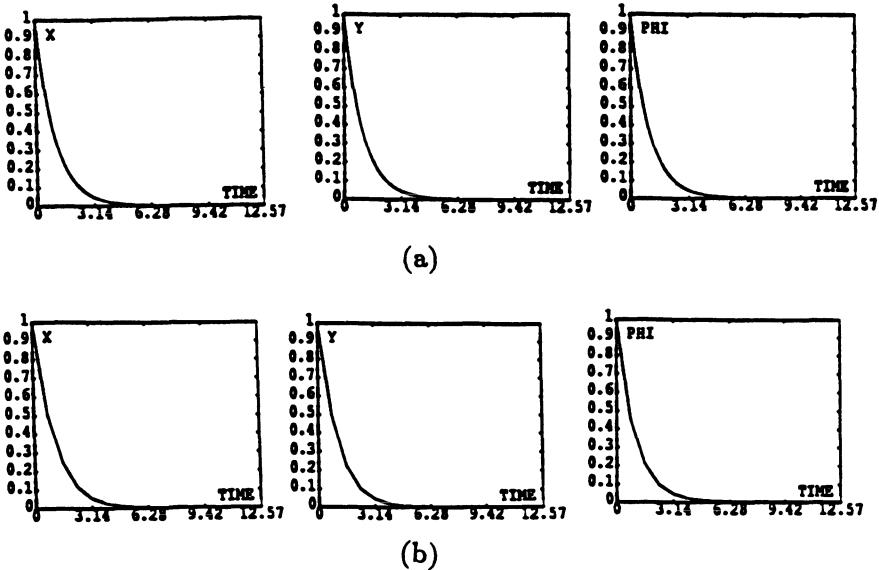


Figure 1: Simulated trajectories for (a) $\epsilon = 0.0001$ and (b) $\epsilon = 0.015625$.

The solution grows out of bound as $\epsilon \rightarrow 0$. To avoid unbounded control, we can use symmetry of a driftless system to rescale the control input

$$\begin{aligned}\tilde{u}_1(\phi, t, \epsilon) &= \sqrt{\epsilon} u_1(\phi, t\sqrt{\epsilon}, \epsilon) \\ \tilde{u}_2 &= \sqrt{\epsilon} u_2(\phi, x, y, t\sqrt{\epsilon}, \epsilon)\end{aligned}$$

Let $\tilde{x}(t) \in \mathbb{R}^3$ be the corresponding nonholonomic trajectory, we have that

$$x(t\sqrt{\epsilon}) = \tilde{x}(t)$$

Figure 1 show simulation results for two different values of ϵ and scaled control inputs \tilde{u}_i . \square

Remark 2.2 (1) The feedback input constructed above does not vanish at the origin, as $u_1(0, t) \neq 0$, while the control input in [Cor91] is identically zero at the origin. (2) It is impossible to find (β_1, β_2) which are differentiable at the origin and satisfy $\beta_1(0) = \beta_2(0) = 0$. If so, from the equation

$$\beta_1 \beta_2 = x \sin \phi - y \cos \phi$$

we have that

$$\frac{\partial}{\partial y}(\beta_1 \beta_2)|_0 = \beta_2 \frac{\partial \beta_1}{\partial y} + \beta_1 \frac{\partial \beta_2}{\partial y} = -\cos 0 = -1$$

resulting in a contradiction. \square

It is interesting to note that solutions of the nonholonomic system converge to a small attractor containing the origin. Denote by B a ball of the origin. Then, solutions of the nonholonomic system have the form

$$x(t) = e^{-t}x(0) + \sqrt{\epsilon}G(x, t), \quad t \in [0, 2\pi]$$

for all $x_0 \in B$, where $G(x, t)$ is bounded. For sufficiently small ϵ , $x(2\pi)$ is also in B . We can use $x(2\pi)$ as the new initial condition and apply the same set of control inputs to the system. Consequently, trajectories of the nonholonomic system converge to a subset of a ball centered at the origin with radius

$$\sqrt{\epsilon} \max_{x \in B, t \in [0, 2\pi]} G(x, t).$$

2.2 Higher-Order Lie Bracket Systems

To compute time-periodic feedback solutions for higher-order Lie bracket systems we introduce multiple small parameters and use a recursive algorithm. Consider a r^{th} -order Lie bracket system,

$$\dot{x} = b_1 u_1 + \cdots + b_m u_m, \quad x \in \Re^n \tag{13}$$

together with the following holonomic system

$$\dot{y} = g(y), \quad y(0) = x(0) \in \Re^n \tag{14}$$

and time interval $[0, 2\pi]$.

Algorithm 2.2 (For Higher-Order Systems)

Step 1 Compute a Lie bracket completion matrix for the system

$$C(x) = \{b_1, \dots, b_m, [b_i, b_j], \dots, \underbrace{[b_{i_1}, [b_{i_2}, [\dots, b_{i_r}, \dots]]]}_r\} \quad (15)$$

Step 2 Compute the extended inputs by the formula

$$(v_1, \dots, v_m, v_{m+1}, \dots, v_{m+\eta})^T = C^\sharp \cdot g(x).$$

where $C^\sharp = C^T(CCT)^{-1}$ is the generalized inverse of C and η is the number of entries of $C(x)$ less m .

Step 3 Let $r > 1$ be the order of $C(x)$, we form another matrix $C_l(x)$ of order no more than $(r - 1)$ as follows. First, rearrange the entries of $C(x)$ so that

$$C(x) = \{\bar{C}_l(x), C_h(x)\} \quad (16)$$

where $\bar{C}_l(x)$ consists of vectors of order strictly less than r and $C_h(x)$ consists of the highest order vectors. Second, let $\tilde{C}_l(x)$ be a minimal set of vector fields such that its order is strictly less than r and any vector in $C_h(x)$ can be obtained by a single Lie bracket operation on the entries of $\bar{C}_l(x)$ and $\tilde{C}_l(x)$. Finally, set

$$C_l(x) = \{\bar{C}_l(x), \tilde{C}_l(x)\} \triangleq \underbrace{\{b_1, \dots, b_m, f_1, \dots, f_{s_1}, g_1, \dots, g_{s_2}\}}_{\bar{C}_l(x)} \cup \underbrace{\{f_1, \dots, f_{s_1}, g_1, \dots, g_{s_2}\}}_{\tilde{C}_l(x)}. \quad (17)$$

It is clear from the above construction that, using Algorithm 2.1, trajectories of the system defined by $C(x)$ can be approximated by that of the system defined by $C_l(x)$.

Step 4 Introduce a small parameter ϵ_1 and use Algorithm 2.1 to compute the coefficients of the following input functions

$$\begin{aligned} u_i^{\epsilon_1} &= \gamma_i(x, t) + \frac{1}{\sqrt{\epsilon}} \sum_{k=1}^{\rho} \alpha_{k,1}^i(x, t) \sin\left(\frac{kt}{\epsilon_1}\right) + \beta_{k,1}^i(x, t) \cos\left(\frac{kt}{\epsilon_1}\right), \\ v_i^{\epsilon_1} &= \mu_i(x, t) + \frac{1}{\sqrt{\epsilon}} \sum_{k=1}^{\rho} \alpha_{k,2}^i(x, t) \sin\left(\frac{kt}{\epsilon_1}\right) + \beta_{k,2}^i(x, t) \cos\left(\frac{kt}{\epsilon_1}\right), \\ w_i^{\epsilon_1} &= \frac{1}{\sqrt{\epsilon}} \sum_{k=1}^{\rho} \alpha_{k,3}^i(x, t) \sin\left(\frac{kt}{\epsilon_1}\right) + \beta_{k,3}^i(x, t) \cos\left(\frac{kt}{\epsilon_1}\right) \end{aligned}$$

where the index i runs from 1 to m for the first equation, 1 to s_1 for the second equation and 1 to s_2 for the last equation, and ρ is defined as follows: First, let X be the minimal set of vector fields such that any element of $C_h(x)$ is the Lie bracket vector field of two vectors in X . Then, denote by p the number of entries of X , and we have

$$\rho = p(p - 1)/2.$$

Step 5 If the Lie bracket order of $C_l(x)$ is no less than 1, then replace $C(x)$ with $C_l(x)$ and repeat Step 3 and 4, using another small parameter and the extended inputs computed previously.

Note that the asymptotic expression for the error function in the k^{th} -step of the above recursive algorithm can be shown to be (see Section 2.3)

$$\begin{aligned} \Delta_0 &= 0, \\ \Delta_{s+1} &= \sqrt{\epsilon_{s+1}} (C_1 K_s + e^{2\pi C_3 K_s} C_2 K_s^2) \end{aligned} \quad (18)$$

$$\begin{aligned} K_0 &= \max_{i,k,x} \{ \|f_i(x)\|, \|G_k(x)\| \}, \\ K_{s+1} &= \frac{1}{\sqrt{\epsilon_{s+1}}} C_4 K_s, \end{aligned} \quad (19)$$

and the accumulated error is

$$\Delta = \|x - y\|_{C[0,2\pi]} = \Delta_1 + \cdots + \Delta_r.$$

In order for the nonholonomic trajectories to converge to that of the nominal system, the ϵ_s 's have to be chosen so that Δ is within an acceptable bound.

Example 2.2 The constraint equations of a ball rolling on a plane have the form

$$\dot{x} = \begin{bmatrix} 0 \\ \sec x_1 \\ -\sin x_5 \\ -\cos x_5 \\ -\tan x_1 \end{bmatrix} u_1 + \begin{bmatrix} -1 \\ 0 \\ -\cos x_5 \\ \sin x_5 \\ 0 \end{bmatrix} u_2 \stackrel{\Delta}{=} b_1(x)u_1 + b_2(x)u_2$$

Let $b_3 = [b_1, b_2]$, $b_4 = [b_1, [b_1, b_2]] = [b_1, b_3]$ and $b_5 = [b_2, b_3]$, it is not difficult to check that (see, e.g., [LC90])

$$C(x) = \{b_1, b_2, b_3, b_4, b_5\}$$

defines a Lie-bracket completion of the system and the Lie-bracket order is 2.

Initially, Step 3 of the algorithm gives

$$\bar{C}_l(x) = \{b_1, b_2, b_3\}$$

$$C_h(x) = \{b_4, b_5\},$$

$\tilde{C}_l(x)$ is empty and $\rho = 3$. □

Example 2.3 Consider another 2-inputs and 3-states system of the form

$$\dot{x} = b_1(x)u_1 + b_2(x)u_2 \quad (20)$$

Assume that the Lie-bracket completion matrix has the form

$$C(x) = \{b_1, b_2, [[b_1, b_2], [b_1, [b_1, b_2]]]\}$$

In other words, this is an order 3 system.

Initially, we have that

$$\bar{C}_l(x) = \{b_1, b_2\}, \quad C_h(x) = \{[[b_1, b_2], [b_1, [b_1, b_2]]]\}$$

$$\tilde{C}_l(x) = \{[b_1, b_2], [b_1, [b_1, b_2]]\}$$

and $\rho = 1$. Thus, the nominal system is given by

$$\dot{y} = b_1v_1 + b_2v_2 + [[b_1, b_2], [b_1, [b_1, b_2]]]v_3 \quad (21)$$

and the *nonholonomic system* by

$$\dot{x} = b_1w_1 + b_2w_2 + [b_1, b_2]w_3 + [b_1, [b_1, b_2]]w_4 \quad (22)$$

Applying Step 4 of the algorithm we can find the w_i 's in terms of the v_i 's.

$$\begin{aligned} w_1 &= v_1 \\ w_2 &= v_2 \\ w_3 &= \sqrt{\frac{2}{\epsilon_1}} \cos\left(\frac{t}{\epsilon_1}\right) v_3 \\ w_4 &= \sqrt{\frac{2}{\epsilon_1}} \sin\left(\frac{t}{\epsilon_1}\right) \end{aligned}$$

Then, the new *nominal system* is given by (22) while the *non-holonomic system* by

$$\dot{x} = b_1 r_1 + b_2 r_2 + [b_1, b_2] r_3. \quad (23)$$

Introduce another small parameter ϵ_2 and solving for the r'_i 's in terms of the w'_i 's we get

$$\begin{aligned} r_1 &= w_1 + \sqrt{\frac{2}{\epsilon_2}} \cos\left(\frac{t}{\epsilon_2}\right) = v_1 + \sqrt{\frac{2}{\epsilon_2}} \cos\left(\frac{t}{\epsilon_2}\right) \\ r_2 &= w_2 = v_2 \\ r_3 &= w_3 + \sqrt{\frac{2}{\epsilon_2}} \cos\left(\frac{t}{\epsilon_2}\right) w_4 \\ &= \sqrt{\frac{2}{\epsilon_1}} \cos\left(\frac{t}{\epsilon_1}\right) v_3 + \sqrt{\frac{2}{\epsilon_2}} \cos\left(\frac{2}{\epsilon_2}\right) \sqrt{\frac{2}{\epsilon_1}} \sin\left(\frac{t}{\epsilon_1}\right) \end{aligned}$$

Finally, with *nominal system* given by (23) and the nonholonomic system given by (20) the true control inputs are found to be

$$\begin{aligned} u_1 &= v_1 + \sqrt{\frac{2}{\epsilon_2}} \cos\left(\frac{t}{\epsilon_2}\right) + \sqrt{\frac{2}{\epsilon_3}} \cos\left(\frac{t}{\epsilon_3}\right) \\ u_2 &= v_2 + \sqrt{\frac{2}{\epsilon_3}} \sin\left(\frac{t}{\epsilon_3}\right) \left(\sqrt{\frac{2}{\epsilon_1}} \cos\left(\frac{t}{\epsilon_1}\right) v_3 + \sqrt{\frac{2}{\epsilon_2}} \cos\left(\frac{t}{\epsilon_2}\right) \sqrt{\frac{2}{\epsilon_1}} \sin\left(\frac{t}{\epsilon_1}\right) \right) \end{aligned} .$$

□

2.3 Averaging and Asymptotic Behaviour

In this section, we use averaging techniques to compute asymptotic behaviour of a nonholonomic system under application of some highly oscillatory inputs.

Consider the following nonholonomic system

$$\dot{x} = \sum_{i=1}^m b_i(x)u_i^\epsilon \quad (24)$$

with control input of the form

$$u_i^\epsilon = \bar{u}_i(t) + \frac{1}{\sqrt{\epsilon}}\tilde{u}_i(t, \theta) \quad (25)$$

where $\theta = t/\epsilon$, and the function \tilde{u}_i is periodic in θ with period 2π , and has zero average

$$\int_0^{2\pi} \tilde{u}_i(t, \theta) d\theta = 0. \quad (26)$$

We can identify θ as the *fast* variable, and (x, t) the *slow* variables. Using the Averaging Theorem ([Arn78], pp. 293-297) we can, for small enough ϵ , replace the above system with an “averaged” system for which the fast variable is eliminated. Moreover, the averaged system serves as a good approximation of system (24).

Theorem 2.1 *For sufficiently small ϵ , the trajectory of system (24) is bounded by solution of the following system*

$$\dot{z} = \sum_{i=1}^m b_i(z)\bar{u}_i + \frac{1}{2\pi} \sum_{i < j} [b_i, b_j] \nu_{i,j}, \quad z(0) = x(0) = x_0 \quad (27)$$

in the sense that

$$\|x - z\|_{C[0, 2\pi]} \stackrel{\Delta}{=} \Delta_\epsilon \leq \Delta_{1,\epsilon} + \Delta_{2,\epsilon}$$

where

$$\nu_{i,j} = \int_0^{2\pi} \int_0^\theta \tilde{u}_i(t, \tau) \tilde{u}_j(t, \theta) d\tau d\theta \quad (28)$$

and $(\Delta_{1,\epsilon}, \Delta_{2,\epsilon})$ are parameters which tend to zero as $\epsilon \rightarrow 0$.

In other words, system (27) defines the asymptotic behaviour of system (24) as $\epsilon \rightarrow 0$.

Proof. The derivatives of the slow variables with respect to the fast variable are seen to be

$$\begin{aligned} x' &\triangleq \frac{dx}{d\theta} = \epsilon \sum_{i=1}^m b_i(x) \bar{u}_i(t) + \sqrt{\epsilon} \sum_{i=1}^m b_i(x) \tilde{u}_i(t, \theta) \\ t' &= \epsilon \end{aligned} \quad (29)$$

Introduce new variable $y \in \Re^n$ by

$$x = y + \sqrt{\epsilon} \phi_1(y, t, \theta) + \epsilon \phi_2(y, t, \theta) + \dots \quad (30)$$

where the function (ϕ_1, ϕ_2) are 2π -periodic in θ and are chosen so that y satisfies a simpler differential equation

$$y' = \frac{dy}{d\theta} = \epsilon \left(f_1(y, t) + \sqrt{\epsilon} f_2(y, t) + \dots \right) \quad (31)$$

We can find the functions $(\phi_1, \phi_2, f_1, f_2, \dots)$ by first differentiating (30) with respect to θ ,

$$\begin{aligned} x' &= y' + \sqrt{\epsilon} \left(\frac{\partial \phi_1}{\partial y} y' + \epsilon \frac{\partial \phi_1}{\partial t} + \frac{\partial \phi_1}{\partial \theta} \right) + \epsilon \left(\frac{\partial \phi_2}{\partial y} y' + \epsilon \frac{\partial \phi_2}{\partial t} + \frac{\partial \phi_2}{\partial \theta} \right) + \dots \\ &= \sqrt{\epsilon} \frac{\partial \phi_1}{\partial \theta} + \epsilon \frac{\partial \phi_2}{\partial \theta} + \epsilon^{3/2} \frac{\partial \phi_1}{\partial t} + \epsilon^2 \frac{\partial \phi_2}{\partial t} + \left(1 + \sqrt{\epsilon} \frac{\partial \phi_1}{\partial y} + \epsilon \frac{\partial \phi_2}{\partial y} \right) y' \\ &= \sqrt{\epsilon} \frac{\partial \phi_1}{\partial \theta} + \epsilon \frac{\partial \phi_2}{\partial \theta} + \epsilon^{3/2} \frac{\partial \phi_1}{\partial t} + \epsilon^2 \frac{\partial \phi_2}{\partial t} + \dots \\ &\quad + \left(1 + \sqrt{\epsilon} \frac{\partial \phi_1}{\partial y} + \epsilon \frac{\partial \phi_2}{\partial y} \right) (\epsilon f_1(y, t) + \epsilon^{3/2} f_2(y, t) + \dots) + \dots \quad (32) \end{aligned}$$

and then computing the Taylor series expansion of (29) using (30)

$$\begin{aligned} x' &= \epsilon \sum_{i=1}^m b_i(y) \bar{u}_i(t) + \sqrt{\epsilon} \sum_{i=1}^m b_i(y) \tilde{u}_i(t, \theta) \\ &\quad + \epsilon \sum_{i=1}^m \frac{\partial b_i}{\partial y} \left(\sqrt{\epsilon} \phi_1 + \epsilon \phi_2 + \dots \right) \bar{u}_i + \sqrt{\epsilon} \sum_{i=1}^m \frac{\partial b_i}{\partial y} \left(\sqrt{\epsilon} \phi_1 + \epsilon \phi_2 + \dots \right) \tilde{u}_i \\ &\quad + \epsilon^{3/2} \sum_{i=1}^m \sum_{k=1}^n \left\langle \frac{\partial^2 b_i^k}{\partial^2 y} \phi_1, \phi_1 \right\rangle e_k \tilde{u}_i + O(\epsilon^2) \quad (33) \end{aligned}$$

where

$$b_i = \sum_{k=1}^n b_i^k e_k,$$

and $\{e_k\}_{k=1}^n$ is the canonical basis of \Re^n .

Finally, equating the coefficients of $\epsilon^{1/2}$ from Eqs. (32) and (33) we have that

$$\frac{\partial \phi_1}{\partial \theta} = \sum_{i=1}^m b_i(y) \tilde{u}_i(t, \theta) \quad (34)$$

The solution of Eq. (34) is given by

$$\phi_1(y, t, \theta) = \sum_{i=1}^k b_i(y) \int_0^\theta \tilde{u}_i(t, \tau) d\tau + \bar{\phi}_1(y, t).$$

Note that $\phi_1(y, t, \theta)$ is 2π -periodic in θ because of (26).

Equating the coefficients of ϵ we have that

$$\sum_{i=1}^m b_i(y) \bar{u}_i(t) + \sum_{i=1}^m \frac{\partial b_i}{\partial y} \phi_1 \tilde{u}_i - \frac{\partial \phi_2}{\partial \theta} = f_1(y, t) \quad (35)$$

Integrating both sides of (35) over 2π and taking into account that $f_1(y, t)$ is independent of θ yield

$$\begin{aligned} f_1(y, t) &= \frac{1}{2\pi} \int_0^{2\pi} \left(\sum_{i=1}^m b_i(y) \bar{u}_i(t) + \sum_{i=1}^m \frac{\partial b_i}{\partial y} \tilde{u}_i(t, \theta) \phi_1(y, t, \theta) - \frac{\partial \phi_2}{\partial \theta} \right) d\theta \\ &= \sum_{i=1}^m b_i(y) \bar{u}_i(t) \\ &+ \frac{1}{2\pi} \sum_{i=1}^m \int_0^{2\pi} \frac{\partial b_i}{\partial y} \tilde{u}_i(t, \theta) \left(\sum_j^m b_j \int_0^\theta \tilde{u}_j(t, \tau) d\tau + \bar{\phi}_1(y, t) \right) d\theta \\ &= \sum_{i=1}^m b_i(y) \bar{u}_i(t) + \frac{1}{2\pi} \sum_i \sum_j \frac{\partial b_i}{\partial y} b_j \int_0^{2\pi} \int_0^\theta \tilde{u}_j(t, \tau) d\tau \tilde{u}_i(t, \theta) d\theta \\ &= \sum_{i=1}^m b_i(y) \bar{u}_i(t) + \frac{1}{2\pi} \sum_{i < j} [b_i, b_j] \nu_{i,j} \end{aligned}$$

where

$$\nu_{i,j} = \int_0^{2\pi} \int_0^\theta \tilde{u}_i(t, \tau) \tilde{u}_j(t, \theta) d\tau d\theta \quad (36)$$

In the above derivation, we have made use of the facts that ϕ_2 is 2π -periodic in θ , and $\tilde{u}_i(t, \theta)$ has zero average (26). The last equality follows from integration by parts.

Using the formula for $f_1(y, t)$ we have for ϕ_2 that

$$\begin{aligned}\frac{\partial \phi_2}{\partial \theta} &= \sum_i^m b_i(y) \bar{u}_i(t) + \sum_i^m \frac{\partial b_i}{\partial y} \tilde{u}_i(t, \theta) \phi_1(y, t, \theta) - f_1(y, t) \\ &= \sum_i^m \frac{\partial b_i}{\partial y} \tilde{u}_i(t, \theta) \phi_1(y, t, \theta) - \sum_{i < j} [b_i, b_j] \nu_{i,j}\end{aligned}$$

which gives

$$\phi_2 = \int_0^\theta \left(\sum_i^m \frac{\partial b_i}{\partial y} \tilde{u}_i(t, \tau) \phi_1(y, t, \tau) - \sum_{i < j} [b_i, b_j] \nu_{i,j} \right) d\tau + \bar{\phi}_2(y, t).$$

Equating the coefficients of $\epsilon^{3/2}$ we have that

$$\sum_i^m \frac{\partial b_i}{\partial y} \phi_1 \bar{u}_i + \sum_i^m \frac{\partial b_i}{\partial y} \phi_2 \tilde{u}_i(t, \theta) + \sum_i^m \sum_{k=1}^n \langle \frac{\partial^2 f_i^k}{\partial^2 y} \phi_1, \phi_1 \rangle = f_2(y, t) + \frac{\partial \phi_1}{\partial t}$$

Integrating the above equation with respect to θ from 0 to 2π yields

$$\begin{aligned}f_2(y, t) &= \frac{1}{2\pi} \int_0^{2\pi} \sum_i^m \frac{\partial b_i}{\partial y} (\phi_1 \bar{u}_i + \phi_2 \tilde{u}_i) d\theta \\ &+ \frac{1}{2\pi} \int_0^{2\pi} \left(\sum_i^m \sum_{k=1}^n \tilde{u}_i \langle \frac{\partial^2 b_i^k}{\partial^2 y} \phi_1, \phi_1 \rangle e_k - \frac{\partial \phi_1}{\partial t} \right) d\theta \quad (37)\end{aligned}$$

We choose $\bar{\phi}_1(y, t)$ to be

$$\bar{\phi}_1(y, t) = - \int_0^{2\pi} \sum_i^m b_i(y) \int_0^{2\pi} \tilde{u}_i(t, \tau) d\tau d\theta$$

so that $\phi_1(y, t, \theta)$ has zero average,

$$\int_0^{2\pi} \phi_1(y, t, \theta) d\theta = 0$$

Using the above property of ϕ_1 in Eq. (37) we have

$$f_2(y, t) = \frac{1}{2\pi} \int_0^{2\pi} \left(\sum_i^k \frac{\partial b_i}{\partial y} \phi_2 \tilde{u}_i(t, \theta) + \sum_i^m \sum_{k=1}^n \tilde{u}_i \langle \frac{\partial^2 b_i^k}{\partial^2 y} \phi_1, \phi_1 \rangle e_k \right) d\theta$$

Finally, we have the following form for the derivative of y with respect to the slow variable t

$$\begin{aligned}\frac{dy}{dt} &= \frac{1}{\epsilon} \frac{dy}{d\theta} = f_1(y, t) + \epsilon^{1/2} f_2(y, t) + \dots \\ &= \sum_{i=1}^m b_i(y) \bar{u}_i + \frac{1}{2\pi} \sum_{i < j} [b_i, b_j] \nu_{i,j} + \epsilon^{1/2} f_2(y, t) + O(\epsilon)\end{aligned}\quad (38)$$

and

$$x(t) = y + \sqrt{\epsilon} \phi_1 + \epsilon \phi_2 + \dots$$

with $x(0) = y(0) = x_0$.

Define the averaged system by

$$\dot{z} = \sum_{i=1}^m b_i(z) \bar{u}_i(t) + \frac{1}{2\pi} \sum_{i < j} [b_i, b_j] \nu_{i,j}, \quad z(0) = y(0) = x(0). \quad (39)$$

The error between the averaged system and the original system can be computed using techniques from ([Arn78]) and is given by

$$\Delta_\epsilon = \sup_{t \in [0, 2\pi]} \|x(t) - z(t)\| \leq \Delta_{1,\epsilon} + \Delta_{2,\epsilon}$$

where

$$\Delta_{1,\epsilon} = \|x - y\|_{C[0, 2\pi]}, \quad \Delta_2 = \|z - y\|_{C[0, 2\pi]}$$

For sufficiently small ϵ , we also have

$$\begin{aligned}\Delta_{1,\epsilon} &\leq \sqrt{\epsilon} \|\phi_1\| + O(\epsilon) \\ \Delta_{2,\epsilon} &\leq \sqrt{\epsilon} (e^{2\pi K}/K) \|f_2\|_{C[0, 2\pi]}\end{aligned}$$

where

$$K = \sum_{i=1}^m \left\| \frac{\partial b_i}{\partial y} \right\| \cdot \|\bar{u}_i\| + \sum_{i < j} \left\| \frac{\partial [b_i, b_j]}{\partial y} \right\| \cdot \|\nu_{i,j}\|$$

and all norms are the maximum norm defined in some neighborhood of $z(t)$. \square

In view of Eq. (28) the problem is to find, for given $\nu_{i,j}$, ($i < j$), 2π -periodic function $\tilde{u}_j(t, \theta)$ which has zero average and satisfies

$$\frac{1}{2\pi} \int_0^{2\pi} \tilde{u}_j(t, \theta) \int_0^\theta \tilde{u}_i(t, \tau) d\tau d\theta = \frac{1}{2\pi} \nu_{i,j}(t) \quad (40)$$

The solution to the problem, using Fourier method of separation of variables, has the form

$$\tilde{u}_i(t, \theta) = \sum_{p=1}^k \alpha_p^i(t) \sin p\theta + \beta_p^i(t) \cos p\theta \quad (41)$$

Define

$$\alpha_p = (\alpha_p^1, \dots, \alpha_p^m)^T, \quad \beta_p = (\beta_p^1, \dots, \beta_p^m)^T$$

It is easy to check, through direct integration, that

$$\text{Skew}(\nu_{i,j}) = \pi \sum_{p=1}^k \frac{1}{p} \beta_p \wedge \alpha_p \quad (42)$$

where

$$\beta_p \wedge \alpha_p = \beta_p \alpha_p^T - \alpha_p \beta_p^T.$$

In other words, we have to solve the following *factorization* problem: For a given skew-symmetric matrix S , find β_p, α_p such that

$$S = \sum_{p=1}^k \beta_p \wedge \alpha_p \quad (43)$$

In the case $k = m(m - 1)/2$, the solution is given by

$$\text{Skew}(\nu_{i,j}) = \sum_{1 \leq i < j \leq m} (\nu_{i,j} e_i \wedge e_j)$$

Remark 2.3 Here the elements of S, β_p and α_p belong to some commutative algebra X over \mathfrak{R} . In our context it is the functional algebra. In other words, we are solving nonholonomic motion planning problem for the following infinite dimensional system

$$\begin{aligned} \dot{x} &= u, \quad x(0) = x(2\pi), \\ \dot{S} &= x \wedge u, \quad S(2\pi) - S(0) = \text{Skew}(\nu_{i,j}) \end{aligned}$$

using finite Fourier representation for the control input

$$u = \sum_p \alpha_p \sin pt + \beta_p \cos pt$$

This motivates one to study the canonical systems introduced by R. Brockett ([Bro81]). In fact, the same ideas can be applied to

the family of chained systems of higher Lie bracket orders, with states and control inputs belonging to an arbitrary commutative algebra over \mathfrak{R} . For example, the algorithm given in [MS90] works for any commutative algebra.

If the algebra is the reals or the complex numbers, we can easily find out the minimal k solving Eq. (43), namely

$$\min(k) = \text{rank}(S)/2$$

and the solution is given by the so-called symplectic decomposition of S . In the general case the problem is difficult to solve.

Proposition 2.1 *Assume the following input to system (29)*

$$u_i = \bar{u}_i(x) + \sum_{k=1}^{m(m-1)/2} \alpha_i^k(x) \sin(k \frac{t}{\epsilon}) + \beta_i^k(x) \cos(k \frac{t}{\epsilon}) \quad (44)$$

then the asymptotic trajectory is defined by solution of system (45)

$$\dot{z} = \sum_{i=1}^m \delta_i b_i(x) + \sum_{1 \leq i < j \leq m} [b_i, b_j] \nu_{i,j}, \quad z(0) = x(0) \quad (45)$$

where

$$\begin{aligned} \delta_i &= \bar{u}_i(x) + \sum_{j=1}^m \sum_{k=1}^{m(m-1)/2} \left(\langle \nabla \beta_i^k, b_j \rangle \alpha_j^k - \langle \nabla \alpha_i^k, b_j \rangle \beta_j^k \right) \text{sgn}(j-i) \frac{1}{k} \\ Skew(\nu_{i,j}) &= \sum_{k=1}^{m(m-1)/2} \left(\beta^k \wedge \alpha^k \right) \frac{1}{k} \end{aligned}$$

and

$$\beta^k = (\beta_1^k \cdots \beta_m^k)^T, \quad \alpha_k = (\alpha_1^k \cdots \alpha_m^k)^T.$$

Proof. Apply the control input (44) to system (29) we have

$$\dot{x} = \sum_{i=1}^m G_i(x) + \sqrt{\frac{2}{\epsilon}} \sum_{k=1}^{m(m-1)/2} R_k(x) \sin\left(\frac{kt}{\epsilon}\right) + Q_k(x) \cos\left(\frac{kt}{\epsilon}\right)$$

where

$$\begin{aligned} G_i(x) &= b_i(x)\bar{u}_i(x) \\ R_k(x) &= \sum_{i=1}^m b_i(x)\alpha_i^k(x) \\ Q_k(x) &= \sum_{i=1}^m b_i(x)\beta_i^k(x) \end{aligned}$$

The desired result follows from the proof of Theorem 2.1 and the fact that

$$[b_1\alpha(x), b_2\beta(x)] = \alpha\beta[b_1, b_2] + b_2\langle\nabla\beta, b_1\rangle - b_1\langle\nabla\alpha, b_2\rangle.$$

□

3 Controllability and Holonomy

In this section we provide the motivation for using the control input in Algorithm 2.1.

Consider a system of the form

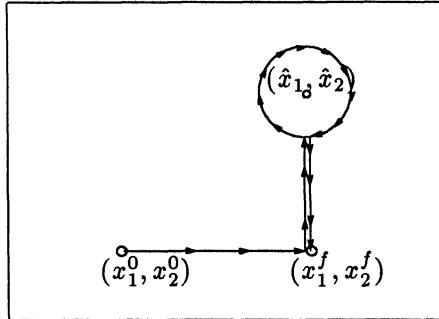
$$\begin{aligned} \dot{x}_1 &= u_1 \\ \dot{x}_2 &= u_2 \\ \dot{x}_3 &= f_1(x_1, x_2)u_1 + f_2(x_1, x_2)u_2 \end{aligned}$$

where $(x_1, x_2, x_3)^T \in \Re^3$. Assume that f_1, f_2 are smooth functions and define by

$$g(x_1, x_2) = \frac{\partial f_2}{\partial x_1} - \frac{\partial f_1}{\partial x_2}.$$

If there exists a ball $B_\epsilon \in \Re^2$ such that $g(x_1, x_2) \neq 0$ for all (x_1, x_2) inside the ball and zero outside the ball, then the system is controllable but not locally-locally controllable (see, e.g., [HH70]). To see this, let $x_0 = (x_1^0, x_2^0, x_3^0)$ and $x_f = (x_1^f, x_2^f, x_3^f)$ be two given points and $(\hat{x}_1, \hat{x}_2) \in \Re^2$ a point such that $g(\hat{x}_1, \hat{x}_2) \neq 0$. Suppose that Ω is a closed ball in B_ϵ containing (\hat{x}_1, \hat{x}_2) with boundary $\delta\Omega$, then by Green's Theorem we have that

$$\int_{\delta\Omega} f_1(x_1, x_2)dx_1 + f_2(x_1, x_2)dx_2 = \iint_{\Omega} g(x_1, x_2)dx_1 dx_2.$$

Figure 2: Holonomy of a control path in \mathbb{R}^2 .

A trajectory which links x_0 to x_f is shown in Figure 2, where control input $u(t) \in \mathbb{R}^2, t \in [0, T_1]$, is chosen to steer the system from (x_1^0, x_2^0) to (x_1^f, x_2^f) . Then, Ω is chosen so that

$$n = \left(x_3^f - x_3^0 - \int_0^{T_1} f_1 u_1 dt + f_2 u_2 dt \right) \left(\iint_{\Omega} g(x_1, x_2) dx_1 dx_2 \right)^{-1}$$

is an integer (could be negative) and control input $u(t) \in \mathbb{R}^2, t \in [T_1, T]$, is chosen to cause the system to encircle Ω n times, clockwise if n is negative. It is quite clear that the system is not locally-locally controllable for a point lying outside the B_ϵ ball. In order for it to be locally-locally controllable the set $\{(x_1, x_2) \in \mathbb{R}^2, g(x_1, x_2) \neq 0\}$ has to be dense in \mathbb{R}^2 .

Generalizing from the above example, we have the following result.

Theorem 3.1 Consider a nonlinear control system of the form

$$\begin{aligned} \dot{x} &= u \\ \dot{y} &= f(x)u \end{aligned} \tag{46}$$

where $\{(x, y), x \in \mathbb{R}^n, y \in \mathbb{R}^m\}$ is the state, $u \in \mathbb{R}^n$ the control input, and $f(x) = [f_1(x), \dots, f_n(x)]$ a collection of x -dependent C^1 vector fields. The system is controllable if and only if the span of

$$\{g_{ij}(x), x \in \mathbb{R}^n, 1 \leq i, j \leq n\}$$

is equal to \Re^m , where

$$g_{ij} = \frac{\partial f_i}{\partial x_j} - \frac{\partial f_j}{\partial x_i}$$

In order to prove Theorem 3.1, we need the following lemma.

Lemma 3.1 (1) Let $c : [0, 1] \rightarrow \Re^m$, $c(0) = 0$, be a C^1 curve and associate with c the following grid

$$G(c) = \{y \in \Re^m, y = \sum_{i=1}^k n_i c(t_i), n_i \in \mathbb{Z}, t_i \in [0, 1], i \in [1, k], 1 \leq k \leq \infty\}$$

we claim that $G(c)$ is in fact a linear subspace of \Re^m .

(2) Moreover,

$$G(c) = \left\{ \sum_{i=1}^k \pm n c(t_i), n \in N, t_i \in [0, 1] \right\}$$

is a subspace of \Re^m of dimension at least $k/2$, where Z denotes the set of integers and N the set of nonnegative integers.

Remark 3.1 The lemma does not hold for ∞ -dimensional space. Consider, for example, the case $c(t) \in L_2[0, 1]$ such that

$$c(t)(x) = \begin{cases} 1 - x/t, & 0 \leq x < t, \\ 0, & t \leq x \leq t, \end{cases}$$

for $t \neq 0$ and $c(0) = 0$. Then, $\tilde{c}(t) \triangleq c(t^2)$ is C^1 and $G(\tilde{c})$ is not a linear subspace, because for any $\mu \in G(\tilde{c})$ we have that

$$\mu(0) \in N.$$

□

Proof of Lemma 3.1. Since

$$c(t) = \int_0^t c'(\tau) d\tau$$

so we have that

$$G(c) \subseteq \text{span}\{c'(\tau), \tau \in [0, 1]\} \triangleq L$$

We will show that

$$G(c) = L \quad (47)$$

Let l be the dimension of L and pick l points $\{\tau_i\}_{i=1}^l$ such that

$$\{c'(\tau_i), i \in [1, l]\}$$

forms a basis of L .

Define a map $C : \Re^l \longrightarrow L$ by

$$C(\delta_1, \dots, \delta_l) = \sum_{i=1}^l c(\tau_i + \delta_i)$$

The directional derivative of C is

$$\frac{\partial C}{\partial \delta_i}|_{\delta_i=0} = c'(\tau_i)$$

and thus the Jacobian of C is

$$[c'(\tau_1), \dots, c'(\tau_l)] \triangleq D$$

It is clear that D is nonsingular and by the rank Theorem, for sufficiently small δ , $C(B_\delta(0))$ has nonempty interior in L , where $B_\delta(0)$ is an open-ball of radius δ . Thus, by construction we have that

$$\text{Im}(C) \subset G(c)$$

On the other hand, $G(c)$ has the additional properties that

$$G(c) + G(c) \subset G(c)$$

and

$$kG(c) \subset G(c), \quad \forall k \in Z.$$

Using these facts we conclude that $G(c) = L$. \square

Proof of Theorem 3.1. To show that the condition is necessary, suppose that

$$\text{Span}\{g_{ij}(x), x \in \Re^n, 1 \leq i, j \leq n\} \neq \Re^m$$

Then, there exists a vector $e \in \Re^m$ such that

$$\langle e, g_{ij} \rangle = 0, \forall 1 \leq i, j \leq n$$

Consider the differential equation for $\langle y, e \rangle$

$$\langle y, e \rangle' = \langle f(x)u, e \rangle = \sum_{i=1}^n \langle f_i(x), e \rangle \dot{x}_i = \sum_{i=1}^n \tilde{f}_i(x) \dot{x}_i,$$

where $\tilde{f}_i(x) = \sum_{j=1}^m f_i^j(x) e_j$.

Note that $\tilde{f}_i : \Re^n \rightarrow \Re$ is a C^1 function. It is clear that

$$\frac{\partial \tilde{f}_i}{\partial x_j} - \frac{\partial \tilde{f}_j}{\partial x_i} = \langle e, g_{ij}(x) \rangle = 0, 1 \leq i, j \leq n$$

In other words, the integrability condition is satisfied and there exists a C^2 function g such that

$$\frac{\partial g}{\partial x_i} = \tilde{f}_i$$

Therefore, for any $x(0) = x(1)$ we have that

$$\langle y(0), e \rangle = \langle y(1), e \rangle$$

This contradicts the assumption that the system is controllable.

To show that the system is controllable at (x_0, y_0) , choose m points $\{\bar{x}_k\}_{k=1}^m$ such that

$$\{g_{i_k j_k}(\bar{x}_k), 1 \leq k \leq m, 1 \leq i_k, j_k \leq n\} \triangleq F$$

forms a basis of \Re^m . We will construct a curve that takes the system from (x_0, y_0) to an arbitrary state (x, y) as follows:

First, denote by $S_k(\epsilon)$ a simple closed curve based at \bar{x}_k and lying on the $x_{i_k} - x_{j_k}$ plane, e.g., circle or square,

$$\begin{aligned} S_k(\epsilon) &= \{(x^1, \dots, x^n), x^i = \bar{x}_k^i, i \neq i_k, j_k, x^{i_k} \in [\bar{x}_k^{i_k}, \bar{x}_k^{i_k} + \epsilon], \\ &\quad x^{j_k} \in [\bar{x}_k^{j_k}, \bar{x}_k^{j_k} + \epsilon]\} \end{aligned}$$

Define a primitive curve $M(k, \epsilon, N)$ in the x or the control space to be a curve with the following three segments:

1. The first segment links x_0 to \bar{x}_k .
2. The second segment traverses $S_k(\epsilon)$ N times, clockwise if $N > 0$ and counter clockwise if $N < 0$.
3. The last segment is the negative of segment 1, and thus links \bar{x}_k back to x_0 .

Let c_k be a curve of \Re^m such that

$$c_k(\epsilon) = \iint_{S_k(\epsilon)} g_{i_k j_k}(x) dx_{i_k} dx_{j_k}, \quad c_k(0) = 0$$

Then, by Green's Theorem, the final destination of traversing the primitive curve $M(k, \epsilon, N)$ is

$$(x_0, y_0 + nc_k(\epsilon))$$

Therefore, if we use all possible motions of $M(k, \epsilon, N)$ we can achieve points of the form

$$\{(x_0, y) : y \in y_0 + \sum_{k=1}^m G(c_k)\}$$

By Lemma 3.1,

$$\sum_{k=1}^m G(c_k)$$

is a linear subspace of \Re^m , which contains the set

$$B \triangleq \{g_{i_k j_k}(\bar{x}_k) \epsilon^2 + o(\epsilon^2), 1 \leq k \leq m\}$$

But, by assumption B forms a basis of \Re^m for sufficiently small ϵ . Therefore,

$$\sum_{k=1}^m G(c_k) = \Re^m$$

and

$$y_0 + \sum_{k=1}^m G(c_k) = \Re^m.$$

□

Using the constructive procedure in the proof of Theorem 3.1, we can obtain an upper bound for the optimal control input.

Proposition 3.1 Consider the system

$$\begin{aligned}\dot{x} &= u \\ \dot{y} &= f(x)u\end{aligned}$$

with boundary conditions $x(0) = x(1) = x_0$, $y(0) = y_0$ and $y(1) = y_f$. Here $(x, y) \in \mathbb{R}^n \times \mathbb{R}^m$. Assume that there exist m points $\{\bar{x}_k\}_{k=1}^m$ such that

$$\{g_{i_k j_k}(\bar{x}_k), 1 \leq k \leq m, 1 \leq i_k, j_k \leq n\}$$

forms a basis of \mathbb{R}^m . Then the optimal control input, which is piecewise smooth, is bounded above by

$$\int_0^1 \|u(t)\|^2 dt \leq C_1 \|y_f - y_0\|^2 \left(C_2 \sum_{k=1}^m \|x_0 - \bar{x}_k\|^2 + C_3 \right) \quad (48)$$

where C_1, C_2 and C_3 are constants.

Proof. From the proof of Theorem 3.1, we know that there exists a ball of radius σ , $B_\sigma(0)$, which lies in the set

$$\left\{ \sum_{k=1}^m c_k(\epsilon_k^1) - c_k(\epsilon_k^2), 0 \leq \epsilon_k^i \leq \bar{\epsilon} \right\} \supset B_\sigma(0).$$

Therefore, we can express

$$y_f - y_0 = \eta \left(\sum_{k=1}^m c_k(\epsilon_k^1) - c_k(\epsilon_k^2) \right) \quad (49)$$

where

$$\eta = [\|y_f - y_0\|/\sigma] + 1, \quad \epsilon_k^i \in [0, \bar{\epsilon}].$$

and $[\|y_f - y_0\|/\sigma]$ denotes the integer part of $\|y_f - y_0\|/\sigma$.

First, construct control input $u(t), t \in [0, 1]$, so that the resulting x -trajectory is the composition of the following primitive motions

$$M(1, \epsilon_1, 1), \dots M(m, \epsilon_m, 1)$$

For example, the associated control input for $M(1, \epsilon_1, 1)$ has the form

$$u = \begin{cases} 3m(\bar{x}_1 - x_0), & t \in [0, \frac{1}{3m}], \\ \tilde{u}((t - \frac{2}{3m})3m), & t \in (\frac{1}{3m}, \frac{2}{3m}], \\ -3m(\bar{x}_1 - x_0), & t \in (\frac{2}{3m}, \frac{1}{m}] \end{cases}$$

where

$$\tilde{u} = (0, \dots, u_{i_1}(t), 0, \dots, u_{j_1}(t), 0, \dots, 0)^T$$

and

$$u_{i_1}(t) = \begin{cases} \epsilon_1, & t \in [0, \frac{1}{4}], \\ 0, & t \in (\frac{1}{4}, \frac{1}{2}], \\ -\epsilon_1, & t \in (\frac{1}{2}, \frac{3}{4}], \\ 0, & t \in [\frac{3}{4}, 1] \end{cases} \quad u_{j_1} = \begin{cases} 0, & t \in [0, \frac{1}{4}], \\ \epsilon_1, & t \in (\frac{1}{4}, \frac{1}{2}], \\ 0, & t \in (\frac{1}{2}, \frac{3}{4}], \\ -\epsilon_1, & t \in (\frac{3}{4}, 1] \end{cases}$$

The corresponding cost is computed to be

$$\int_0^{1/m} \langle u, u \rangle dt = \frac{2}{3m} \|x_0 - \bar{x}_1\|^2 + 3m\epsilon_1^2$$

and the total cost is

$$I_1 = \frac{2}{3m} \sum_{i=1}^m \|x_0 - \bar{x}_i\|^2 + 3m \sum_{i=1}^m \epsilon_i^2.$$

Then, using the above control input and time scaling to construct new control input so that the resulting x -trajectory is the composition of the primitive motions

$$M(1, \epsilon_1, \eta), \dots, M(m, \epsilon_m, \eta)$$

Obviously, the final input cost is given by

$$\eta^2 I_1$$

and from which we have

$$\eta^2 I_1 \leq \frac{\|y_f - y_0\|^2}{\sigma^2} \left(\frac{2}{3m} \sum_{i=1}^m \|x_0 - \bar{x}_i\|^2 + 3m \sum_{i=1}^m \epsilon_i^2 \right).$$

□

Proposition 3.2 *Under the same conditions as in Proposition 3.1, if we restrict that*

$$\|x(t)\| \leq \Omega$$

and assume that $\|y_f - y_0\|$ is sufficiently large, then the optimal control input

$$I_{op} = \int_0^1 \langle u, u \rangle dt$$

is bounded by the following inequalities

$$C_1 \|y_f - y_0\|^2 \leq I_{op} \leq C_2 \|y_f - y_0\|^2 \quad (50)$$

where C_1 and C_2 are some constants.

Proof. Because of Proposition 3.1 we only need to show the lower bound. The following inequalities follow from Schwartz inequality

$$\begin{aligned} \|y_f - y_0\| &= \left\| \int_0^1 f(x) u dt \right\| \\ &\leq \left(\int_0^1 \|f(x)\|^2 dt \right)^{1/2} \left(\int_0^1 \langle u, u \rangle dt \right)^{1/2} \\ &\leq \max_{\|x\| \leq \Omega} \|f(x)\| \cdot \left(\int_0^1 \langle u, u \rangle dt \right)^{1/2} \end{aligned}$$

Therefore, we have that

$$\int_0^1 \langle u, u \rangle dt \geq \frac{\|y_f - y_0\|^2}{\left(\max_{\|x\| \leq \Omega} \|f(x)\| \right)^2} \triangleq C_1 \|y_f - y_0\|^2.$$

□

From the proof of Proposition 3.1, we observe that a system of nonlinear equations, namely that defined by Eq. (49), needs to be solved if we use inputs of bounded quadratic cost. It is, however, possible to obtain asymptotic control input by solving only a system of linear equations if we allow the cost to be unbounded quadratic functions. We present such an algorithm below for system of the form

$$\begin{aligned} \dot{x} &= u, \quad x \in \Re^n, \\ \dot{y} &= f(x)u, \quad y \in \Re^m \end{aligned}$$

with boundary conditions

$$x(0) = x(1) = x_0, \quad y(1) - y(0) = \delta.$$

Algorithm 3.1 Step 1 Choose a small $\epsilon > 0$.

Step 2 Compute the vector $z = (z_1, \dots, z_m)^T$ by solving the following system of linear equations

$$\epsilon^2 Dz = \delta$$

where

$$D = \{g_{i_1, j_1}(\bar{x}_1), \dots, g_{i_m, j_m}(\bar{x}_m)\}$$

Step 3 Use the control input in the proof of Proposition 3.1 to cause the x -trajectory to follow the following primitive motions

$$M(1, \epsilon, [z_1]), M(2, \epsilon, [z_2]), \dots, M(m, \epsilon, [z_k])$$

where $[z_k]$ denotes the closest integer to z_k .

It is easy to see that the contribution of the above primitive motions in the y -space has the form

$$\tilde{\delta} = \sum_{k=1}^m [z_k] (g_{i_k, j_k} \epsilon^2 + O(\epsilon^2))$$

where

$$g_{i_k, j_k} \epsilon^2 + O(\epsilon^2) = C_k(\epsilon) = \iint_{S_k(\epsilon)} g_{i_k, j_k}(x) dx_{i_k} dx_{j_k}$$

The error term is bounded by

$$\|\tilde{\delta} - \delta\| \leq \|\tilde{\delta} - \bar{\delta}\| + \|\bar{\delta} - \delta\| \quad (51)$$

and

$$\bar{\delta} = \sum_{k=1}^m z_k (g_{i_k, j_k} \epsilon^2 + O(\epsilon^2))$$

The first term in (51) is in the order of ϵ^2 ,

$$\|\tilde{\delta} - \bar{\delta}\| \sim \epsilon^2.$$

The second term is in the order of 1,

$$\begin{aligned} \bar{\delta} - \delta &= (\epsilon^2 D + O(\epsilon^2)) \epsilon^{-2} D^{-1} \delta - \delta \\ &= ((D + O(1)) D^{-1} - I) \delta \sim O(1). \end{aligned}$$

However, if $f(x)$ is twice continuously differentiable, then it is possible to show that $C_k(\epsilon) = g_{i_k, j_k} \epsilon^2 + O(\epsilon^3)$ and

$$\|\tilde{\delta} - \delta\| \sim \epsilon. \quad (52)$$

For example, if the primitive motions, $M(k, \epsilon, [z_k])$, are circles of radius ϵ with time interval $[0, 2\pi]$, then (52) holds. In this case, the control input has the form

$$\frac{1}{\epsilon} \sin\left(\frac{t}{\epsilon^2}\right), \frac{1}{\epsilon} \cos\left(\frac{t}{\epsilon^2}\right). \quad (53)$$

Remark 3.2 The results of the above algorithm provide the motivation for using highly oscillatory control inputs in Section 2. \square

3.1 Controllability of Linearized Nonholonomic System

In this section, we discuss linearization of a nonholonomic system around a nominal trajectory and controllability properties of the linearized system.

Consider a nonholonomic system of the form

$$\dot{x} = A(x)u, \quad x \in \Re^n, \quad (54)$$

where $u = (u_1, \dots, u_m)^T$, and $A = (A_1, \dots, A_m)$.

Let \bar{u} be a nominal control input to system (54) and δu a perturbation term. Apply the perturbed control input $u = \bar{u} + \delta u$ to (54) and linearize the system around the nominal trajectory \bar{x} , we obtain the following differential equation for the linearized system

$$\frac{d}{dt}(\delta x) = \left(\sum_{i=1}^m \frac{\partial A_i}{\partial x} u_i \right) \delta x + A(\bar{x})\delta u. \quad (55)$$

where

$$x = \bar{x} + \delta x + o(\|\delta x\|)$$

Note that system (55) is linearized about a nominal trajectory, not about a fixed point as in usual practice.

We have the following results for controllability of the linearized system.

Theorem 3.2 *Assume that the columns of $A(x)$ are C^1 functions, and there exist constants C and B such that*

$$\|A(x)\| \leq C + B\|x\|$$

then system (54) is controllable if and only if for every x_0 there exists control input $u(\cdot)$ such that the linearized system (55) is controllable.

Remark 3.3 *The above theorem is used in [Cor91] to construct smooth, time-periodic feedbacks for stabilization of system (54).*

To illustrate Theorem 3.2, consider the the following simpler system

$$\begin{aligned}\dot{x}_1 &= u_1 \\ \dot{x}_2 &= u_2 \\ \dot{y} &= f_1(x)u_1 + f_2(x)u_2, \quad y \in \mathbb{R}^m.\end{aligned}\tag{56}$$

Linearize the system about a nominal trajectory $(\bar{u}_1, \bar{u}_2) \in L_2[0, 1]$, we have

$$\begin{aligned}\frac{d}{dt}(\delta x_1) &= \delta u_1 \\ \frac{d}{dt}(\delta x_2) &= \delta u_2 \\ \frac{d}{dt}(\delta y) &= \left(\sum_{i=1}^2 \frac{\partial f_i}{\partial x_1} \bar{u}_i \right) \delta x_1 + \left(\sum_{i=1}^2 \frac{\partial f_i}{\partial x_2} \bar{u}_i \right) \delta x_2 + \sum_{i=1}^2 f_i(\bar{x}) \delta u_i.\end{aligned}\tag{57}$$

It is easy to see that the linearized system is controllable if and only if it is possible to reach from the origin any state of the form $(0, 0, \delta y)$. Therefore, we can assume that

$$\int_0^1 \delta u_1 dt = \int_0^1 \delta u_2 dt = 0$$

Integrating the third equation of (57) by parts and using the following boundary conditions

$$\delta x_i(0) = \delta x_i(1) = 0, \quad i = 1, 2$$

yield

$$\begin{aligned} \delta y(1) &= \int_0^1 \left(\frac{\partial f_1}{\partial x_1} \bar{u}_1 + \frac{\partial f_2}{\partial x_1} \bar{u}_2 \right) \delta x_1 dt + \int_0^1 \left(\frac{\partial f_1}{\partial x_2} \bar{u}_1 + \frac{\partial f_2}{\partial x_2} \bar{u}_2 \right) \delta x_2 dt \\ &+ \int_0^1 (f_1(\bar{x}) \delta u_1 + f_2(\bar{x}) \delta u_2) dt \\ &= \int_0^1 \left(\frac{\partial f_1}{\partial x_1} \bar{u}_1 + \frac{\partial f_2}{\partial x_1} \bar{u}_2 \right) \delta x_1 dt + \int_0^1 \left(\frac{\partial f_1}{\partial x_2} \bar{u}_1 + \frac{\partial f_2}{\partial x_2} \bar{u}_2 \right) \delta x_2 dt \\ &+ \int_0^1 \left(f_1(\bar{x}) \frac{d}{dt} (\delta x_1) + f_2(\bar{x}) \frac{d}{dt} (\delta x_2) \right) dt \\ &= \int_0^1 \left(\frac{\partial f_2}{\partial x_1} - \frac{\partial f_1}{\partial x_2} \right) (\bar{u}_2 \delta x_1 - \bar{u}_1 \delta x_2) dt \end{aligned}$$

Assuming for simplicity that $(\bar{u}_1(t), \bar{u}_2(t))$, $t \in [0, 1]$, are continuous functions, then we have the following criteria for controllability of the linearized system: *there exist $t_1, \dots, t_m \in [0, 1]$ such that*

$$\begin{aligned} \bar{u}_1^2(t_i) + \bar{u}_2^2(t_i) &\neq 0 \\ \text{Span} \left(\left(\frac{\partial f_1}{\partial x_2} - \frac{\partial f_2}{\partial x_1} \right) (t_i), 1 \leq i \leq m \right) &= \mathfrak{R}^m. \end{aligned}$$

Note that we can use the above result to provide an alternative proof of Theorem 3.1.

References

- [Arn78] V. Arnold. *Mathematical Methods of Classical Mechanics*. Springer-Verlag, 2nd edition, 1978.
- [BD91] R. Brockett and L. Dai. Non-holonomic kinematics and the role of elliptic functions in constructive controllability. Technical report, Applied Science Division, Harvard University, 1991.

- [BMR90] A. Bloch, N.H. McClamroch, and M. Reyhanoglu. Controllability and stabilizability properties of a nonholonomic control system. In *Conf. on Decision and Control*, pages 1312–1314, 1990.
- [Bro81] R. Brockett. Control theory and singular riemannian geometry. In P. Hilton and G. Young, editors, *New Directions in Applied Mathematics*, pages 11 –27. Springer-Verlag, 1981.
- [Bro83] R. Brockett. Asymptotic stability and feedback stabilization. In R. Millman R. Brockett and H. Sussmann, editors, *Differential Geometry and Geometrical Control Theory*. Birkhauser, Boston, 1983.
- [Bro88] R. W. Brockett. On the rectification of vibratory motion. In *Proceedings of Microactuators and Micromechanism*, Salt Lake City, Utah, 1988.
- [Can88] J.F. Canny. *The complexity of robot motion planning*. MIT Press, 1988.
- [Cor91] J. M. Coron. Global asymptotical stabilization for controllable system without drift. Technical report, preprint, 1991.
- [FGL91a] C. Fernandes, L. Gurvits, and Z.X. Li. An efficient algorithm for optimal nonholonomic motion planning. In *IEEE Int. Conf. on Robotics and Automation*, 1991.
- [FGL91b] C. Fernandes, L. Gurvits, and Z.X. Li. Foundations of nonholonomic motion planning. Technical report, Robotics Research Laboratory, Courant Institute of Mathematical Sciences, 1991.
- [Fri81] S. Friedland. Convex spectral functions. *Linear and multilinear algebra*, pages 299–317, 1981.
- [Gur91] L. Gurvits. Averaging approach to nonholonomic motion planning. Technical report, Robotics Laboratory, Courant Institute of Mathematical Sciences, New York University, 1991.

- [HH70] G.W. Haynes and H. Hermes. Nonlinear controllability via lie theory. *SIAM J. Control*, 8(4):450–460, 1970.
- [JC89] P. Jacobs and J. Canny. Planning smooth paths for mobile robots. In *IEEE Int. Conf. on Robotics and Automation*, pages 2–7, 1989.
- [Kod87] D.E. Koditschek. Exact robot navigation by means of potential functions: some topological considerations. In *IEEE Int. Conf. on Robotics and Automation*, pages 1–6, 1987.
- [Kri90] P.S. Krishnaprasad. Geometric phases, and optimal re-configuration for multibody systems. Technical report, System research center, University of Maryland at College Park, 1990.
- [Lau92] J. Laumond. Singularities and topological aspects in nonholonomic motion planning. In Z.X. Li and J. Canny, editors, *Progress in Nonholonomic Motion Planning*. Kluwer Academic Publisher, 1992.
- [LC90] Z. Li and J. Canny. Motion of two rigid bodies with rolling constraint. *IEEE Trans. on Robotics and Automation*, RA2-06:62–72, 1990.
- [Li89] Z. Li. *Kinematics, Planning and Control of Dextrous Robot Hands*. PhD thesis, Dept. of EECS, Univ. of Calif. at Berkeley, 1989.
- [LS90] G. Lafferriere and H. Sussmann. Motion planning for controllable systems without drift: A preliminary report. Technical report, Rutgers University Systems and Control Center Report SYCON-90- 04, 1990.
- [Mon89] R. Montgomery. Optim. contr. of deform. body, isohol. prob., and sub-riem. geom. Technical report, MSRI, UC Berkeley, 1989.
- [Mon92] R. Montgomery. Nonholonomic control and gauge theory. In Z.X. Li and J. Canny, editors, *Progress in Nonholonomic Motion Planning*. Kluwer Academic Publisher, 1992.

- [MS90] R. Murray and S. Sastry. Grasping and manipulation using multifingered robot hands. Technical report, University of California at Berkeley, 1990.
- [MS92] R. Murray and S. Sastry. Steering nonholonomic control systems using sinusoids. In Z.X. Li and J. Canny, editors, *Progress in Nonholonomic Motion Planning*. Kluwer Academic Publisher, 1992.
- [NM90] Y. Nakamura and R. Mukherjee. Nonholonomic path planning of space robotics via bidirectional approach. In *IEEE Int. Conf. on Robotics and Automation*, 1990.
- [PD90] E. Papadopoulos and S. Dubowsky. On the nature of control algorithms for space manipulators. In *IEEE Int. Conf. on Robotics and Automation*, 1990.
- [Rim90] E. Rimon. *Exact Robot Navigation Using Artificial Potential Functions*. PhD thesis, Yale University, 1990.
- [RMB92] M. Reyhanoglu, N.H. McClamroch, and A. Bloch. Motion planning for nonholonomic dynamic systems. In Z.X. Li and J. Canny, editors, *Progress in Nonholonomic Motion Planning*. Kluwer Academic Publisher, 1992.
- [SL91] H. Sussmann and W. Liu. Limits of highly oscillatory controls and the approximation of general paths by admissible trajectories. Technical report, Rutgers University Center for System and Control Report, SYCON-91-02, 1991.
- [SS86] J. Schwartz and M. Sharir. Planning, geometry and complexity of robot motion. In *Proc. of Int. Congress of Math.*, Berkeley, 1986.
- [VG88] A. M. Vershik and V. Ya. Gershkovich. Nonholonomic problems and the theory of distribution. *ACTA Applicandae Mathematicae*, Vol. 12, No. 2:181–210, 1988.
- [Yap87] C. Yap. *Advances in Robotics*. Lawrence Erlbaum Assoc., 1987.

Appendix A: Comparision with Coron's Results

It is worthwhile to compare results of this paper with that of Coron's ([Cor91]). First, let's summarize several relevant results of this paper.

Consider a nonholonomic system of the form

$$\dot{x} = b_1(x)u_1 + \cdots + b_m(x)u_m, \quad x \in \mathbb{R}^n \quad (58)$$

and a holonomic system of the form

$$\dot{y} = g(y, t), \quad y \in \mathbb{R}^n,$$

Let l be the Lie-bracket order of (58) and fix a ball centered at x_0 and of radius δ , $B(x_0, \delta)$.

1. If $g(y, t) \in C^r$ for any t , then there exists feedback solution $u_\epsilon(x, t, \epsilon) \in C^{r-l}$ for any t and ϵ such that

$$\|x - y\|_{C[0, 2\pi]} \rightarrow 0, \text{ as } \epsilon \rightarrow 0.$$

2. If $g(y, t)$ is 2π -periodic in t and $y(2\pi) \in B(x, \alpha\delta)$, ($0 < \alpha < 1$), for any $y(0) \in B(x, \delta)$, then

$$\|x - y\|_{C[0, \infty)} \rightarrow 0, \text{ as } \epsilon \rightarrow 0.$$

3. If the holonomic system is exponentially stable, i.e.,

$$\|y(t) - x_0\| \leq e^{-\alpha t} \|y(0) - x(0)\|$$

for any $y(0) \in B(x_0, \delta)$, then there exists an attractor A_ϵ such that

$$\text{Dist}(x, A_\epsilon) \leq C e^{-\alpha t}$$

and

$$\max_{z \in A_\epsilon} \|x_0 - z\| \rightarrow 0, \text{ as } \epsilon \rightarrow 0.$$

Recall that, for a first-order Lie bracket system where the nominal system is given by $\dot{y} = -y$, the control input has the form

$$u(x, t, \epsilon) = \alpha(x) + \sqrt{\frac{2}{\epsilon}} \sum_{i=1}^{m(m-1)/2} \beta_i(x) \sin\left(\frac{it}{\epsilon}\right) + \gamma_i(x) \cos\left(\frac{it}{\epsilon}\right)$$

which is unbounded as $\epsilon \rightarrow 0$. To avoid using unbounded control we can rescale time with

$$v_\epsilon = \sqrt{\epsilon} u(x, t\sqrt{\epsilon}, \epsilon)$$

The trajectory of the system under the application of v_ϵ , denoted by $\tilde{x}(t)$, is related to $x(t)$ by

$$\tilde{x}(t) = x(t\sqrt{\epsilon})$$

and the rate of convergence to an attractor of radius $\sim \sqrt{\epsilon}$ is $e^{-t\sqrt{\epsilon}}$.

It is interesting to note that a similar rate of convergence is also observed in the solution of Riccati equation. Consider a discrete-time Riccati equation

$$P(n+1) = P(n) \left(1 - \frac{P(n)}{1+P(n)} \right) = \frac{P(n)}{1+P(n)}, \quad P(0) \geq 0$$

associated with solution of the following system

$$\begin{aligned} x(n+1) &= x(n) + u(n) \\ \min & \sum_{n=1}^{\infty} u(n)^2 \end{aligned}$$

It is clear that the solution $P(n) = \frac{P(0)}{1+nP(0)} \rightarrow 0$ as $n \rightarrow \infty$.

Consider the perturbed equation

$$P(n+1) = \epsilon + \frac{P(n)}{1+P(n)}$$

we have $P(n) \rightarrow \tilde{P} > 0$, where $\tilde{P} = \epsilon + \frac{\tilde{P}}{1+\tilde{P}}$. Thus, the equation for \tilde{P} is

$$\tilde{P}^2 - \epsilon\tilde{P} - \epsilon = 0$$

which gives

$$\tilde{P} = \frac{\epsilon + \sqrt{\epsilon^2 + 4\epsilon}}{2} \sim \sqrt{\epsilon}.$$

Therefore,

$$P(n) - \tilde{P} \sim \left(\frac{1}{1 + \tilde{P}}\right)^{2n} \sim e^{-2\sqrt{\epsilon}n}.$$

We would like to ask the following question: *Can we have exponential stability of the goal point using the smooth time-periodic feedbacks given in our algorithm?* Note that in order for this to be true, the control input has to satisfy

$$u(x, 0, \epsilon) = 0$$

which implies that

$$\alpha(0) = 0, \gamma_i(0) = 0.$$

We will now show that the answer to the above question is negative²

Recall the Gershkovich inequalities ([VG88]): If a system of the form $\dot{x} = B(x)u$, $x \in \mathbb{R}^n$, $u \in \mathbb{R}^m$, is regular in some neighborhood of x_0 , $\eta_1(x_0) = m$, $\eta_i(x_0)$ is the dimension of the space V_i generated by Lie brackets of order less than i . Denote by $h(x_0) = \max_i \{\eta_i > \eta_{i-1}\} = \min\{i, \eta_i = n\}$ and define by $\phi(i) = j$ for $\eta_{j-1} < i \leq \eta_j$ ($\eta_0 = 0$), and some local coordinate system $\mathcal{X}_i : B(x_0, \sigma) \rightarrow \mathbb{R}$ parallelepiped

$$\Pi_{C,\epsilon}(x_0) = \{y \in B(x_0, \sigma) : |\mathcal{X}_i(y) - \mathcal{X}_i(x_0)| \leq C\epsilon^{\phi(i)}\}.$$

Also denote by $D_\epsilon(x_0)$ the set of all x_F such that

$$\dot{x} = B(x)u,$$

$x(0) = x_0$, $x(1) = x_F$ and $\int_0^1 \langle u, u \rangle^{\frac{1}{2}} dt \leq \epsilon$. Then there are local coordinate systems $\{\mathcal{X}_i\}$, positive C , c , ϵ_0 , such that for $\epsilon < \epsilon_0$,

$$\Pi_{c,\epsilon}(x_0) \subset D_\epsilon(x_0) \subset \Pi_{C,\epsilon}(x_0).$$

²This result was first established in [FGL91b].

Notice that $\|x - x_0\|_* \stackrel{\Delta}{=} \|x - x_0\|$ is equivalent to $\sum_{i=1}^n |\mathcal{X}_i(x) - \mathcal{X}_i(x_0)|$ for small enough $\|x - x_0\|$.

We need the following consequences of this fact: There exist $\alpha, \beta, \epsilon''$ such that

1. For any $y \in B(x_0, \epsilon'')$

$$\mathcal{J}(y) \leq \alpha \|y - x_0\|^{\frac{1}{\eta(x_0)}}.$$

2. There exists a sequence $y_k \rightarrow x_0$, such that

$$\mathcal{J}(y_k) \geq \beta \|y_k - x_0\|^{\frac{1}{\eta(x_0)}}.$$

Here $\mathcal{J}(y)$ is the optimal cost for the problem $\dot{x} = B(x)u$, $x(0) = x_0$, $x(1) = y$, $\mathcal{J}(y) = \min_u \int_0^1 \langle u, u \rangle^{\frac{1}{2}} dt$.

Definition 3.1 *The system $\dot{x} = B(x)u$ is $(\sigma, g(t))$ stabilizable at x_0 if and only if there exist positive k and Σ_0 such that for any $y \in B(x_0, \Sigma_0)$ there is a control $\tilde{u}(t), 0 \leq t < \infty$, satisfying the following condition*

$$\dot{x} = B(x)\tilde{u}(t), \quad x(0) = y,$$

$$\|\tilde{u}(t)\| \leq k \|x(t) - x_0\|^\sigma$$

and

$$\|x(t) - x_0\| \leq \|y - x_0\| g(t), \quad g(t) \geq 0.$$

Theorem 3.3 . *The system is not $(C, g(t))$ stabilizable at x_0 if $\int_0^\infty (g(t))^\sigma dt < \infty$ and $\sigma > \frac{1}{\eta(x_0)}$.*

Proof: Let us pick a sequence $y_k \rightarrow x_0$, $\mathcal{J}(y_k) \geq \frac{1}{c} \|y_k - x_0\|^{\frac{1}{\eta(x_0)}}$. The following “scaling symmetry” is well-known: If $\dot{x} = B(x)u$, $x(0) = x_0$, $x(T) = x_f$, $\mathcal{J}(x_0, x_f, T) = \min_u \int_0^T \langle u, u \rangle^{\frac{1}{2}} dt$, then $\mathcal{J}(x_0, x_f, T) \equiv \mathcal{J}(x_0, x_f, 1)$ ($0 < T < \infty$).

Let us construct on the time interval $[0, T + 1]$ the following control:

1. On $[0, T]$, “stabilizing” control $\tilde{u}(t)$, $\dot{x} = B(x)\tilde{u}$, $x(0) = y_k$.
2. On $[T, T + 1]$, optimal control for the problem

$$\begin{aligned}\dot{y} &= B(y)u, \quad y(T) = x(T), \quad y(T + 1) = x_0, \\ &\min \int_T^{T+1} \langle u, u \rangle^{\frac{1}{2}} dt\end{aligned}$$

Then using the Gershkovich inequality and the property of optimality we have

$$\begin{aligned}\beta \|y_k - x_0\|^{\frac{1}{\eta(x_0)}} &\leq \mathcal{J}(y_k) \\ &\leq \int_0^T \|u(t)\| + \mathcal{J}(x(t)) \\ &\leq \beta \int_0^T \|y_k - x_0\|^\sigma g(t)^\sigma + \alpha g(T)^{\frac{1}{\eta(x_0)}} \|y_k - x_0\|^{\frac{1}{\eta(x_0)}}.\end{aligned}$$

Let $\sigma > \frac{1}{\eta(x_0)}$, $\int_0^\infty (g(t))^\sigma dt = \mathcal{D} < \infty$. Then

$$\beta - \alpha g(T) \leq k\mathcal{D} \|y_k - x_0\|^{\sigma - \frac{1}{\eta(x_0)}},$$

for any k and T . Because for some sequence $g(T_m) \rightarrow 0$, we obtain $\beta \leq 0$, a contradiction. \square

Appendix B: Analytic Solutions of Generalized Brockett System

In this appendix, we study analytic solutions for the following generalized Brockett system

$$\begin{aligned}\dot{x} &= u, \\ \dot{y} &= \left(\sum_{i=1}^n A_i u_i \right) x\end{aligned}\tag{59}$$

where $x, u \in \mathbb{R}^n$, $y \in \mathbb{R}^m$, $A_i \in \mathbb{R}^{m \times n}$ for $i = 1, \dots, n$, and boundary conditions are

$$x(2\pi) = x(0) = 0, \quad y(2\pi) - y(0) = \delta = (\delta_1, \dots, \delta_m)^T.$$

If we use the following Fourier representation for the control input

$$u_i = \sum_{p=1}^{\infty} a_p^i \sin pt + \sum_{p=1}^{\infty} b_p^i \cos pt, \quad i = 1, \dots, n\tag{60}$$

then the cost to be minimized has the form

$$J = \int_0^{2\pi} \langle u, u \rangle dt = \sum_{p=1}^{\infty} \sum_{i=1}^n ((a_p^i)^2 + (b_p^i)^2)$$

Integrating the second equation of (59) by parts yields

$$\begin{aligned}\delta_i &= \int_0^{2\pi} \sum_{j=1}^n \sum_{k=1}^n (A_k(i, j) u_k) x_j dt \\ &= \int_0^{2\pi} \sum_{j=1}^n \sum_{k=1}^n A_k(i, j) \dot{x}_k x_j dt \\ &= \sum_{k < j} (A_k(i, j) - A_j(i, k)) \int_0^{2\pi} \dot{x}_k x_j dt \\ &= \pi \sum_{k < j} (A_k(i, j) - A_j(i, k)) \sum_{p=1}^{\infty} \frac{1}{p} (a_p^k b_p^j - a_p^j b_p^k)\end{aligned}$$

Denote by

$$X_p^1 = (a_p^1, \dots, a_p^n)^T, \quad X_p^2 = (b_p^1, \dots, b_p^n)^T$$

and define a linear operator

$$D(i, (k, j)) = A_k(i, j) - A_j(i, k), \quad k < j, \quad (61)$$

from the space of $n \times n$ skew-symmetric matrices, denoted by $so(n)$, to \Re^m . Then, the expression for δ can be written more compactly as

$$\delta = \pi D\left(\sum_{p=1}^{\infty} \frac{1}{p} X_p^1 \wedge X_p^2\right). \quad (62)$$

It is not difficult to see that the necessary and sufficient condition for controllability of system (59) is that the linear operator D has rank m .

In the case that $m = n(n - 1)/2$ we can identify \Re^m with $so(n)$. If D is the identity operator, then system (59) reduces to the Brockett's canonical system

$$\begin{aligned} \dot{x} &= u, \\ \dot{Z} &= x \wedge u, \quad Z \in so(n). \end{aligned}$$

Without loss of generality, we will therefore consider optimal solutions for systems of the form

$$\begin{aligned} \dot{x} &= u, \quad x(0) = x(2\pi) \in \Re^n \\ \dot{y} &= D(x \wedge u), \quad y(2\pi) - y(0) = \delta \in \Re^m \end{aligned} \quad (63)$$

where the control input has the representation given by (60) and the cost to be minimized is

$$\int_0^{2\pi} \langle u, u \rangle dt = \sum_{p=1}^{\infty} \|X_p^1\|^2 + \|X_p^2\|^2$$

In order to describe the set of optimal solutions for the problem, we introduce the following geometric construction. First, denote by

$$J_n = \begin{pmatrix} 0 & I \\ -I & 0 \end{pmatrix}$$

the canonical symplectic matrix of \Re^{2n} , and by $Sp(n)$ the set of representations of the form

$$J_n = \sum_{i=1}^n x_i^1 \wedge x_i^2$$

where $\{x_i^l, 1 \leq i \leq n, 1 \leq l \leq 2\}$ is an orthonormal basis of \Re^{2n} . Then, $Sp(n)$ has the following decomposition

$$Sp(n) = S_{2n} \times S_{2(n-1)} \times \cdots \times S_4 \times S_2 \quad (64)$$

where S_k is the unit sphere of \Re^k . Consider, for example, the case $n = 2$. Let $\{x_i^1, x_i^2\}_{i=1}^2$ be an orthonormal basis of \Re^4 . Then,

$$J_2 = x_1^1 \wedge x_1^2 + x_2^1 \wedge x_2^2$$

and the linear subspace L_i defined by

$$L_i = \text{Span}\{x_i^1, x_i^2\}_{i=1}^2$$

is invariant under J_2 , i.e., $J_2(L_i) \subset L_i$ because

$$J_2(x_i^1) = -x_i^2$$

and

$$J_2(x_i^2) = x_i^1.$$

Using these properties we can construct a decomposition of $Sp(2)$ as follows: First, pick an arbitrary $x_1^2 \in S_4$ and compute

$$x_1^1 = J_2 x_1^2$$

Then, compute an orthogonal complement of $\text{Span}\{x_1^1, x_1^2\}$

$$L_2 = R^4 \ominus \text{Span}\{x_1^1, x_1^2\} \sim R^2.$$

Finally, pick an arbitrary $x_2^2 \in S_2$ and compute

$$x_2^1 = J_2 x_2^2$$

As a result we have that $Sp(2) = S_4 \times S_2$.

Theorem 3.4 Consider system (63). Suppose that there exists a unique $S \in so(n)$ such that $\delta = \pi D(S)^3$

$$S = \alpha_1 J_{n_1} \oplus \alpha_2 J_{n_2} \cdots \oplus \alpha_k J_{n_k} \oplus 0, \quad \alpha_1 \geq \alpha_2 \cdots \geq \alpha_k \geq 0$$

and

$$R^n = Z_{n_1} \oplus Z_{n_2} \cdots \oplus Z_{n_k} \oplus Ker(S)$$

with $Dim(Z_{n_i}) = 2n_i$ and J_{n_i} an operator in Z_{n_i} , then the set of optimal control to system (63) has the form

$$\begin{aligned} u &= \sum_{p=1}^{\text{rank}(S)/2} X_p^1 \sin pt + X_p^2 \cos pt \\ &= \sum_{i=1}^k \sum_{j=1}^{n_i} X_{i,j}^1 \sin(n_1 + \cdots + n_{i-1} + j) + X_{i,j}^2 \cos(n_1 + \cdots + n_{i-1} + j) \end{aligned}$$

where

$$X_{i,j}^l = \sqrt{\alpha_i(n_1 + \cdots + n_{i-1} + j)} Y_{i,j}^l \in Z_{n_i}, \quad l = 1, 2$$

and

$$\{Y_{i,j}^1, Y_{i,j}^2, 1 \leq j \leq n_i\} \in Sp(n_i)$$

and the optimal cost is given by

$$I_{opt} = 2 \sum_{i=1}^k \alpha_i (n_1 + \cdots + n_{i-1} + \frac{1+n_i}{2}) n_i$$

The above theorem is a special case of the following result.

Proposition 3.3 Let $S = \alpha_1 J_{n_1} \oplus \cdots \oplus \alpha_k J_{n_k} \oplus 0$, $\alpha_1 \geq \cdots \geq \alpha_k \geq 0$, and consider the following optimization problem: Find $\{x_i^1, x_i^2\}$ such that the constraint

$$S = \sum_{i=1}^{\infty} x_i^1 \wedge x_i^2$$

is satisfied and the cost function

$$I(x) = \sum_{i=1}^{\infty} C_i (\|x_i^1\|^2 + \|x_i^2\|^2), \quad 0 < C_i < C_{i+1} \rightarrow \infty.$$

³This is equivalent to $\delta \in Im(D)$ and $Ker(D) = 0$.

is minimized.

For any solution of the problem we have that

$$x_n^l = Y_{i,j}^l \sqrt{\alpha_i}$$

where $\{Y_{i,j}^l, 1 \leq l \leq 2, 1 \leq j \leq n_i\} \subset Sp(n_i)$,

$$n_1 + \cdots + n_{i-1} < n < n_1 + \cdots + n_i$$

and the optimal cost is

$$2 \sum_{i=1}^k \alpha_i \sum_{j=n_1 + \cdots + n_{i-1} + 1}^{n_1 + \cdots + n_i} C_j.$$

To prove Proposition 3.3, we need the following lemma.

Lemma 3.2 Let $\{x_1^1, x_1^2, x_2^1, x_2^2\}$ be a solution to the following optimization problem:

$$S = x_1^1 \wedge x_1^2 + x_2^1 \wedge x_2^2$$

and cost function

$$C_1 (\|x_1^1\|^2 + \|x_1^2\|^2) + C_2 (\|x_2^1\|^2 + \|x_2^2\|^2), \quad 0 < C_1 < C_2$$

Then,

$$\|x_i^1\| = \|x_i^2\|, x_i^k \perp x_j^l \text{ for } (i, k) \neq (j, l)$$

Proof of Lemma 3.2. First, let us show that $x_i^1 \perp x_i^2$. Suppose that

$$x_i^2 = \mu x_i^1 + \gamma$$

Then, $\langle \gamma, x_i^1 \rangle = 0$, $\|\gamma\| < \|x_i^2\|$, $x_i^1 \wedge x_i^2 = x_i^1 \wedge \gamma$, and

$$C_1 (\|x_i^1\|^2 + \|\gamma\|^2) < C_1 (\|x_i^1\|^2 + \|x_i^2\|^2)$$

resulting in a contradiction.

Suppose that $\alpha = \|x_i^1\| \neq \|x_i^2\| = \beta$. Then, we can decrease $\|x_i^1\|^2 + \|x_i^2\|^2$ without changing $(x_i^1 \wedge x_i^2)$ by redefining

$$\tilde{x}_i^1 = \sqrt{\alpha\beta}x_i^1, \tilde{x}_i^2 = \sqrt{\alpha\beta}x_i^2$$

It remains to show that $x_1^k \perp x_2^l, k, l \in \{1, 2\}$. Since

$$\text{Rank}(S) \leq \text{Rank}(x_1^1 \wedge x_1^2) + \text{Rank}(x_2^1 \wedge x_2^2) = 4$$

we can restrict the problem to \Re^4 . In this case,

$$S = \alpha_1 J_2 + \alpha_2 J_2, \alpha_1 \geq \alpha_2 \geq 0.$$

There are two possibilities: (i) $\alpha_2 = 0$, then $S = x \wedge y$ and $\|x\|^2 = \|y\|^2 = \alpha_1$. Let us show that $x_1^1 = x, x_1^2 = y$ and $x_2^1 = x_2^2 = 0$. If not, then we have that

$$C_1\|x_1^1\|^2 + C_2\|x_2^1\|^2 > C_1(\|x_1^1\|^2 + \|x_2^1\|^2) \geq \alpha_1 C_1$$

But, if $(x_1^1 = x, x_1^2 = y, x_2^1 = x_2^2 = 0)$, then $S = x_1^1 \wedge x_1^2$ and

$$C_1\|x_1^1\|^2 + C_2\|x_2^1\|^2 = C_1\alpha_1$$

(ii) $\alpha_2 \neq 0$. Let us show that

$$c = \min(\|x_1^1\|^2, \|x_2^1\|^2) \geq \alpha_2.$$

If not, then

$$\| (S - x_1^1 \wedge x_1^2) x \| \geq (\alpha_2 - c) \|x\|$$

and $\text{Rank}(x_2^1 \wedge x_2^2) = 4$, results in a contradiction.

Suppose that

$$L_2^\perp \triangleq (\text{Span}(x_2^1, x_2^2))^\perp \neq \text{Span}(x_1^1, x_1^2) \triangleq L_1$$

we can show that in this case $c > \alpha_2$. If not, for instance $\|x_1^1\| = \alpha_2$, pick $y \in (\text{Span}(x_2^1, x_2^2))^\perp$ and $y \notin \text{Span}(x_1^1, x_1^2)$. Then,

$$S(y) = (x_1^1 \wedge x_1^2)(z), \|z\| \leq \|y\|$$

where z is the projection of y onto L_1 , and $\|S(y)\| < \alpha_2 \|y\|$. But, for any vector $z \in \Re^4$ we have $\|S(z)\| \geq \alpha_2 \|z\|$, a contradiction.

Thus, we have the following system of linear inequalities for $A_1 \triangleq \|x_1^1\|^2$ and $A_2 \triangleq \|x_2^1\|^2$

$$A_1 + A_2 \geq \alpha_1 + \alpha_2; (\alpha_2 \leq \alpha_1) \text{ (because of Claim 2)}$$

and

$$A_1 > \alpha_2, A_2 > \alpha_1, A_1 \geq A_2$$

Consequently, we obtain

$$C_1 A_2 + C_2 A_1 \geq C_1 \alpha_1 + C_2 \alpha_2$$

which contradicts the optimality condition. \square

Claim 1: Let $A_0 = A_1 + A_2$, where A_i 's are $n \times n$ square matrices with singular values $\{\lambda_i^l, 1 \leq i \leq n, 1 \leq l \leq 2\}$. Then,

$$\left(\sum_{i=1}^n (\lambda_i^l)^p \right)^{1/p} \leq \sum_{i=1}^2 \left(\sum_{j=1}^n (\lambda_j^l)^p \right)^{1/p}, \quad p \geq 1.$$

Claim 2: Let $S = \alpha_1 J_{n_1} \oplus \cdots \oplus \alpha_k J_{n_k} \oplus 0$, $S = \sum_n x_n^1 \wedge x_n^2$, $\|x_n^1\| = \|x_n^2\|$, $\langle x_n^1, x_n^2 \rangle = 0$, then

$$\sum_{i=1}^k \alpha_i n_i \leq \sum_n \|x_n^1\|^2.$$

Proof of Proposition 3.3. First, let us prove the existence of optimal solutions. For this consider a sequence of near-optimal solutions $\{x_m\} \triangleq \{x_{m,n}^l, 1 \leq l \leq 2\}$, $1 \leq m \leq \infty$ such that

$$S = \sum_{n=1}^{\infty} x_{m,n}^1 \wedge x_{m,n}^2, \quad \min I(x_n) = \inf_x I(x).$$

Because $C_i \rightarrow \infty$ the set $\{x_m\}$ is compact in the strong topology of l_2 . Thus, there exist a subsequence

$$x_{m_k} \rightarrow \bar{x} = \{x_n^l, 1 \leq n < \infty, 1 \leq l \leq 2\}$$

such that $S = \sum_{n=1}^{\infty} x_n^1 \wedge x_n^2$ and

$$\sum_{n=1}^{\infty} \|x_n^1\|^2 + \|x_n^2\|^2 = \inf_x I(x) = \min_x I(x)$$

Therefore, let $\bar{x} = \{x_n^l, 1 \leq n \leq \infty, 1 \leq l \leq 2\}$ be an optimal solution. From the Lemma we have that

$$x_n^1 = x_n^2, x_n^k \perp x_n^m, \text{ for } (n, l) \neq (k, m),$$

and there exists a map

$$\pi : [1, \sum_{i=1}^k n_i] \rightarrow N$$

such that

$$\|x_n^1\|^2 = \begin{cases} 0, n \notin Im(D) \\ \alpha_k, n = \pi(i), n_1 + \dots + n_{k-1} < i \leq n_1 + \dots + n_k \end{cases}$$

To complete the proof, recall the following inequality: If $\alpha_1 \leq \alpha_2 \dots \leq \alpha_n$, and $b_1 \geq b_2 \dots \geq b_n$, then for any permutation π

$$\sum_{i=1}^n \alpha_i b_{\pi(i)} \geq \sum_{i=1}^n \alpha_i b_i.$$

□

Remark 3.4 Let $S = \alpha_1 J_{n_1} \oplus \dots \oplus \alpha_k J_{n_k} \oplus 0$, and $\alpha_1 \geq \alpha_2 \dots \geq \alpha_k > 0$. Define the following functional on $so(n)$

$$f(S) = \sum_i \alpha_i i$$

If $\text{Ker}(D) \neq 0$, then the optimal control problem for system (63) is equivalent to

$$\min_S \{f(S) | D(S) = y(2\pi) - y(0)\}$$

We will give an analytic solution to this problem for $y \in \mathfrak{R}$:

$$S = \sum_{i=1}^k \alpha_i (x_i^1 \wedge x_i^2), \quad \alpha_1 \geq \alpha_2 \dots \geq \alpha_k \geq 0 \quad (65)$$

and $x_i^k \perp x_j^l$ if $(i, k) \neq (j, l)$, $\|x_i^j\| = 1$,

$$\min \{f(S) = \sum_{i=0}^k \alpha_i C_i\}, \quad 0 < C_1 < \dots < C_k, \quad D(S) = \delta \in \mathfrak{R}.$$

Note that if $D : so(n) \rightarrow \mathfrak{R}$ is a linear functional then there exists unique $\tilde{S} \in SO(n)$ such that

$$D(S) = \text{tr}(S\tilde{S})$$

where $\text{tr}(S)$ stands for the trace of S .

Claim 3: (1) Suppose that

$$S = \sum_{i \geq 1}^k \alpha_i (x_i^1 \wedge x_i^2), \quad \alpha_1 \geq \alpha_2 \cdots \geq \alpha_k \geq 0$$

where $\|x_i^k\| = 1$ and $x_i^k \perp x_j^l$ for $(i, k) \neq (j, l)$, and

$$\tilde{S} = \sum_{i \geq 1}^k \beta_i (z_i^1 \wedge z_i^2), \quad \beta_1 \geq \beta_2 \cdots \geq \beta_k \geq 0$$

where $\|z_i^k\| = 1$ and $z_i^k \perp z_j^l$ for $(i, k) \neq (j, l)$. Then,

$$|\delta| = |\text{tr}(S\tilde{S})| \leq \sum_{i=1}^k \alpha_i \beta_i$$

(2)

$$\sum_{i=1}^k \alpha_i C_i \geq \frac{|\delta|}{\beta_1} C_1.$$

Proof of Claim 3. First, note that

$$|\text{tr}(S\tilde{S})| = \left| \sum_{i=1}^k \sum_{j=1}^k \alpha_i \beta_j \text{tr}((x_i^1 \wedge x_i^2)(z_j^1 \wedge z_j^2)) \right| \leq \sum_{i=1}^k \alpha_i \beta_i \Lambda ij$$

where

$$\begin{aligned} \Lambda ij &= |\text{tr}((x_i^1 \wedge x_i^2)(z_j^1 \wedge z_j^2))| \\ &= |\langle x_i^1, z_j^2 \rangle \langle x_i^2, z_j^1 \rangle - \langle x_i^1, z_j^1 \rangle \langle x_i^2, z_j^2 \rangle| \\ &\leq (\langle x_i^1, z_j^1 \rangle^2 + \langle x_i^1, z_j^2 \rangle^2 + \langle x_i^2, z_j^1 \rangle^2 + \langle x_i^2, z_j^2 \rangle^2) / 2 \end{aligned}$$

Using orthonormality condition, we have

$$\sum_{j=1}^k \Lambda ij \leq \frac{\|x_i^1\|^2 + \|x_i^2\|^2}{2} = 1.$$

$$\sum_{i=1}^k \Lambda_{ij} \leq \frac{\|z_i^1\|^2 + \|z_i^2\|^2}{2} = 1.$$

Therefore, $\{\Lambda_{ij}\}$ is a semibystochastic matrix and

$$\sum_{i=1}^k \sum_{j=1}^k \alpha_i \beta_j \Lambda_{ij} = \sum_{\pi} \gamma_{\pi} \left(\sum_{i=1}^k \alpha_i \beta_{\pi(i)} \right)$$

where π is a permutation, $\gamma_{\pi} \geq 0, \sum_{\pi} \gamma_{\pi} \leq 1$. But, for any permutation π we have that

$$\sum_{i=1}^k \alpha_i \beta_{\pi(i)} \leq \sum_{i=1}^k \alpha_i \beta_i.$$

To show the second part, observe that

$$\begin{aligned} \sum_{i=1}^k \alpha_i C_i &= \sum_{i=1}^k \alpha_i \beta_i \frac{C_i}{\beta_i} \geq \min_i \left(\frac{C_i}{\beta_i} \right) \sum_{i=1}^k \alpha_i \beta_i \\ &= \frac{C_1}{\beta_1} \sum_{i=1}^k \alpha_i \beta_i \geq \frac{C_1}{\beta_1} |\delta| \end{aligned}$$

□

Proposition 3.4 Suppose that

$$\tilde{S} = \beta_1 J_{n_1} \oplus \cdots \oplus \beta_k J_{n_k} \oplus 0, \quad \beta_1 \geq \beta_2 \geq \cdots \geq \beta_k \geq 0 \in so(n)$$

and

$$\mathfrak{R}^n = Z_{n_1} \oplus \cdots \oplus Z_{n_k} \oplus Ker(\tilde{S}).$$

Then, the set of optimal solutions of problem (65) has the form

$$\left\{ \frac{\delta}{\beta_1} (Z \wedge -S(Z)), \|Z\| = 1, Z \in Z_{n_1} \right\}$$

and

$$\min f(S) = \frac{|\delta|}{\beta_1} C_1$$

The proof is a direct application of Claim 3.

Remark 3.5 Proposition 3.3 is correct even if $C_i \not\rightarrow \infty$. Necessary and sufficient condition for the existence of solutions is: There exist $C_{\pi(1)}, \dots, C_{\pi(k)}$ such that

$$0 < C_{\pi(1)} \leq \dots \leq C_{\pi(k)} \leq C_n, n \geq 1$$

where $k = \text{Rank}(S)/2$.

Claim 4: Consider optimal control of the following system

$$\dot{x} = B(x)u, \quad x \in \Re^n, u \in \Re^m$$

with initial condition $x(0) = x_0, x(t) = x_f$ and cost

$$\inf \int_0^t \langle u, u \rangle dt = I_t(x_0, x_f)$$

Then, I_t satisfies the following triangle inequality

$$\sqrt{I_t(x_0, x_f)} \leq \sqrt{I_t(x_0, z)} + \sqrt{I_t(z, x_f)}$$

The proof follows from the time symmetry of the cost functional, i.e., $I_t(x_0, x_f) = \frac{1}{t} I_1(x_0, x_f)$.

Consider now Brockett's system

$$\begin{aligned} \dot{x} &= u \\ \dot{Y} &= \frac{1}{\pi}(x \wedge u), Y \in so(n) \end{aligned}$$

with boundary conditions $x(0) = x(2\pi) = 0$, and $Y(0) = 0, Y(2\pi) = S$, and cost $I = \int_0^2 \langle u, u \rangle dt$. Then, we have that $f(S) = I_{2\pi}((0, 0), (0, S))$, and the following inequalities follow from the claim.

$$\sqrt{f(S_1 + S_2)} \leq \sqrt{f(S_1)} + \sqrt{f(S_2)}, \quad S \in so(n). \quad (66)$$

For a symmetric matrix $A = A^*$, we can also define the analog of the functional f : Let $\sigma(A) = (a_1, a_2, \dots, a_n)$, $|a_1| \geq |a_2| \geq \dots \geq |a_n|$, be the spectrum values of A and define

$$\tilde{f}(A) = \sum |a_i| c_i$$

According to [Fri81], this functional is concave on the convex set of nonnegative matrices.

Corollary 3.1 If $c_i = i$, then the following inequality holds

$$\sqrt{\tilde{f}(A_1 + A_2)} \leq \sqrt{\tilde{f}(A_1)} + \sqrt{\tilde{f}(A_2)}$$

Proof: Associate with A_i the following skew-symmetric matrix

$$S_i = \begin{bmatrix} 0 & A_i \\ -A_i & 0 \end{bmatrix}$$

and apply inequality (66). \square

In what follows we will give a *control system model* of Proposition 3.3. Let

$$K : L_2[0, 1] \rightarrow L_2[0, 1]$$

be a bounded linear operator, and suppose that

$$K - K^* = \sum_{i=1}^{\infty} \frac{1}{c_i} x_i^1 \wedge x_i^2, \quad x_i^k \in L_2[0, 1]$$

where $\|x_i^k\| = 1$, $x_i^k \perp x_j^l$ if $(i, k) \neq (j, l)$ and $c_i \geq \epsilon > 0$. Therefore,

$$(K - K^*)(v) = \sum \frac{1}{c_i} x_i^1 \langle v, x_i^2 \rangle - x_i^2 \langle v, x_i^1 \rangle, \quad v \in L_2[0, 1]$$

If $u = (u_1, \dots, u_m)^T$ is a vector-valued function, we then define

$$K(u) = (K(u_1), \dots, K(u_m))^T$$

Consider nonlinear control problem for the following system

$$\dot{y} = \left(\sum_{i=1}^n A_i u_i \right) K u, \quad y \in \mathbb{R}^m \tag{67}$$

with boundary condition $y(1) - y(0) = \delta$ and cost to be minimized

$$\min \int_0^1 \langle u, u \rangle dt.$$

It is possible to show that an optimal solution of the above system has to satisfy

$$u(t) = \sum_{n=1}^{\infty} X_n^1 x_n^1(t) + X_n^2 x_n^2(t)$$

$$y(1) - y(0) = D \sum_{n=1}^{\infty} \frac{1}{c_n} (X_n^1 \wedge X_n^2),$$

where $D : so(n) \rightarrow \Re^m$ is the same operator as defined in (61). We will call systems of this kind *integral nonholonomic systems*, in contrast to the usual differential nonholonomic systems.

This problem can be solved using previous results, and also note that system (67) is controllable if and only if

$$\max_{y \in \Re^m} \min_{D(S)=y} \text{Rank}(S) \leq \text{Rank}(K - K^*).$$

4

LIE BRACKET EXTENSIONS AND AVERAGING: THE SINGLE-BRACKET CASE

Héctor J. Sussmann* and Wensheng Liu

Department of Mathematics

Rutgers University

New Brunswick, NJ 08903

Phone: (908) 932-5407 FAX: (908) 932-5530

e-mail: sussmann@hamilton.rutgers.edu

wliu@math.rutgers.edu

December 10, 1991

Abstract

We explain a general approximation technique for nonholonomic systems by discussing in detail a special example, chosen so as to illustrate some of the technical aspects of the general construction. The example considered is that of an extension of a two-input system obtained by adding a single bracket of degree five. This bracket is sufficiently complicated to exhibit some phenomena, such as multiplicity, that do not occur for brackets of lower degree.

*This author's work was supported in part by the National Science Foundation under NSF Grant DMS-8902994.

INTRODUCTION

In [19], we introduced a general approximation technique for nonholonomic systems which gives an explicit construction, for any given path γ in state space, of a sequence of admissible trajectories converging to γ , and in addition has a number of other desirable properties. The technique is based on an improved version of a theorem proved by Haynes and Hermes in [5], and uses a result on convergence of trajectories that extends earlier work ([7], [8], [9]) by Kurzweil and Jarnik. (Cf. Remark 2 for a discussion of this work.)

The purpose of this paper is to explain this technique and some of its properties, by discussing in detail a special example, selected so as to illustrate a number of nontrivial issues that arise in the general construction.

We will consider a two-input driftless control-linear system

$$\dot{x} = u_1(t)f_1(x) + u_2(t)f_2(x), \quad (1)$$

together with a “Lie bracket extension” of (1) given by

$$\dot{x} = v_1(t)f_1(x) + v_2(t)f_2(x) + v_3(t)[[f_1, f_2], [f_1, [f_1, f_2]]](x). \quad (2)$$

We will describe an explicit procedure for constructing, for each input $v(t) = (v_1(t), v_2(t), v_3(t))$ for (2), defined on an interval $[a, b]$, a sequence $\{u^j\}$ of inputs for (1) that generate, for each initial condition $x(a) = \bar{x}$, trajectories x^j that converge to the trajectory \hat{x} of (2) generated by v . This is a particular case of the general algorithm presented in [19], which provides a similar approximation for arbitrary Lie bracket extensions of general m -input systems. We have selected this particular example because it is in a sense the simplest one whose structure is sufficiently rich to serve as a good illustration of the general situation. In particular, the bracket used in this extension has *multiplicity two* (cf. below), and is the simplest bracket whose multiplicity is nontrivial, i.e. > 1 . As explained below, for brackets with nontrivial multiplicity some complications arise that would never be seen in the trivial

case, and this is the main reason for our choice of the bracket $[[f_1, f_2], [f_1, [f_1, f_2]]]$.

The general definitions of [19] are as follows. A *Lie bracket extension* of an m -input system

$$\dot{x} = \sum_{k=1}^m u_k(t) f_k(x), \quad (3)$$

where f_1, \dots, f_m are smooth vector fields on a smooth manifold M , is a system

$$\dot{x} = \sum_{k=1}^r v_k(t) f_k(x), \quad (4)$$

where the first m vector fields f_k , $k = 1, \dots, m$, are the same as those in (3), and the new directions f_{m+1}, \dots, f_r are Lie brackets of the f_k , $k \in \{1, \dots, m\}$. A sequence of L^1 inputs $u^j : [a, b] \rightarrow \mathbb{R}^m$ for (3) *converges in the trajectory sense* (or, for short, *T-converges*) to an L^1 input $v : [a, b] \rightarrow \mathbb{R}^r$ for (4) if, for each initial condition $x(a) = p$, the trajectories x^j of (3) generated by the u^j converge uniformly to the trajectory \hat{x} of (4) for v .¹ The algorithm of [19] produces, for each input v for (4), a sequence $\{u^j\}$ of inputs for (3) that T-converges to v . As formulated, the definition of T-convergence depends on the choice of M and of specific vector fields f_1, \dots, f_m , but in fact our construction is “universal” in the sense that the u^j only depend on the formal structure of the Lie brackets used in the extension, but not on the specific choice of M, f_1, \dots, f_m . (This will be given a precise meaning below, by defining the concept of “EI-convergence,” but what it means in our special case is, simply, that for any given input v for (2) we will construct a sequence $\{u^j\}$ that T-converges to v for all M and all possible pairs (f_1, f_2) of sufficiently smooth vector fields on M .)

The situation studied in this paper is a special case of that analyzed in [19]. Here, rather than discuss arbitrary Lie bracket

¹In general, the trajectories of (3) and (4) might fail to be unique, or might have explosions. To take care of this, our precise definition of T-convergence is as follows: given a system (3) with state space M , and a Lie bracket extension (4), the sequence $\{u^j\}$ *T-converges to v* if for every point p , if $\hat{x} : [a, b] \rightarrow M$ is a trajectory of (4) for v that satisfies $\hat{x}(a) = p$, then for sufficiently large j there are trajectories $x^j : [a, b] \rightarrow M$ of (3) for u^j that satisfy $x^j(a) = p$, and $x^j \rightarrow \hat{x}$ uniformly as $j \rightarrow \infty$.

extensions, as was done in [19], we will consider the very simple “single-bracket” extension (2). We hope that a detailed description of the procedure for such a simple case will make the general method clearer. We remark that the extension of the method from the single-bracket case to the general multibracket situation is nontrivial, since it gives rise to new technical issues. In the last section of this paper we briefly outline this generalization and some related technical problems, and we explain how our techniques also make it possible to handle certain systems with drift, and to approximate feedback control laws.

Explicit approximation algorithms are of interest for the following reason. It is clear that an extended system such as (4) gives rise to many more trajectories than the original system (3) does. Therefore, when we look for paths of a special kind (e.g. paths that join two given points and avoid certain obstacles), it will be easier to find them for (4) than for (3). In particular, if the right-hand side of (4) is so rich that direct control in *every* direction is possible, then every curve of class C^1 will be a trajectory of (4). The existence of an extension which is “sufficiently rich” in this sense is guaranteed if (3) satisfies the *Lie algebra rank condition* (LARC)². Precisely, if the LARC holds, and K is any prescribed compact subset of M , then we can choose r large enough such that the set of vectors $\{f_k(x), k = 1, \dots, r\}$ spans $T_x M$ for every $x \in K$. On the other hand, the LARC is a natural condition for this kind of problems, because under fairly general conditions—for instance, real analyticity—it is equivalent to *complete controllability*, i.e. the property that given any two points p, q there is a trajectory from p to q . For a system (3) that satisfies the LARC, every smooth curve $\gamma : [a, b] \rightarrow M$ is a trajectory of a suitably chosen Lie bracket extension (4) for some input $v : [a, b] \rightarrow \mathbb{R}^r$. If we know how to produce a sequence $\{u^j\}$ of inputs for (3) that T-converges to v as $j \rightarrow \infty$, then we get, in particular, a solution of the problem of approximating any given smooth curve in M by trajectories of (3). If there is an “obstacle,” i.e. a closed set C that

²System (3) satisfies the LARC if for every $x \in M$ we have $\Lambda(x) = T_x M$, where Λ is the Lie algebra of vector fields generated by the $f_k, k = 1, \dots, m$, $\Lambda(x) = \{X(x) : X \in \Lambda\}$, and $T_x M$ is the tangent space of M at x .

we wish to avoid, then any solution of the path finding problem with obstacle avoidance for totally unrestricted paths will yield an approximate solution of the similar restricted problem, in which the paths are in addition required to be trajectories of (3). This leads to an alternative approach to the Motion Planning Problem for nonholonomic systems studied, e.g., by Brockett, Sastry, Hauser, Murray, Gurvits, Li, Lafferriere and Sussmann (cf. [2], [3], [4], [6], [10], [11], [12], [13], [14], [15]).

Our strategy for constructing the desired approximation relies on a *T-convergence theorem* that gives sufficient conditions for T-convergence of a sequence of inputs u^j for (3) to an input v for (4). Applying this theorem, we can explicitly compute the limits of certain special sequences. A particular case is that of *highly oscillatory sequences* (HOS's) of inputs, defined below. An HOS of inputs involves several arbitrary functions η_ω , and the limit turns out to be given by an explicit formula in terms of the η_ω . The final step is then to realize that this formula implies that any desired limit v can be achieved by a suitable choice of the η_ω .

Our T-convergence theorem generalizes a classical convergence result. This classical result says that, if $\{u^j\}$ is a sequence of inputs belonging to $L^1([a, b], \mathbb{R}^m)$, and if x^j is the solution of (3) corresponding to u^j , with an initial condition $x^j(a) = p$, then the x^j converge uniformly on $[a, b]$ to the solution x^∞ of

$$\begin{aligned}\dot{x}(t) &= \sum_{k=1}^m u_k^\infty(t) f_k(x(t)) , \\ x(a) &= p ,\end{aligned}\tag{5}$$

provided that

- (a) the vector fields f_k are locally Lipschitz,
- (b) x^∞ exists on the whole interval $[a, b]$,
- (c) the indefinite integrals $U^j(t) = \int_a^t u^j(s) ds$ converge uniformly on $[a, b]$ to $U^\infty(t) = \int_a^t u^\infty(s) ds$,
- (d) the functions u^j are uniformly bounded in L^1 , i.e. there exists a finite constant C such that $\int_a^b \|u^j(t)\| dt \leq C$ for all j .

Remark 1. The most widely known form of the classical convergence theorem is a slightly weaker statement, in which conditions (c) and (d) are replaced by the stronger hypothesis that the u^j converge weakly in L^1 to u^∞ . It is not hard to see that (c) and (d) together are strictly weaker than weak convergence. Indeed, weak convergence is equivalent to uniform convergence of the integrals plus equiintegrability, so it implies (c) and (d), but the converse need not be true, since a sequence satisfying (c) and (d) need not be equiintegrable. ■

The above classical convergence theorem can be restated as a “first-order” T-convergence result: if (a), (c) and (d) hold then the u^j T-converge to the *ordinary* input u^∞ , so that the limiting equation contains no new Lie brackets. Our generalization is a high-order T-convergence theorem in which limiting equations involving brackets are obtained. It is clear that, for any such generalization, Condition (c) will have to be kept. Indeed, if the u^j T-converge to v for all possible choices of sufficiently smooth f_1, \dots, f_m , then they have to T-converge to v in the special case when f_1, \dots, f_m are linearly independent constant vector fields on \mathbb{R}^m , and this suffices to imply that (c) holds if we let $u^\infty = (v_1, \dots, v_m)$. So our extension will have to be based on dropping the boundedness condition (d). It is not hard to see that, if (d) fails, then the x^j may fail to converge to x^∞ , but it is possible for them to converge to a trajectory of a Lie bracket extension (4), which could fail to be a trajectory of (3). Moreover, this can happen in the “universal” way described above, so that in fact the u^j T-converge to some v for all possible choices of sufficiently smooth f_1, \dots, f_m .

For a simple illustration of this phenomenon, consider first the system

$$\dot{x}_1 = u_1, \quad \dot{x}_2 = u_2, \quad \dot{x}_3 = u_2 x_1, \quad (6)$$

on an interval $[0, T]$, with initial condition $(0, 0, 0)$. If we let

$$u^j(t) = \sqrt{j}(\cos jt, \sin jt),$$

then the u^j “converge to 0” in the sense that their indefinite integrals $U^j(t) = \int_0^t u^j(s) ds$ converge to 0 uniformly. So, if we let

$u^\infty \equiv (0, 0)$, we might think that the u^j T-converge to u^∞ , so that the $x^j(t)$ converge uniformly to $x^\infty(t) \equiv (0, 0, 0)$. However, a simple calculation shows that this is not so, and in fact $x^j(t)$ converges uniformly to $\hat{x}(t) = (0, 0, \frac{t}{2})$. On the other hand, the vector fields f_1, f_2 have components $(1, 0, 0)$ and $(0, 1, x_1)$, respectively. So the Lie bracket $f_3 = [f_1, f_2]$ has components $(0, 0, 1)$, and the limiting function \hat{x} turns out to satisfy $\dot{\hat{x}} = \frac{1}{2}f_3(x)$. Moreover, one can verify by an integration by parts that for *every* two-input system (1) with C^2 vector fields f_1, f_2 , and every initial condition $x(0)$, the trajectories x^j that correspond to the u^j converge to the solution of $\dot{x} = \frac{1}{2}[f_1, f_2](x)$, provided that the latter exists on $[0, T]$. In other words, the ordinary \mathbb{R}^2 -valued inputs u^j for (1) T-converge to the \mathbb{R}^3 -valued input $v = (0, 0, \frac{1}{2})$ for $\dot{x} = v_1 f_1(x) + v_2 f_2(x) + v_3 [f_1, f_2](x)$ for all choices of C^2 vector fields f_1, f_2 . We can restate this in vector-field-free language by introducing *formal indeterminates* X_1, X_2 , to be thought of as “slots” where vector fields can be plugged in. Our example then simply says that the ordinary inputs $u^j(t) = \sqrt{j} \cos jt X_1 + \sqrt{j} \sin jt X_2$ T-converge to the “extended input” $\frac{1}{2}[X_1, X_2]$ for all choices of C^2 vector fields f_1, f_2 . We rise to an even higher level of abstraction and completely eliminate the vector fields, by just saying that the $u^j(t) = \sqrt{j} \cos jt X_1 + \sqrt{j} \sin jt X_2$ EI-converge (that is, “converge in the extended input sense”) to $\frac{1}{2}[X_1, X_2]$.

Our general convergence theorem gives sufficient conditions for EI-convergence of a sequence of ordinary inputs u^j to an extended input v . An application of this theorem will give us a sufficient condition for inputs $u^j = u_1^j X_1 + u_2^j X_2$ to EI-converge on an interval $[a, b]$ to an extended input

$$\mathbf{v} = v_1 X_1 + v_2 X_2 + v_3 [[X_1, X_2], [X_1, [X_1, X_2]]],$$

where v_1, v_2, v_3 are C^1 functions on $[a, b]$. Using this, we will explicitly compute the limit of certain HOS’s involving twelve unknown functions, and then it will be obvious how to choose these functions so as to match any given \mathbf{v} of the above form, for any C^1 functions v_1, v_2, v_3 .

Remark 2. Haynes and Hermes proved, in [5], that, if a driftless system (3) with state space M satisfies the Lie algebra rank

condition, then every continuous curve in M can be uniformly approximated by trajectories of (3). Kurzweil and Jarnik found ([7], [8], [9]) more restrictive versions of our general convergence result (Theorem 2). The idea of making systematic use of Lie bracket extensions was introduced in Lafferriere-Sussmann [11]. ■

SOME ALGEBRAIC PRELIMINARIES

To clarify the presentation, it will be convenient to introduce some algebraic machinery. We follow the notations and definitions of [17]. For simplicity, we restrict ourselves to the two-input case.

We let $\mathbf{X} = \{X_1, X_2\}$ be a sequence of two objects, that will be called *indeterminates*. We use $A(\mathbf{X})$, $L(\mathbf{X})$ to denote, respectively, the free associative algebra in X_1, X_2 and the Lie subalgebra of $A(\mathbf{X})$ generated by X_1, X_2 . (Then $L(\mathbf{X})$ can be identified with the free Lie algebra in X_1, X_2 .) In simpler terms, the elements of $A(\mathbf{X})$ are *formal noncommutative polynomials* in X_1, X_2 , i.e. finite linear combinations such as $3X_1 - 7X_2 + 4X_1^2 + 8X_1X_2 + 5X_2X_1$ of *formal monomials* (cf. below) in X_1, X_2 . The elements of $L(\mathbf{X})$ are the *Lie polynomials* in X_1, X_2 , i.e. the $P \in A(\mathbf{X})$ that can be written as linear combinations of Lie brackets of X_1 and X_2 . (For example, X_1X_2 is not in $L(\mathbf{X})$, but $X_1X_2 - X_2X_1$, which by definition is equal to $[X_1, X_2]$, is in $L(\mathbf{X})$. Other examples of members of $L(\mathbf{X})$: $3X_1 - 2X_2 - [X_1, X_2], [[X_1, X_2], [X_1, [X_1, X_2]]]$.)

We also let $\hat{A}(\mathbf{X})$, $\hat{L}(\mathbf{X})$ be, respectively, the algebra of *formal noncommutative power series* and the Lie algebra of *Lie series* in X_1, X_2 . The elements of $\hat{A}(\mathbf{X})$ are the *noncommutative power series* in X_1, X_2 , i.e. arbitrary, not necessarily finite, linear combinations of monomials, and the elements of $\hat{L}(\mathbf{X})$ are those power series that are actually *Lie series*, i.e. possibly infinite linear combinations of Lie brackets.

A *monomial* in X_1, X_2 is a formal product such as $X_1X_2X_1$, or $X_2X_1X_1X_1X_2X_1$. (As usual, we use power notation for repeated products, so the latter monomial is written more simply

as $X_2X_1^4X_2X_1$.) Monomials are multiplied by concatenation so that, for instance, if $M_1 = X_1X_2X_1$ and $M_2 = X_2X_1^4X_2X_1$, then $M_1M_2 = X_1X_2X_1X_2X_1^4X_2X_1$. (Notice that $M_1M_2 \neq M_2M_1$.) Multiplication of monomials is associative, i.e. $(MN)P = M(NP)$ for any monomials M, N, P . More precisely, monomials can be labeled by multiindices $I = (\ell_1, \dots, \ell_k)$ whose entries ℓ_s are integers equal to 1 or 2. We write

$$X_I \stackrel{\text{def}}{=} X_{\ell_1}X_{\ell_2}\cdots X_{\ell_k}. \quad (7)$$

Use $|I|$ to denote the length of I , i.e. $|I| = k$ if $I = (\ell_1, \dots, \ell_k)$. Then it is clear that, as I ranges over all possible multiindices I of lengths $|I| \geq 0$, the X_I form a basis of $A(\mathbf{X})$. (The only index of length 0 is the empty index \emptyset , and then $X_\emptyset \stackrel{\text{def}}{=} 1$.) So every element P of $A(\mathbf{X})$ has a unique expression as a finite linear combination $\sum_I P_I X_I$ of the X_I . In both $A(\mathbf{X})$ and $\hat{A}(\mathbf{X})$, addition is done componentwise, and multiplication is carried out using the formula $X_I X_J = X_{IJ}$, where IJ is the concatenation of I and J , i.e. the multiindex obtained by writing, in order, first the components of I and then those of J .

The Lie Algebra $L(\mathbf{X})$ is spanned by the *brackets* of X_1 and X_2 . Precisely we define $\mathcal{Br}(\mathbf{X})$ to be the smallest subset of $L(\mathbf{X})$ that contains X_1, X_2 and is closed under bracketing. The elements of $\mathcal{Br}(\mathbf{X})$ will be referred to as *brackets* of \mathbf{X} .

Remark 3. Later on we will also have to consider the set $\mathcal{FBr}(\mathbf{X})$ of *formal brackets*, which are certain formal expressions in the symbols X_1, X_2 , the comma and the left and right square brackets. (Precisely, $\mathcal{FBr}(\mathbf{X})$ is the smallest set S of such expressions that contains X_1 and X_2 , and is such that, whenever P and Q are in S , it follows that $[P, Q]$ is in S .) The elements of $\mathcal{FBr}(\mathbf{X})$ are *purely formal expressions*, whereas those of $\mathcal{Br}(\mathbf{X})$ are elements of $L(\mathbf{X})$. To see the difference, notice that, for instance, $[X_1, [X_1, X_2]]$ and $[[X_2, X_1], X_1]$ are the same element of $\mathcal{Br}(\mathbf{X})$ —as can be seen by two applications of the skew-symmetry law $[P, Q] = -[Q, P]$ — but are different as elements of $\mathcal{FBr}(\mathbf{X})$. Formal brackets have a *unique factorization property*: every $B \in \mathcal{FBr}(\mathbf{X})$ except X_1 and X_2 can be written in a unique way as a bracket $[B_1, B_2]$, with

$B_1, B_2 \in \mathcal{FBr}(\mathbf{X})$. (The brackets B_1, B_2 are, respectively, the *left and right factors* of B .) This fails for “real brackets.” For example, if $B = [X_1, [X_1, X_2]]$ then B has the unique factorization $B = [B_1, B_2]$ in $\mathcal{FBr}(\mathbf{X})$, with $B_1 = X_1$ and $B_2 = [X_1, X_2]$, but the factorization in $\mathcal{Br}(\mathbf{X})$ is not unique, because we can also write $B = [\tilde{B}_1, \tilde{B}_2]$ with $\tilde{B}_1 = [X_2, X_1]$, $\tilde{B}_2 = X_1$. ■

For each bracket $B \in \mathcal{Br}(\mathbf{X})$ we use $\delta(B)$ to denote the total degree of B , and let $\delta_k(B)$, $k = 1, 2$ be the degree of B with respect to X_k , so $\delta_1(B) + \delta_2(B) = \delta(B)$. For example if $B = [X_1, [X_1, X_2]]$, then $\delta_1(B) = 2$, $\delta_2(B) = 1$, and $\delta(B) = 3$.

An *extended input* on an interval $[a, b]$ is an $\hat{L}(\mathbf{X})$ -valued, integrable function on $[a, b]$. (If \mathbf{v} is any $\hat{A}(\mathbf{X})$ -valued function on $[a, b]$, we say that \mathbf{v} is *integrable* if all the coefficient functions v_I in the expression $\mathbf{v}(t) = \sum_I v_I(t)X_I$ are integrable on $[a, b]$. This makes sense in particular if \mathbf{v} is $\hat{L}(\mathbf{X})$ -valued.) An ordinary input $u = (u_1, u_2)$ can be regarded as an extended input by identifying it with the function $\mathbf{u}(t) = u_1(t)X_1 + u_2(t)X_2$. As explained above, an extended input such as $v_1(t)X_1 + v_2(t)X_2 + v_3(t)[X_1, X_2]$ should be thought of as an object that contains “slots” (the indeterminates X_1, X_2) where vector fields can be plugged in, giving rise to an ordinary differential equation. For example, the extended input $v_1(t)X_1 + v_2(t)X_2 + v_3(t)[X_1, X_2]$ gives rise, for each choice of sufficiently smooth vector fields f_1, f_2 , to the differential equation

$$\dot{x} = v_1(t)f_1(x) + v_2(t)f_2(x) + v_3(t)[f_1, f_2](x) .$$

We say that an extended input \mathbf{v} is *of order* $\leq k$ if it only involves brackets of degree $\leq k$. (If \mathbf{v} is of order $\leq k$ for some integer k then we say that \mathbf{v} is *of finite order*. If k is the smallest integer for which this happens, then we say that \mathbf{v} is of order k .) An extended input v of order $\leq k$ can be “plugged” into any system (1), if the vector fields f_1, f_2 are of class C^κ with $\kappa \geq k - 1$. The result is a time-varying differential equation whose right-hand side is of class $C^{\kappa+1-k}$ with respect to the state variable x . (In particular, if $\kappa \geq k$ then the right-hand side is locally Lipschitz in x and satisfies the conditions of the existence and uniqueness theorem of Carathéodory.) Since, for each choice of sufficiently

smooth vector fields, extended inputs of finite order give rise to trajectories for each initial condition, the concept of *T-convergence* of such inputs is well defined.

Let \mathbf{v} be an extended input of finite order k , and let r be an integer such that $r \geq k - 1$. A sequence $\{u^j\}$ of ordinary inputs will be said to *EI-converge to \mathbf{v} for systems of class C^r* if it T-converges to \mathbf{v} for all choices of vector fields f_1, f_2 of class C^r . (The requirement that $r \geq k - 1$ is needed to make sure that the vector fields can be plugged into \mathbf{v} .)

With this terminology, we can reformulate the approximation problem as follows:

(AP) Given an extended input \mathbf{v} of order $\leq r$, find a sequence of ordinary inputs that EI-converges to \mathbf{v} for systems of class C^r .

As indicated in the introduction, the basic idea for solving (AP) is to use *highly oscillatory sequences* (HOS's) of inputs. An HOS is a sequence $\{u^j\}$ whose components u_1^j, u_2^j are finite linear combinations of expressions such as $j^\alpha \eta_\omega(t) \cos(j\omega t)$ and $j^\alpha \eta_\omega(t) \sin(j\omega t)$. The HOS's used in our general approximation algorithm involve a finite set of frequencies ω chosen so as to satisfy some special resonance conditions. In the general treatment given in [19], where \mathbf{v} is an arbitrary extended input of finite order, we proceed by trying to handle each bracket in \mathbf{v} separately. We split \mathbf{v} into a sum $\sum_{\nu_1, \nu_2} \mathbf{v}_{\nu_1, \nu_2}$ of “totally homogeneous components” (cf. below) and then try to produce, for each $\mathbf{v}_{\nu_1, \nu_2}$, an HOS $\{u_{\nu_1, \nu_2}^j\}$ that EI-converges to $\mathbf{v}_{\nu_1, \nu_2}$. We then let $u^j = \sum_{\nu_1, \nu_2} u_{\nu_1, \nu_2}^j$, and hope that this will work. It turns out that, even though the input-to-trajectory map is highly nonlinear, a kind of “high frequency superposition principle” holds, and the u^j EI-converge to \mathbf{v} , provided that the frequencies associated to the various components $\mathbf{v}_{\nu_1, \nu_2}$ are independent in a sense that will be made precise below.

The precise definition of the $\mathbf{v}_{\nu_1, \nu_2}$ is as follows. Let $L(\mathbf{X})_{\nu_1, \nu_2}$ denote the linear span of all the brackets $B \in Br(\mathbf{X})$ such that

$\delta_1(B) = \nu_1$ and $\delta_2(B) = \nu_2$. Then $L(\mathbf{X})$ is the direct sum of the spaces $L(\mathbf{X})_{\nu_1, \nu_2}$, so every element $w \in L(\mathbf{X})$ has a unique decomposition as a sum $\sum_{\nu_1, \nu_2} w_{\nu_1, \nu_2}$, $w_{\nu_1, \nu_2} \in L(\mathbf{X})_{\nu_1, \nu_2}$. Similarly, an extended input $\mathbf{v} : [a, b] \rightarrow L(\mathbf{X})$ has a unique decomposition as a sum $\sum_{\nu_1, \nu_2} v_{\nu_1, \nu_2}$, where each v_{ν_1, ν_2} is $L(\mathbf{X})_{\nu_1, \nu_2}$ -valued. These v_{ν_1, ν_2} are the *totally homogeneous components* of \mathbf{v} . If $B \in \mathcal{B}r(\mathbf{X})$, $\delta_1(B) = \nu_1$ and $\delta_2(B) = \nu_2$, then the *multiplicity* of B is the dimension of $L(\mathbf{X})_{\nu_1, \nu_2}$.

For the special case of the Lie bracket extension (2) to be discussed in this paper, the right-hand side of the extension only involves one bracket. Let us write $B_0 = [[X_1, X_2], [X_1, [X_1, X_2]]]$. Our goal is to exhibit, for every triple of functions v_1, v_2, v_3 of class C^1 on some interval $[a, b]$, a sequence of ordinary inputs that EI-converges to the extended input $\bar{\mathbf{v}} \stackrel{\text{def}}{=} v_1(t)X_1 + v_2(t)X_2 + v_3(t)B_0$. The bracket B_0 has degrees $\delta_1(B_0) = 3$, $\delta_2(B_0) = 2$. It is easy to verify that the component $L(\mathbf{X})_{3,2}$ is *two-dimensional*, with a basis given by the two brackets B_0 and B_1 , where $B_1 = [X_2, [X_1, [X_1, [X_1, X_2]]]]$. So B_0 has multiplicity two.

Unfortunately, the “*high frequency superposition principle*” mentioned above does not make it possible to separate linearly independent brackets in the same totally homogeneous component of $L(\mathbf{X})$. This creates an additional technical complication, that is resolved in [19] as follows. We declare two brackets *equivalent* if they belong to the same totally homogeneous component. Then, for each component $E = L(\mathbf{X})_{\nu_1, \nu_2}$ we use an HOS $\{u^j\}$ with a number of frequencies equal to $\mu(\nu_1 + \nu_2)$, where $\mu = \dim E$, and the frequencies are grouped into μ groups Ω^k of $\nu_1 + \nu_2$ frequencies each, of which ν_1 are allocated to u_1 and ν_2 to u_2 . These frequencies are required to satisfy certain resonance and nonresonance conditions. If P_1, \dots, P_μ is a suitably chosen basis of E (a P. Hall basis, to be precise, cf. below), then it turns out that we can produce a sequence $\{u^{E,j}\}$ that EI-converges to a limit $\sum_\ell \psi_\ell P_\ell$, where (i) each ψ_ℓ is in turn a sum $\sum_k \xi_k g_{P_\ell}(\Omega^k)$, (ii) the ξ_k are certain functions that can be chosen arbitrarily, and (iii) each $\Omega \rightarrow g_{P_\ell}(\Omega)$ is a rational function of $(\nu_1 + \nu_2)$ -tuples Ω of frequencies. One then needs the algebraic fact that the matrix $(g_{P_\ell}(\Omega^k))_{1 \leq k, \ell \leq \mu}$ can be chosen

to be invertible. Using this, one can clearly find ξ_k such that the limit of the $u^{E,j}$ is any desired E -valued extended input v . (The general proof of this algebraic fact seems to be quite complicated, but in our special case we will be able to verify it easily.)

THE APPROXIMATING INPUTS

We now turn to the details, and give a precise description of the approximating sequence $\{u^j\}$.

Without loss of generality, we will fix an interval $[0, T]$ and assume that all the inputs, ordinary or extended, are defined on $[0, T]$. We follow the prescription of [19]. Since $\delta_1(B_0) = 3$, $\delta_2(B_0) = 2$, and the multiplicity of B_0 is two, we take two 5-tuples

$$\bar{\Omega} = (w_1, w_2, w_3, w_4, w_5), \quad \bar{\Omega}' = (w'_1, w'_2, w'_3, w'_4, w'_5), \quad (8)$$

each one consisting of five different frequencies, three of which (the w_k and w'_k for $k = 1, 2, 3$) are allocated to u_1 , while the other two are assigned to u_2 . We require:

(ch1) $\bar{\Omega}$ and $\bar{\Omega}'$ are minimally canceling.

(ch2) the 5-tuples $\bar{\Omega}$ and $\bar{\Omega}'$ are independent.

(An m -tuple $\Omega = (\omega_1, \dots, \omega_m)$ of real numbers is called *minimally canceling* (MC) if, whenever $\{a_k\}_{k=1}^m$ is a collection of integers such that $\sum_{k=1}^m |a_k| \leq m$, then it follows that $\sum_{k=1}^m a_k \omega_k = 0$ if and only if all the a_k are all equal. Two m -tuples $\Omega = (\omega_1, \dots, \omega_m)$, $\Omega' = (\omega'_1, \dots, \omega'_m)$ are *independent* if, whenever $\{a_k\}_{k=1}^m$ and $\{a'_k\}_{k=1}^m$ are two collections of integers such that $\sum_{k=1}^m (|a_k| + |a'_k|) \leq m$ and $\sum_{k=1}^m (a_k \omega_k + a'_k \omega'_k) = 0$, it follows that $\sum_{k=1}^m a_k \omega_k = \sum_{k=1}^m a'_k \omega'_k = 0$.)

In other words, Condition **(ch1)** says that the five frequencies in $\bar{\Omega}$ and those in $\bar{\Omega}'$ “resonate” (in the sense that their sum is

zero) but no resonance occurs with other combinations of five or fewer frequencies, even with repetitions. (So that, for instance, $w_1 + w_2 + w_3 + 2w_4 \neq 0$, $2w_1 + w_2 - w_4 \neq 0$, $2w_2 - 3w_5 \neq 0$, $w_2 + w_3 \neq 0$, etc.) Condition (ch2) says that no resonance can be produced by picking five frequencies in a “mixed” way, i.e. some from $\bar{\Omega}$ and some from $\bar{\Omega}'$.

From now on, the ten numbers w_k, w'_k will be fixed. The letter ω , and symbols such as ω_1, ω_2 , will be used to denote *variables* ranging over frequencies.

We now let

$$\Omega(1) = \{\pm w_1, \pm w_2, \pm w_3, \pm w'_1, \pm w'_2, \pm w'_3\}, \quad (9)$$

$$\Omega(2) = \{\pm w_4, \pm w_5, \pm w'_4, \pm w'_5\}, \quad (10)$$

$$\Omega = \Omega(1) \cup \Omega(2), \quad (11)$$

and think of the 12 frequencies in $\Omega(1)$ as being assigned to u_1 , and the 8 frequencies in $\Omega(2)$ as allocated to u_2 .

We then take the approximating control sequence to be

$$u_1^j(t) = \eta_{1,0}(t) + j^{\frac{4}{5}} \sum_{\omega \in \Omega(1)} \eta_\omega(t) e^{ij\omega t}, \quad (12)$$

$$u_2^j(t) = \eta_{2,0}(t) + j^{\frac{4}{5}} \sum_{\omega \in \Omega(2)} \eta_\omega(t) e^{ij\omega t}, \quad (13)$$

where

(ch3) the functions $\eta_{k,0}, \eta_\omega$ are of class C^1 on $[0, T]$, $\eta_{k,0}$ are real valued, η_ω are complex valued, and $\eta_{-\omega} = \bar{\eta}_\omega$.

(As usual, $\bar{\cdot}$ stands for “complex conjugate,” and $i = \sqrt{-1}$. Notice that the various terms of (12) and (13) are in principle complex, but Condition (ch3) implies that the u_k^j are real. The exponent $\frac{4}{5}$ is chosen because our bracket B_0 has degree five. In general, for brackets of degree ν the exponent would be $\frac{\nu-1}{\nu}$.)

We then have:

Theorem 1 Let $u^j(t) = (u_1^j(t), u_2^j(t))$ be the control sequence defined in (12) and (13) with $\bar{\Omega}$, $\bar{\Omega}'$, $\Omega(1)$, $\Omega(2)$, the w 's and the η 's satisfying (8), (9), (10), (ch1), (ch2), (ch3). Then $\{u^j\}$ EI-converges to

$$\begin{aligned} \mathbf{u}^\infty(t) &= \eta_{1,0}(t)X_1 + \eta_{2,0}(t)X_2 \\ &\quad + (g_{B_0}(\bar{\Omega})\xi(t) + g_{B_0}(\bar{\Omega}')\tilde{\xi}(t))B_0 \\ &\quad + (g_{B_1}(\bar{\Omega})\xi(t) + g_{B_1}(\bar{\Omega}')\tilde{\xi}(t))B_1, \end{aligned} \quad (14)$$

where

$$\xi(t) = \eta_{w_1}(t) \dots \eta_{w_5}(t) + \overline{\eta_{w_1}(t) \dots \eta_{w_5}(t)}, \quad (15)$$

$$\tilde{\xi}(t) = \eta_{w'_1}(t) \dots \eta_{w'_5}(t) + \overline{\eta_{w'_1}(t) \dots \eta_{w'_5}(t)}, \quad (16)$$

and the functions g_{B_0} , g_{B_1} are defined by

$$\begin{aligned} g_{B_0}(\omega_1, \dots, \omega_5) &= \frac{1}{\omega_1 \omega_2 \omega_3} \left[\frac{1}{\omega_1 + \omega_4} + \frac{1}{\omega_1 + \omega_5} + \frac{1}{\omega_2 + \omega_4} \right. \\ &\quad \left. + \frac{1}{\omega_2 + \omega_5} + \frac{1}{\omega_3 + \omega_4} + \frac{1}{\omega_3 + \omega_5} \right], \\ g_{B_1}(\omega_1, \dots, \omega_5) &= \frac{1}{\omega_1 \omega_2 \omega_3} \left[\frac{1}{\omega_4} + \frac{1}{\omega_5} \right]. \blacksquare \end{aligned}$$

It is now easy to see how to choose the $\eta_{k,0}, \eta_\omega$ so that the limiting input \mathbf{u}^∞ has a desired value $\bar{\mathbf{v}} = v_1X_1 + v_2X_2 + v_3B_0$. It suffices to take $\eta_{1,0} = v_1$, $\eta_{2,0} = v_2$, $\eta_{w_\ell}(t) = \eta_{w'_\ell}(t) = 1$ for $\ell = 1, 2, 3, 4$, and $\eta_{w_5}(t) = \frac{\zeta(t)}{2}$, $\eta_{w'_5}(t) = \frac{\tilde{\zeta}(t)}{2}$, where the functions $\zeta, \tilde{\zeta}$ satisfy the system of linear equations

$$g_{B_0}(\bar{\Omega})\zeta(t) + g_{B_0}(\bar{\Omega}')\tilde{\zeta}(t) = v_3(t), \quad (17)$$

$$g_{B_1}(\bar{\Omega})\zeta(t) + g_{B_1}(\bar{\Omega}')\tilde{\zeta}(t) = 0. \quad (18)$$

To be sure that the solution of (17) and (18) exists for arbitrary v_3 , we need the extra condition

$$\det \begin{pmatrix} g_{B_0}(\bar{\Omega}) & g_{B_0}(\bar{\Omega}') \\ g_{B_1}(\bar{\Omega}) & g_{B_1}(\bar{\Omega}') \end{pmatrix} \neq 0. \quad (19)$$

So we need to choose $\bar{\Omega}$ and $\bar{\Omega}'$ such that (19) holds in addition to (ch1) and (ch2). It is easy to see that this is always possible.

Indeed, if we let H be the hyperplane in \mathbb{R}^5 consisting of those 5-tuples $\omega = (\omega_1, \dots, \omega_5)$ such that $\omega_1 + \dots + \omega_5 = 0$, then g_{B_0} and g_{B_1} are rational functions on H , and are well defined and analytic on a relatively open and dense subset of H . If we take $\omega = (1, 1, 1, 1, 1)$, then $g_{B_0}(\omega) = 3$ and $g_{B_1}(\omega) = 2$. If we choose instead $\omega = (1, 1, 1, 2, 2)$, then $g_{B_0}(\omega) = 2$ and $g_{B_1}(\omega) = 1$. So (19) holds for at least one pair of 5-tuples of frequencies in H , and hence for all pairs in a relatively open dense subset of $H \times H$. Conditions (ch1) and (ch2) also hold on a relatively open dense subset of $H \times H$, so it is possible to satisfy them simultaneously with (19).

So we impose the extra requirement that

(ch4) $\bar{\Omega}$ and $\bar{\Omega}'$ satisfy (19).

Then for this choice of the frequencies $\bar{\Omega}$ and $\bar{\Omega}'$ we can always choose $\eta_{k,0}, \eta_\omega$ as above to make the limiting input \mathbf{u}^∞ in (14) be identical to $\bar{\mathbf{v}}$. This completes the description of the approximating HOS $\{u^j\}$. In the following sections we will show that the u^j EI-converge to \mathbf{u}^∞ .

THE CONVERGENCE THEOREM

Our first step is to state and prove a general convergence theorem, giving conditions under which a sequence $\{u^j\}$ of ordinary inputs EI-converges to an extended input \mathbf{v} of finite order.

Given any sequence $\{u^j\}$ in $L^1([0, T], \mathbb{R}^2)$, let x^j be the solution of the initial value problem

$$\dot{x}(t) = u_1^j(t)f_1(x) + u_2^j(t)f_2(x) , \quad x(0) = x_0 , \quad (20)$$

where f_1, f_2 are smooth vector fields. Rewrite (20) as an integral equation

$$x^j(t) = x_0 + \sum_{\ell=1}^2 \int_0^t u_\ell^j(s)f_\ell(x^j(s)) ds .$$

Assume that we can “remove the convergent part” from $\int_0^t u_\ell^j(s)ds$ by finding a function $v_\ell \in L^1[0, T]$ with the property that the integral $\int_0^t (v_\ell(s) - u_\ell^j(s))ds$ converges to 0 uniformly as $j \rightarrow \infty$. Let $\tilde{u}_\ell^j = v_\ell - u_\ell^j$, $\tilde{U}_\ell^j(t) = \int_0^t \tilde{u}_\ell^j(s)ds$. Integration by parts gives

$$\begin{aligned} x^j(t) &= x_0 + \sum_{\ell=1}^2 \int_0^t u_\ell^j(s) f_\ell(x^j(s)) ds \\ &= x_0 + \sum_{\ell=1}^2 \int_0^t v_\ell(s) f_\ell(x^j(s)) ds - \sum_{\ell=1}^2 \int_0^t \tilde{u}_\ell^j(s) f_\ell(x^j(s)) ds \\ &= x_0 + \sum_{\ell=1}^2 \int_0^t v_\ell(s) f_\ell(x^j(s)) ds - \sum_{\ell=1}^2 \tilde{U}_\ell^j(t) f_\ell(x^j(t)) \\ &\quad + \sum_{\ell_1, \ell_2=1}^2 \int_0^t u_{\ell_1}^j(s) \tilde{U}_{\ell_2}^j(s) (Df_{\ell_2}) f_{\ell_1}(x^j(s)) ds, \end{aligned}$$

where $Df(x)$ denotes the Jacobian matrix of the vector field f , evaluated at x . If the “residual functions” \tilde{u}_ℓ^j are uniformly bounded in L^1 then, since the \tilde{U}_ℓ^j converge to zero uniformly, it follows that the x^j converge to a solution of

$$x(t) = x_0 + \sum_{\ell=1}^2 \int_0^t v_\ell(s) f_\ell(x(s)) ds,$$

i.e. of

$$\dot{x}(t) = \sum_{\ell=1}^2 v_\ell(t) f_\ell(x(t)),$$

with initial condition $x(0) = x_0$. Equivalently, $\{u^j\}$ EI-converges to

$$\mathbf{v} = \sum_{\ell=1}^2 v_\ell(t) X_\ell.$$

Now let us no longer assume that the \tilde{u}_ℓ^j are uniformly bounded in L^1 , but suppose that we can “remove the convergent part” from each function $u_{\ell_1}^j \tilde{U}_{\ell_2}^j$, by choosing $v_{\ell_1, \ell_2} \in L^1[0, T]$ such that

$$\tilde{U}_{\ell_1, \ell_2}^j(t) = \int_0^t \tilde{u}_{\ell_1, \ell_2}^j(s) ds \stackrel{\text{def}}{=} \int_0^t (v_{\ell_1, \ell_2}(s) - u_{\ell_1}^j(s) \tilde{U}_{\ell_2}^j(s)) ds \rightarrow 0$$

uniformly. (If the \tilde{u}_ℓ^j were uniformly bounded in L^1 , then it is easy to see that v_{ℓ_1, ℓ_2} have to be $\equiv 0$.)

We then get

$$\begin{aligned}
 x^j(t) &= x_0 + \sum_{\ell=1}^2 \int_0^t v_\ell(s) f_\ell(x^j(s)) ds - \sum_{\ell=1}^2 \tilde{U}_\ell^j(t) f_\ell(x^j(t)) \\
 &\quad + \sum_{\ell_1, \ell_2=1}^2 \int_0^t u_{\ell_1}^j(s) \tilde{U}_{\ell_2}^j(s) (Df_{\ell_2}) f_{\ell_1}(x^j(s)) ds \\
 &= x_0 + \sum_{\ell=1}^2 \int_0^t v_\ell(s) f_\ell(x^j(s)) ds - \sum_{\ell=1}^2 \tilde{U}_\ell^j(t) f_\ell(x^j(t)) \\
 &\quad + \sum_{\ell_1, \ell_2=1}^2 \int_0^t v_{\ell_1, \ell_2}(s) (Df_{\ell_2}) f_{\ell_1}(x^j(s)) ds - B
 \end{aligned}$$

where the “bad part” B is given by

$$B = \sum_{\ell_1, \ell_2=1}^2 \int_0^t \tilde{u}_{\ell_1, \ell_2}^j(s) (Df_{\ell_2}) f_{\ell_1}(x^j(s)) ds. \quad (21)$$

If the functions $\tilde{u}_{\ell_1, \ell_2}^j$ are uniformly bounded in L^1 , then it can be shown that the x^j will converge to a solution of

$$\begin{aligned}
 x(t) &= x_0 + \sum_{\ell=1}^2 \int_0^t v_\ell(s) f_\ell(x(s)) ds \\
 &\quad + \sum_{\ell_1, \ell_2=1}^2 \int_0^t v_{\ell_1, \ell_2}(s) (Df_{\ell_2}) f_{\ell_1}(x(s)) ds,
 \end{aligned}$$

i.e. of

$$\dot{x}(t) = \sum_{\ell=1}^2 v_\ell(t) f_\ell(x(t)) + \sum_{\ell_1, \ell_2=1}^2 v_{\ell_1, \ell_2}(t) (Df_{\ell_2}) f_{\ell_1}(x(t)). \quad (22)$$

(This is not completely obvious at first sight, but is a particular case of the convergence theorem to be proved later, so we will not give a separate proof at this point.)

Remark 4. The right-hand side of (22) is not manifestly a vector field, but one can verify by integration by parts that $v_{\ell_1, \ell_2} = -v_{\ell_2, \ell_1}$. Hence, if we use $[f, g] = (Dg)f - (Df)g$, we can rewrite (22) as

$$\dot{x}(t) = \sum_{\ell=1}^2 v_\ell(t) f_\ell(x(t)) + v_{1,2}(t) [f_1, f_2](x(t)). \quad (23)$$

Equivalently, $\{u^j\}$ EI-converges to

$$\mathbf{v}(t) = v_1(t)X_1 + v_2(t)X_2 + v_{1,2}(t)[X_1, X_2]. \quad (24)$$

Our convergence theorem will generalize this formula to the higher-order case. ■

If the $\tilde{u}_{\ell_1, \ell_2}^j$ are not uniformly bounded in L^1 , then we may rewrite B using integration by parts, as

$$\begin{aligned} B &= \sum_{\ell_1, \ell_2=1}^2 \tilde{U}_{\ell_1, \ell_2}^j(t)(Df_{\ell_2})f_{\ell_1}(x^j(t)) \\ &\quad - \sum_{\ell_1, \ell_2, \ell_3=1}^2 \int_0^t u_{\ell_1}(s)\tilde{U}_{\ell_2, \ell_3}^j(s)D((Df_{\ell_3})f_{\ell_2})f_{\ell_1}(x^j(s))ds. \end{aligned}$$

We might then once again be able to “remove the convergent part” from the $u_{\ell_1}^j \tilde{U}_{\ell_2, \ell_3}^j$ and repeat the above process one more time. To make a systematic study of the general process just described, we introduce the new concept of “extended difference.”

Let $\mathbf{u}(t) = u_1(t)X_1 + u_2(t)X_2$, $\mathbf{v}(t) = \sum_{|I|>0} v_I(t)X_I$ be, respectively, an ordinary input and an integrable $\hat{A}(\mathbf{X})$ -valued function. (Notice that \mathbf{v} need not be an extended input, because it is only assumed to be $\hat{A}(\mathbf{X})$ -valued, and may fail to take values in $\hat{L}(\mathbf{X})$.) We define an *integrated extended difference (IED)* of \mathbf{u} and \mathbf{v} to be an $\hat{A}(\mathbf{X})$ -valued function $\widetilde{\mathbf{UV}}(t) = \sum_{|I|>0} \widetilde{UV}_I(t)X_I$, with absolutely continuous coefficients, that satisfies the equations

$$\widetilde{uv}_\ell(t) = v_\ell(t) - u_\ell(t), \quad (25)$$

$$\widetilde{uv}_{\ell_1, \dots, \ell_k}(t) = v_{\ell_1, \dots, \ell_k}(t) - u_{\ell_1}(t)\widetilde{UV}_{\ell_2, \dots, \ell_k}(t), \quad (26)$$

where the \widetilde{uv}_I^j are the derivatives of the $\widetilde{\mathbf{UV}}_I^j$.

Remark 5. The words “extended difference” are natural because of the following algebraic characterization: an $\hat{A}(\mathbf{X})$ -valued function W with absolutely continuous coefficients and no zeroth-order term is an IED of \mathbf{u} and \mathbf{v} if and only if W satisfies the differential equation

$$\dot{W}(t) = -\mathbf{u}(t)W(t) + \mathbf{v}(t) - \mathbf{u}(t). \quad (27)$$

If $W(0) = 0$, then it is clear that W has some of the properties of a “difference of \mathbf{u} and \mathbf{v} .” (For instance, $W \equiv 0$ iff $\mathbf{u} \equiv \mathbf{v}$.)

In general, if $W(0)$ is “small” in some sense, then it is reasonable to expect that $W(t)$ is “small” for all t if and only if \mathbf{u} and \mathbf{v} are “close.” The convergence theorem will make this precise. ■

Now let $\{\mathbf{u}^j\}$ be a sequence of ordinary inputs and $\{\mathbf{v}^j\}$ a sequence of integrable $\hat{A}(\mathbf{X})$ -valued functions, given by $\mathbf{v}^j(t) = \sum_{0 < |I| \leq r} v_I^j(t) X_I$. Write $\widetilde{\mathbf{U}\mathbf{V}}^j$ rather than $\widetilde{\mathbf{U}^j\mathbf{V}^j}$ to denote an IED $\sum_I \widetilde{\mathbf{U}\mathbf{V}}_I^j X_I$ of \mathbf{u}^j and \mathbf{v}^j , and $\widetilde{\mathbf{u}\mathbf{v}}^j$ for the time derivative of $\widetilde{\mathbf{U}\mathbf{V}}^j$. (The IED $\widetilde{\mathbf{U}\mathbf{V}}$ is not unique, so the notation $\widetilde{\mathbf{U}\mathbf{V}}$ is slightly ambiguous, but we will use it nevertheless. Both $\widetilde{\mathbf{U}\mathbf{V}}^j$ and $\widetilde{\mathbf{u}\mathbf{v}}^j$ are uniquely determined by the choice of $\widetilde{\mathbf{U}\mathbf{V}}^j(0)$.)

Assume that f_1, f_2 are vector fields of class C^r for some positive integer r . Then by repeated integrations by parts we see that the solutions x^j of (20) satisfy the integral equation

$$\begin{aligned} x^j(t) = x_0 &+ \sum_{|I| \leq r} \int_0^t v_I^j(s) f_I(x^j(s)) ds - \sum_{|I| < r} [\widetilde{\mathbf{U}\mathbf{V}}_I^j(\cdot) f_I(x^j(\cdot))]_0^t \\ &- \sum_{|I|=r} \int_0^t \widetilde{\mathbf{u}\mathbf{v}}_I^j(s) f_I(x^j(s)) ds, \end{aligned} \quad (28)$$

where we use the notation $[\psi(\cdot)]_a^b$ for $\psi(b) - \psi(a)$ and, for a multiindex $I = (\ell_1, \dots, \ell_k)$, $f_I(x)$ denotes the vector

$$f_I(x) \stackrel{\text{def}}{=} D(D \cdots D((Df_{\ell_k})f_{\ell_{k-1}}) \dots, f_{\ell_2})f_{\ell_1}(x). \quad (29)$$

For I as above, let us also define

$$[X_I] \stackrel{\text{def}}{=} [X_{\ell_1}, [X_{\ell_2}, \dots, [X_{\ell_{k-1}}, X_{\ell_k}] \dots]], \quad (30)$$

$$[f_I] \stackrel{\text{def}}{=} [f_{\ell_1}, [f_{\ell_2}, \dots, [f_{\ell_{k-1}}, f_{\ell_k}] \dots]]. \quad (31)$$

With these notations, we can now state our convergence theorem.

Theorem 2 *Let r be a positive integer. Let $\{\mathbf{u}^j\} \subset L^1([0, T], \mathbb{R}^2)$ be a sequence of ordinary inputs. Let \mathbf{v} be an integrable $\hat{A}(\mathbf{X})$ -valued function on $[0, T]$, given by*

$$\mathbf{v}(t) = \sum_{0 < |I| \leq r} v_I(t) X_I.$$

Let $V_I(t) = \int_0^t v_I(s)ds$. Assume that there exists a sequence $\{\mathbf{V}^j\}$ of $\hat{A}(\mathbf{X})$ -valued functions on $[0, T]$ with absolutely continuous components, given by $\mathbf{V}^j(t) = \sum_{0 < |I| \leq r} V_I^j(t) X_I$, and a sequence $\{\widetilde{UV}^j\}$ of IED's of \mathbf{u}^j and \mathbf{v}^j , such that

A1. $\{V_I^j\}_{j=1}^\infty$ converges uniformly to V_I for $0 < |I| \leq r$;

A2. $\{\widetilde{UV}_I^j\}_{j=1}^\infty$ converges uniformly to 0 for $0 < |I| \leq r$;

A3. the L^1 norms of the derivatives v_I^j for $0 < |I| \leq r$ and \widetilde{uv}_I^j for $|I| = r$ are uniformly bounded.

Then:

C1. \mathbf{v} is an extended input of order $\leq r$;

C2. \mathbf{v} is given by the formula

$$\mathbf{v} = \sum_{0 < |I| \leq r} \frac{v_I(t)}{|I|} [X_I] ; \quad (32)$$

C3. the \mathbf{u}^j EI-converge to \mathbf{v} for C^r systems of the form (1).

Remark 6. In simpler terms, the conclusion of Theorem 2 says that, for every choice of C^r vector fields f_1 , f_2 , and every initial condition x_0 , if the unique solution $x^\infty(t)$ of the initial value problem

$$\dot{x}(t) = \sum_{|I| \leq r} \frac{v_I(t)}{|I|} [f_I](x) , \quad x(0) = x_0 \quad (33)$$

is defined on the whole interval $[0, T]$, and x^j is the solution of (20), then the x^j are defined on the whole interval $[0, T]$ for j large enough, and converge to x^∞ uniformly as $j \rightarrow \infty$. Notice that the initial value problem (33) satisfies the conditions of the Carathéodory theorem on existence and uniqueness of solutions of

ordinary differential equations, because the right-hand side is of class C^1 in x for each fixed t , measurable in t for each x , and is bounded, together with its first partial derivatives with respect to x , by an integrable function of t , as long as x stays in a compact set. ■

PROOF OF THEOREM 2. We will first establish the property stated in Remark 6 for the initial value problem

$$\dot{x}(t) = \sum_{|I| \leq r} v_I(t) f_I(x) , \quad x(0) = x_0 , \quad (34)$$

where the f_I were defined in (29). We will then show that the right-hand side of the first equation of (34) is actually equal to that of the first equation of (33).

Assume first that f_1 and f_2 are compactly supported, so all the trajectories x^j and x^∞ exist on the whole interval $[0, T]$ and are contained in a fixed compact set K . If we could prove that the sequence $\{x^j\}$ is equicontinuous, then our conclusion would follow from (28), using the boundedness of the L^1 norms of the v_I^j and \widetilde{v}_I^j , and the fact that the $\widetilde{U}\widetilde{V}_I^j$ converge uniformly to zero. (By equicontinuity, every subsequence $\{x^{j(\ell)}\}$ of $\{x^j\}$ has a subsequence $\{x^{j(\ell(k))}\} \stackrel{\text{def}}{\equiv} \{\sigma^k\}$ that converges uniformly to a limit σ . Then the integrals $\int_0^t \tilde{v}_I^k(s) f_I(\sigma^k(s)) ds - \int_0^t \tilde{v}_I^k(s) f_I(\sigma(s)) ds$, where $\tilde{v}_I^k = v_I^{j(\ell(k))}$, converge uniformly to zero, because the L^1 norms of the \tilde{v}_I^k are bounded. The integrals $\int_0^t \tilde{v}_I^k(s) f_I(\sigma(s)) ds$ converge to $\int_0^t v_I(s) f_I(\sigma(s)) ds$ because the continuous function $s \rightarrow f_I(\sigma(s))$ can be uniformly approximated by piecewise constant functions, and Assumption A1 implies that $\int_0^t \tilde{v}_I^k(s) \psi(s) ds \rightarrow \int_0^t v_I(s) \psi(s) ds$ if ψ is piecewise constant. So $\int_0^t \tilde{v}_I^k(s) f_I(\sigma^k(s)) ds \rightarrow \int_0^t v_I(s) f_I(\sigma(s)) ds$. A similar argument shows that

$$\int_0^t \widetilde{v}_I^{j(\ell(k))}(s) f_I(\sigma^k(s)) ds \rightarrow 0$$

So $\sigma = x^\infty$ by uniqueness. In particular σ is independent of the subsequence chosen.)

To prove that $\{x^j\}$ is equicontinuous, we let

$$\xi^j(t) = x_0 + \sum_{|I| \leq r} \int_0^t v_I^j(s) f_I(x^j(s)) ds - \sum_{|I|=r} \int_0^t \tilde{w}v_I^j(s) f_I(x^j(s)) ds .$$

Since the \widetilde{UV}_I^j converge uniformly to zero, we have $\xi^j - x^j \rightarrow 0$ uniformly, so equicontinuity of $\{x^j\}$ will follow if we prove that $\{\xi^j\}$ is equicontinuous. Let

$$\zeta^j(t) = x_0 + \sum_{|I| \leq r} \int_0^t v_I^j(s) f_I(\xi^j(s)) ds - \sum_{|I|=r} \int_0^t \tilde{w}v_I^j(s) f_I(\xi^j(s)) ds .$$

Since $f_I(\xi^j) - f_I(x^j) \rightarrow 0$ uniformly, the boundedness of the L^1 norms of the v_I^j and $\tilde{w}v_I^j$ implies that $\zeta^j - \xi^j \rightarrow 0$ uniformly, so it suffices to prove equicontinuity of $\{\zeta^j\}$. We can write

$$\begin{aligned} & \int_0^t v_I^j(s) f_I(\xi^j(s)) ds = \\ & \int_0^t (v_I^j(s) - v_I(s)) f_I(\xi^j(s)) ds + \int_0^t v_I(s) f_I(\xi^j(s)) ds . \end{aligned}$$

The second integral in the right-hand side has an integrand which is the product of a fixed L^1 function times a uniformly bounded sequence, so it is equicontinuous. The first integral can be rewritten via integration by parts as

$$\left[(V_I^j(\cdot) - V_I(\cdot)) f_I(\xi^j(\cdot)) \right]_0^t - \int_0^t (V_I^j(s) - V_I(s)) Df_I(\xi^j(s)) \dot{\xi}^j(s) ds .$$

The definition of the ξ^j as integrals shows that the $\dot{\xi}^j$ are uniformly bounded in L^1 . Since $V_I^j - V_I \rightarrow 0$ uniformly, and the functions $s \rightarrow Df_I(\xi^j(s))$ are uniformly bounded, we conclude that $\int_0^t (V_I^j(s) - V_I(s)) Df_I(\xi^j(s)) \dot{\xi}^j(s) ds \rightarrow 0$ uniformly. Also, $\left[(V_I^j(\cdot) - V_I(\cdot)) f_I(\xi^j(\cdot)) \right]_0^t \rightarrow 0$ uniformly, because $V_I^j(t) - V_I(t) \rightarrow 0$ uniformly and the $f_I(\xi^j(t))$ are uniformly bounded. So the integral $\int_0^t (v_I^j(s) - v_I(s)) f_I(\xi^j(s)) ds$ goes to 0 uniformly. Therefore $\{\int_0^t v_I^j(s) f_I(\xi^j(s)) ds\}$ is equicontinuous. A similar argument shows that the sequence of integrals $\{\int_0^t \tilde{w}v_I^j(s) f_I(\xi^j(s)) ds\}$ is equicontinuous for $|I| = r$. So $\{\zeta^j\}$ is equicontinuous, and the equicontinuity

of $\{x^j\}$ follows. This completes the proof for the case when f_1 and f_2 are compactly supported.

In the general case, we multiply f_1 and f_2 by a smooth compactly supported function $\varphi(x)$ which is equal to one on a neighborhood U of the set $\{x^\infty(t) : 0 \leq t \leq T\}$. We apply the conclusion to the resulting system, and get trajectories \hat{x}^j of this new system that converge to x^∞ . Then for large j the \hat{x}^j will lie in U , so they will be trajectories of the original system. Our conclusion now follows in full generality, except that we have obtained the limiting initial value problem (34) rather than (33).

To complete our proof, we have to establish the formula

$$\sum_{|I| \leq r} v_I(t) f_I(x) = \sum_{|I| \leq r} \frac{v_I(t)}{|I|} [f_I](x) . \quad (35)$$

This can be done as follows. First observe that the definition of the f_I given in (29) implies the identity

$$f_{\ell_1, \dots, \ell_k} = L_{f_{\ell_1}} L_{f_{\ell_2}} \dots L_{f_{\ell_k}} \Phi .$$

Here $\Phi : \mathbb{R}^n \rightarrow \mathbb{R}^n$ is the identity map (i.e. $\Phi(x) = x$) and, for a vector field $g : \mathbb{R}^n \rightarrow \mathbb{R}^n$, L_g is the operator of “Lie differentiation in the direction of g ,” defined by $L_g \psi(x) = D\psi(x).g(x)$, if ψ is a (scalar- or vector-valued) function on \mathbb{R}^n . Next observe that the convergence theorem holds for all sufficiently smooth systems, so in particular it holds for systems in which the vector fields are linear, i.e. $f_k(x) = A_k(x)$, where $A_k : \mathbb{R}^n \rightarrow \mathbb{R}^n$ is some linear map. In this case, one can easily check that all the vector functions f_I are also linear, and given by $f_{\ell_1} \dots f_{\ell_k}(x) = A_{\ell_k} \dots A_{\ell_1}(x)$. A particular example of this is the r -th order nilpotent formal system

$$\dot{S} = S(u_1 X_1 + u_2 X_2) , \quad S(0) = 1 , \quad (36)$$

which evolves in the algebra $A_r(\mathbf{X})$ obtained by “truncating” $A(\mathbf{X})$ at degree r , i.e. declaring all monomials of degree $> r$ to be equal to zero. In this case, the linear map A_k is right multiplication by X_k , so $f_{\ell_1} \dots f_{\ell_k}(S) = S X_{\ell_1} \dots X_{\ell_k}$. The limiting trajectory S^∞ will satisfy $\dot{S}^\infty(t) = S^\infty(t)(\sum_{|I| \leq r} v_I(t) X_I)$, i.e. $\dot{S}^\infty(t) =$

$S^\infty(t)\mathbf{v}(t)$. On the other hand, the series $S^\infty(t)$ is an exponential of a Lie series because of the Campbell-Hausdorff Formula (CHF)³. But then we can write $\mathbf{v}(t) = \lim_{h \rightarrow 0} \frac{1}{h} (S^\infty(t))^{-1} (S^\infty(t+h) - S^\infty(t)) = \lim_{h \rightarrow 0} \frac{1}{h} \left((S^\infty(t))^{-1} S^\infty(t+h) - 1 \right)$. Using CHF again, we conclude that $(S^\infty(t))^{-1} S^\infty(t+h) = e^{\Lambda(h)}$, where $\Lambda(h)$ is a Lie series that goes to zero as $h \rightarrow 0$. So $\mathbf{v}(t) = \lim_{h \rightarrow 0} \frac{1}{h} (\Lambda(h) + \frac{1}{2}\Lambda(h)^2 + \dots) = \lim_{h \rightarrow 0} \frac{\Lambda(h)}{h}$ for almost every t . So \mathbf{v} is Lie-series-valued, i.e. an extended input. We then use a general fact about free Lie algebras, according to which, if a noncommutative series $\sum_I a_I X_I$ is known to be a Lie series, then $\sum_I a_I X_I = \sum_I \frac{a_I}{|I|} [X_I]$, where $[X_I]$ is as defined in (30). In particular,

$$\sum_{|I| \leq r} v_I(t) X_I = \sum_{|I| \leq r} \frac{v_I(t)}{|I|} [X_I].$$

Since this identity is true for purely formal indeterminates X_ℓ , it must remain true if we plug for the X_ℓ any objects in an associative algebra. In particular, we can plug in the operators L_{f_ℓ} , and use repeatedly the fact that $[L_f, L_g] = L_{[f, g]}$, to conclude that $\sum_{|I| \leq r} v_I(t) (L_f)_I = \sum_{|I| \leq r} \frac{v_I(t)}{|I|} L_{[f_I]}$. If we then apply both sides to the map Φ , we get (35). ■

Remark 7. Formula (35) contains as a particular case the observation made in Remark 4. To see this, notice that, according to (35), the expression $\sum_{\ell_1, \ell_2=1}^2 v_{\ell_1, \ell_2} (Df_{\ell_2}) f_{\ell_1}$, which by definition is equal to $\sum_{\ell_1, \ell_2=1}^2 v_{\ell_1, \ell_2} f_{\ell_1, \ell_2}$, has the value $\frac{1}{2}(v_{1,1}[f_1, f_1] + v_{1,2}[f_1, f_2] + v_{2,1}[f_2, f_1] + v_{2,2}[f_2, f_2])$. Since $[f_1, f_1] = [f_2, f_2] = 0$, $[f_1, f_2] = -[f_2, f_1]$, and $v_{1,2} = -v_{2,1}$, we see that this is in fact equal to $v_{1,2}[f_1, f_2]$, as shown in Remark 4. ■

Remark 8. If the vector fields f_1, f_2 are only assumed to be of

³The *exponential* of a series is defined by its usual power series expression. Since we are working in a finite dimensional algebra, there are no convergence problems. The CHF says that a product of exponentials of Lie series is also an exponential of a Lie series. Applying this to a piecewise constant control (u_1, u_2) we conclude that for every trajectory $S(t)$ for such a control, starting with $S(0) = 1$, the series $S(t)$ are exponential Lie series. Taking limits, this remains true for trajectories of arbitrary ordinary controls, and for limits of such trajectories, such as S^∞ .

class C^{r-1} , the conclusion still follows, if one assumes uniqueness of solutions for (33). The proof is almost exactly the same, except that an extra step is required in which the vector-valued functions f_I are approximated by functions g_I of class C^∞ . ■

PROOF OF THEOREM 1

We will apply Theorem 2. For this purpose, we try to define sequences $\{V_I^j\}_{j=1}^\infty$, $\{\widetilde{UV}_I^j\}_{j=1}^\infty$ satisfying the conditions of that theorem. As before, the derivatives of these functions will be denoted by v_I^j , \widetilde{uv}_I^j .

For any multiindex $I = (\ell_1, \dots, \ell_k)$, let $\delta_1(I)$ and $\delta_2(I)$ denote, respectively, the number of occurrences of the indices 1 and 2 in I . Then $|I| = \delta_1(I) + \delta_2(I)$. We use induction to define V_I^j , \widetilde{UV}_I^j for $|I| \leq 5$ such that

A. The V_I^j have the form $V_I^j = V_I + R_I^j$, where

- (i) V_1 and V_2 are given by $V_\ell(t) = \int_0^t \eta_{\ell,0}(s) ds$.
- (ii) If $2 \leq |I| \leq 5$, then $V_I = 0$, unless $\delta_1(I) = 3$ and $\delta_2(I) = 2$.
- (iii) If I has length 5 and $\delta_1(I) = 3$, $\delta_2(I) = 2$,

$$V_I(t) = \int_0^t (\Theta_I \xi(s) + \tilde{\Theta}_I \tilde{\xi}(s)) ds \quad (37)$$

where $\xi, \tilde{\xi}$ are the functions defined in (15) and (16), and the constants $\Theta_I, \tilde{\Theta}_I$ will be evaluated below.

- (iv) The R_I^j converge to 0 uniformly as $j \rightarrow \infty$, and the $\|\dot{R}_I^j\|_{L^1}$ are uniformly bounded for all $|I| \leq 5$.

B. For $0 < |I| \leq 4$, the \widetilde{UV}_I^j can be written as

$$\widetilde{UV}_I^j(t) = (-1)^{|I|} j^{-\frac{|I|}{2}} \sum_{\hat{\omega} \in \Omega(I)} \frac{\eta_{\hat{\omega}}(t) e^{ij(\sum \hat{\omega})t}}{i^{|I|} h(\hat{\omega})}, \quad (38)$$

where, for $I = (\ell_1, \dots, \ell_k)$, if $\hat{\omega} = (\omega_1, \dots, \omega_k)$ belongs to $\Omega(\ell_1) \times \dots \times \Omega(\ell_k)$, we let

$$h(\hat{\omega}) = \omega_k(\omega_k + \omega_{k-1}) \dots (\omega_k + \dots + \omega_1)$$

$$\eta_{\hat{\omega}}(t) = \eta_{\omega_1}(t) \cdots \eta_{\omega_k}(t),$$

$$\sum \hat{\omega} = \omega_1 + \dots + \omega_k$$

$$\Omega(I) = \{\hat{\omega} \in \Omega(\ell_1) \times \dots \times \Omega(\ell_k) : h(\hat{\omega}) \neq 0\}.$$

C. $\widetilde{UV}_I^j = 0$ for $|I| = 5$.

Notice that **A**, **B** and **C** imply that Assumptions **A1**, **A2** and **A3** of Theorem 2 hold.

We now define $V_I^j, \widetilde{UV}_I^j$. Integration by parts gives

$$\begin{aligned} \int_0^t u_\ell^j(s) ds &= \int_0^t \eta_{\ell,0}(s) ds + \int_0^t \sum_{\omega \in \Omega(\ell)} j^{\frac{1}{5}} \eta_\omega(s) e^{ij\omega s} ds \\ &= \int_0^t \eta_{\ell,0}(s) ds + j^{-\frac{1}{5}} \sum_{\omega \in \Omega(\ell)} \left\{ \frac{\eta_\omega(t) e^{ij\omega t}}{i\omega} - \frac{\eta_\omega(0)}{i\omega} - \int_0^t \frac{\eta'_\omega(s) e^{ij\omega s}}{i\omega} ds \right\}. \end{aligned}$$

We let

$$\begin{aligned} V_\ell(t) &= \int_0^t \eta_{\ell,0}(s) ds, \\ R_\ell^j(t) &= -j^{-\frac{1}{5}} \sum_{\omega \in \Omega(\ell)} \left\{ \frac{\eta_\omega(0)}{i\omega} + \int_0^t \frac{\eta'_\omega(s)}{i\omega} e^{ij\omega s} ds \right\}, \\ V_\ell^j(t) &= V_\ell(t) + R_\ell^j(t). \end{aligned}$$

Then

$$\widetilde{UV}_\ell^j(t) = V_\ell^j(t) - \int_0^t u_\ell^j(s) ds = -j^{-\frac{1}{5}} \sum_{\omega \in \Omega(\ell)} \frac{\eta_\omega(t)}{i\omega} e^{ij\omega t}.$$

If we multiply $\widetilde{UV}_{\ell_2}^j(s)$ by $u_{\ell_1}^j(s)$ and integrate, we get

$$\int_0^t u_{\ell_1}^j(s) \widetilde{UV}_{\ell_2}^j(s) ds = A_{\ell_1, \ell_2}^j + B_{\ell_1, \ell_2}^j,$$

where

$$A_{\ell_1, \ell_2}^j = -j^{-\frac{1}{5}} \int_0^t \eta_{\ell_1, 0}(s) \sum_{\omega \in \Omega(\ell_2)} \frac{\eta_\omega(s)}{i\omega} e^{ij\omega s} ds$$

$$B_{\ell_1, \ell_2}^j = -j^{\frac{3}{5}} \sum_{(\omega_1, \omega_2) \in \Omega(\ell_1) \times \Omega(\ell_2)} \int_0^t \frac{\eta_{\omega_1}(s)\eta_{\omega_2}(s)}{i\omega_2} e^{ij(\omega_1 + \omega_2)s} ds.$$

We can split B_{ℓ_1, ℓ_2}^j further into the sum C_{ℓ_1, ℓ_2}^j of those terms for which $\omega_1 + \omega_2 = 0$ and the sum D_{ℓ_1, ℓ_2}^j of those terms for which $\omega_1 + \omega_2 \neq 0$. Then

$$C_{\ell_1, \ell_2}^j = -j^{\frac{3}{5}} \sum_{\substack{(\omega_1, \omega_2) \in \Omega(\ell_1) \times \Omega(\ell_2) \\ \omega_1 + \omega_2 = 0}} \int_0^t \frac{\eta_{\omega_1}(s)\eta_{\omega_2}(s)}{i\omega_2} ds.$$

Since $\eta_{\omega_2} = \overline{\eta_{\omega_1}}$ whenever $\omega_1 + \omega_2 = 0$, we see that each unordered pair $\{\omega_1, \omega_2\}$ such that $\omega_1 + \omega_2 = 0$ makes a contribution to the sum equal to $(\frac{1}{i\omega_1} + \frac{1}{i\omega_2})|\eta_{\omega_1}|^2$, i.e. to zero. Therefore $C_{\ell_1, \ell_2}^j = 0$. So we can rewrite B_{ℓ_1, ℓ_2}^j as a sum of terms $B_{\ell_1, \ell_2}^j(\omega_1, \omega_2)$, ranging over $\omega_1 + \omega_2 \neq 0$. We then use integration by parts to rewrite each of the $B_{\ell_1, \ell_2}^j(\omega_1, \omega_2)$ as a sum of three terms, namely,

$$B_{\ell_1, \ell_2}^j(\omega_1, \omega_2)_1 = -j^{-\frac{2}{5}} \frac{\eta_{\omega_1}(t)\eta_{\omega_2}(t)e^{ij(\omega_1 + \omega_2)t}}{i^2\omega_2(\omega_2 + \omega_1)},$$

$$B_{\ell_1, \ell_2}^j(\omega_1, \omega_2)_2 = j^{-\frac{2}{5}} \frac{\eta_{\omega_1}(0)\eta_{\omega_2}(0)}{i^2\omega_2(\omega_2 + \omega_1)},$$

$$B_{\ell_1, \ell_2}^j(\omega_1, \omega_2)_3 = j^{-\frac{2}{5}} \int_0^t \frac{(\eta_{\omega_1}(s)\eta_{\omega_2}(s))'}{i^2\omega_2(\omega_2 + \omega_1)} e^{ij(\omega_1 + \omega_2)s} ds.$$

It is clear that all the A_{ℓ_1, ℓ_2}^j , $B_{\ell_1, \ell_2}^j(\omega_1, \omega_2)_2$, $B_{\ell_1, \ell_2}^j(\omega_1, \omega_2)_3$ go to zero uniformly and have uniformly bounded derivatives, so we can absorb them into the remainders. So we let

$$R_{\ell_1, \ell_2}^j(t) = A_{\ell_1, \ell_2}^j + \sum B_{\ell_1, \ell_2}^j(\omega_1, \omega_2)_2 + \sum B_{\ell_1, \ell_2}^j(\omega_1, \omega_2)_3,$$

where the sums are over $(\omega_1, \omega_2) \in \Omega(\ell_1) \times \Omega(\ell_2)$, $\omega_1 + \omega_2 \neq 0$. We then set $V_{\ell_1, \ell_2}(t) = 0$, $V_{\ell_1, \ell_2}^j(t) = V_{\ell_1, \ell_2}(t) + R_{\ell_1, \ell_2}^j(t)$. We then have

$$\widetilde{UV}_{\ell_1, \ell_2}^j(t) = V_{\ell_1, \ell_2}^j(t) - \int_0^t u_{\ell_1}^j(s) \widetilde{UV}_{\ell_2}^j(s) ds$$

$$= j^{-\frac{2}{5}} \sum_{\substack{(\omega_1, \omega_2) \in \Omega(\ell_1) \times \Omega(\ell_2) \\ \omega_1 + \omega_2 \neq 0}} \frac{\eta_{\omega_1}(t)\eta_{\omega_2}(t)}{i^2\omega_2(\omega_2 + \omega_1)} e^{ij(\omega_1 + \omega_2)t}.$$

Once again, we can multiply $\widetilde{UV}_{\ell_2, \ell_3}^j(s)$ by $u_{\ell_1}^j(s)$, integrate, and split the result into an A part and a B part. The A part goes to zero uniformly and has uniformly bounded derivatives. The B part is a sum over triples $(\omega_1, \omega_2, \omega_3) \in \Omega(\ell_1) \times \Omega(\ell_2) \times \Omega(\ell_3)$. All these triples necessarily satisfy $\omega_1 + \omega_2 + \omega_3 \neq 0$ and $\omega_2 + \omega_3 \neq 0$. (Notice that there are no triples $(\omega_1, \omega_2, \omega_3)$ whose sum is equal to zero.) We then use integration by parts to rewrite the B part as a sum of three collections of terms, namely,

$$B_{\ell_1, \ell_2, \ell_3}^j(\omega_1, \omega_2, \omega_3)_1 = j^{-\frac{3}{5}} \frac{\eta_{\omega_1}(t)\eta_{\omega_2}(t)\eta_{\omega_3}(t)e^{ij(\omega_1 + \omega_2 + \omega_3)t}}{i^3\omega_3(\omega_3 + \omega_2)(\omega_3 + \omega_2 + \omega_1)},$$

and terms $B_{\ell_1, \ell_2, \ell_3}^j(\omega_1, \omega_2, \omega_3)_2$, $B_{\ell_1, \ell_2, \ell_3}^j(\omega_1, \omega_2, \omega_3)_3$ that go to zero uniformly and have uniformly bounded derivatives. Again, we incorporate the A , B_2 and B_3 parts into the remainder $R_{\ell_1, \ell_2, \ell_3}^j$, define $V_{\ell_1, \ell_2, \ell_3} = 0$, and let $V_{\ell_1, \ell_2, \ell_3}^j = V_{\ell_1, \ell_2, \ell_3} + R_{\ell_1, \ell_2, \ell_3}^j$. Then the $\widetilde{UV}_{\ell_1, \ell_2, \ell_3}^j$ are just the sum of $-j^{-\frac{3}{5}} \frac{\eta_{\omega_1}(t)\eta_{\omega_2}(t)\eta_{\omega_3}(t)e^{ij(\omega_1 + \omega_2 + \omega_3)t}}{i^3\omega_3(\omega_3 + \omega_2)(\omega_3 + \omega_2 + \omega_1)}$, ranging over $(\omega_1, \omega_2, \omega_3) \in \Omega(\ell_1) \times \Omega(\ell_2) \times \Omega(\ell_3)$, $\omega_1 + \omega_2 + \omega_3 \neq 0$, $\omega_2 + \omega_3 \neq 0$.

The next step is exactly the same, except that now there are 4-tuples of frequencies that add up to zero. So, before we carry out the integration by parts, the contribution of these terms must be controlled. This is done exactly as in our analysis of the terms of order two: any such 4-tuple $(\omega_1, \dots, \omega_4)$ must involve two pairs consisting of a frequency and its negative. The contribution of such a 4-tuple is then equal to a numerator that remains unchanged when $(\omega_1, \dots, \omega_4)$ is replaced by $(-\omega_1, \dots, -\omega_4)$, divided by a product of three linear functionals of the ω 's. The contribution of each pair and that of its negative cancel.

Finally, when we carry out the last step, we obtain an expression for $u_{\ell_1}^j \widetilde{UV}_{\ell_2, \ell_3, \ell_4, \ell_5}^j$ in which the exponent $j^{\frac{4}{5}}$ of the first factor exactly offsets the exponent of the second factor. When we integrate, all the terms whose five frequencies do not add up to zero

produce, exactly as before, A and B terms. Moreover, at this point the B_1 terms have a factor $\frac{1}{j}$ in the denominator, so they go to zero uniformly and have uniformly bounded derivatives. So all the terms coming from the 5-tuples of frequencies that do not add up to zero can be incorporated into the remainders. However, at this point there are 5-tuples that add up to zero. These 5-tuples can only occur for multiindices I such that $\delta_1(I) = 3$, $\delta_2(I) = 2$. So all the V_I for $|I| = 5$ will vanish, except possibly when $\delta_1(I) = 3$, $\delta_2(I) = 2$.

Finally, it is easy to compute V_I when $\delta_1(I) = 3$, $\delta_2(I) = 2$, and the result turns out to be precisely of the form given in (37). Indeed,

$$V_I(t) = \sum_{\hat{\omega} \in \tilde{\Omega}(I)} \int_0^t \frac{\eta_{\hat{\omega}}(s)}{\tilde{h}(\hat{\omega})} ds ,$$

where, for $I = (\ell_1, \ell_2, \ell_3, \ell_4, \ell_5)$, $\hat{\omega} = (\omega_1, \dots, \omega_5) \in \Omega(\ell_1) \times \dots \times \Omega(\ell_5)$, we let $\tilde{h}(\hat{\omega}) = h(\omega_2, \omega_3, \omega_4, \omega_5)$, and then define $\tilde{\Omega}(I)$ to be the set of those $\hat{\omega} = (\omega_1, \dots, \omega_5) \in \Omega(\ell_1) \times \dots \times \Omega(\ell_5)$ for which $\tilde{h}(\hat{\omega}) \neq 0$ and $\omega_1 + \dots + \omega_5 = 0$. The crucial point now is that the only way to choose the five components $\omega_1, \dots, \omega_5$ of $\hat{\omega}$ so that their sum is equal to zero is to take them to be the five elements of $\bar{\Omega}$, or their negatives, or the five elements of $\bar{\Omega}'$, or their negatives. Each of these four possibilities leads to 12 choices. For instance, the first possibility says that $(\omega_1, \dots, \omega_5)$ has to be the same as that (w_1, \dots, w_5) up to a permutation π of $\{1, \dots, 5\}$, i.e. $\omega_k = w_{\pi(k)}$. Moreover, π has to be such that $\{\pi(1), \pi(2), \pi(3)\} = \{k : \ell_k = 1\}$. The set P_I of all such permutations obviously contains 12 elements. Moreover, for each such $\hat{\omega}$ the corresponding contribution to V_I is $\frac{\eta_{w_1} \dots \eta_{w_5}}{h(\hat{\omega})}$. Summing over all the $\pi \in P_I$ gives a contribution

$$\left(\sum_{\pi \in P_I} \frac{1}{\tilde{h}(w_{\pi(1)}, \dots, w_{\pi(5)})} \right) \eta_{w_1} \dots \eta_{w_5} .$$

The second possibility, in which the ω_k are the negatives of the w_k , gives exactly the same contribution, except that $\eta_{w_1} \dots \eta_{w_5}$ is replaced by its complex conjugate. (Notice that the functions \tilde{h} are even.) The other two possibilities make an identical contribution, except that now the w_k s are replaced by the w'_k .

So Theorem 2 tells us that the control sequence defined in (12) and (13) EI-converges to

$$\mathbf{u}^\infty(t) = \eta_{1,0}(t)X_1 + \eta_{2,0}(t)X_2 + \sum_{I \in \Lambda} \frac{1}{5}v_I(t)[X_I], \quad (39)$$

where Λ consists of all the indices I such that $\delta_1(I) = 3$, $\delta_2(I) = 2$, and

$$v_I(t) = \Theta_I \xi(t) + \tilde{\Theta}_I \tilde{\xi}(t). \quad (40)$$

One can verify that there are six nonvanishing formal brackets $[X_I]$, $I \in \Lambda$. Moreover, to compute each v_I we need to compute Θ_I and $\tilde{\Theta}_I$, each of which is itself a sum over 12 permutations. So the above formula expresses the $(3, 2)$ homogeneous component of u^∞ as a sum of 144 terms. Although this is in principle not too hard to compute, we prefer to present a shortcut that considerably simplifies the calculation.

Clearly, each member of $\{[X_I] : I \in \Lambda\}$ is a linear combination of B_0 and B_1 . Moreover, the functions v_I are themselves linear combinations with constant coefficients of ξ and $\tilde{\xi}$. This means that

$$\sum_{I \in \Lambda} \frac{1}{5}v_I(t)[X_I] = (\Theta_1 \xi(t) + \Theta_2 \tilde{\xi}(t))B_0 + (\Theta_3 \xi(t) + \Theta_4 \tilde{\xi}(t))B_1.$$

So we need to figure out the values of the four coefficients Θ_i .

To do this, it is convenient to use the *product expansion of the Chen-Fliess series* described in [16]. This is a particularly simple way to express the solution of (36) as a product of exponentials $e^{C_B(t)B}$, where the B belong to a *P. Hall basis* of $L(\mathbf{X})$.

For details about the product expansion we refer the reader to [16]. The main purpose of this expansion is to solve a technical problem that arises when we try to express the solution of (36) in terms of iterated integrals. (We can regard (36) as evolving in $\hat{A}(\mathbf{X})$ or in the “truncated” finite-dimensional algebra $A_r(\mathbf{X})$.) Due to the Campbell-Hausdorff Formula, we know *a priori* that the solution $S(t)$ is an exponential $e^{Z(t)}$, where $Z(t)$ is a Lie series. It is clear that $Z(t)$ is uniquely determined by $S(t)$. However, if

we try to write $Z(t)$ as a linear combination of brackets such as the $[X_I]$ that we introduced earlier, we have to cope with the difficulty that these brackets are not linearly independent, because there are many linear relations among them, arising from the skew symmetry property $[P, Q] + [Q, P] = 0$ and the Jacobi identity $[P, [Q, R]] + [Q, [R, P]] + [R, [P, Q]] = 0$. This leads to a “combinatorial explosion,” an example of which was already encountered above when we pointed out that there are six nonzero brackets $[X_I]$ with $\delta_1([X_I]) = 3$, $\delta_2([X_I]) = 2$.

To reduce the combinatorial explosion one needs an explicit prescription for selecting a basis of $L(\mathbf{X})$. One such choice is provided by the use of a P. Hall basis. The precise definition is as follows. A *P. Hall basis* of $L(\mathbf{X})$ is a set \mathcal{B} of formal brackets $B \in \mathcal{FB}\mathcal{Br}(\mathbf{X})$ (cf. Remark 3), endowed with a total ordering \prec , such that the pair (\mathcal{B}, \prec) satisfies:

PH1. If $B, B' \in \mathcal{B}$ and $\delta(B) < \delta(B')$, then $B \prec B'$.

PH2. Every X_k is in \mathcal{B} .

PH3. If B is a formal bracket and $\delta(B) > 1$, so that B can be written in a unique way as $[B_1, B_2]$ (cf. Remark 3), then $B \in \mathcal{B}$ if and only if (i) $B_1 \in \mathcal{B}$, (ii) $B_2 \in \mathcal{B}$, (iii) $B_1 \prec B_2$ and (iv) either (iv.a) $\delta(B_2) = 1$ or (iv.b) $B_2 = [B_3, B_4]$ and $B_3 \preceq B_1$.

If $B \in \mathcal{B}$ is such that $\delta(B) > 1$, it is clear the left factor (cf. Remark 3) of B is also in \mathcal{B} . Moreover, B can be written in a unique way as $\text{ad}_{B_1}^*(B_2)$, where $B_1 \prec B_2$ and either $\delta(B_2) = 1$ or the left factor B_3 of B_2 satisfies $B_3 \prec B_1$. (Here we are using the standard notation ad_B to denote the operator $Z \rightarrow [B, Z]$.)

The main theorem of [16] gives an explicit formula for $S(t)$ as a product $S(t) = \prod_{B \in \mathcal{B}} e^{C_B(u)(t)B}$, with the factors ordered from right to left. The functions $C_B(u)(t)$ are given by $C_B(u)(t) = \int_0^t c_B(u)(s) ds$. The $c_B(u)$ and $C_B(u)$ are defined recursively as follows. For $B = X_k$, we let $c_B(u)(t) = u_k(t)$, $0 \leq t \leq T$. Assume

that $c_B(u)$ and $C_B(u)$ have been defined for all $B \in \mathcal{B}$ with degree $\leq n$, and let $B \in \mathcal{B}_{n+1}$. Then B can be written in a unique way as $\text{ad}_{B_1}^\kappa(B_2)$, where either $\delta(B_2) = 1$ or the left factor B_3 of B_2 satisfies $B_3 \prec B_1$. We then define

$$c_B(u) = \frac{1}{\kappa!} (C_{B_1}(u))^\kappa c_{B_2}(u).$$

With the help of the product formula, we can easily compute the values of the four coefficients Θ_i . First, notice that the linear combination $2\Theta_1 B_0 + 2\Theta_3 B_1$ is precisely the limiting extended input $\bar{\mathbf{u}}^\infty$ that we would have obtained if we had used, instead of our u^j , the HOS $\{\bar{u}^j\}$ given by a formula similar to (13) but with $\eta_{1,0} = \eta_{2,0} \equiv 0$, $\eta_{w_\ell} \equiv 1$, and $\eta_{w'_\ell} \equiv 0$. So to compute Θ_1 and Θ_3 it suffices to set

$$\bar{\Omega}(1) = \{\pm w_1, \pm w_2, \pm w_3, \}, \quad (41)$$

$$\bar{\Omega}(2) = \{\pm w_4, \pm w_5, \}, \quad (42)$$

$$\bar{u}_1^j(t) = j^{\frac{1}{5}} \sum_{\omega \in \bar{\Omega}(1)} e^{ij\omega t}, \quad (43)$$

$$\bar{u}_2^j(t) = j^{\frac{1}{5}} \sum_{\omega \in \bar{\Omega}(2)} e^{ij\omega t}, \quad (44)$$

compute the product expansion of the corresponding solution \bar{S}^j of (36), and then take the limit as $j \rightarrow \infty$. We know that the limit \bar{S}^∞ of the S^j satisfies

$$\frac{d}{dt} \bar{S}^\infty(t) = \bar{S}^\infty(t) \bar{\mathbf{u}}^\infty(t).$$

Therefore

$$\bar{S}^\infty(t) = e^{t(2\Theta_1 B_0 + 2\Theta_3 B_1)}.$$

If we work in the nilpotent algebra $A_5(\mathbf{X})$, then B_0 and B_1 commute, so we have $\bar{S}^\infty(t) = e^{2\Theta_1 t B_0} e^{2\Theta_3 t B_1}$. It is easily seen that B_0 and B_1 are different elements of a P. Hall basis. So

$$\Theta_1 = \frac{1}{2t} \lim_{j \rightarrow \infty} C_{B_0}(\bar{u}^j)(t),$$

$$\Theta_3 = \frac{1}{2t} \lim_{j \rightarrow \infty} C_{B_1}(\bar{u}^j)(t).$$

The product formula gives

$$\begin{aligned} C_{B_0}(\bar{u}^j)(t) &= \frac{1}{2!} \int_0^t \left(\int_0^s \left(\int_0^{\tau_1} \bar{u}_1^j(\tau_2) d\tau_2 \right) \bar{u}_2^j(\tau_1) d\tau_1 \right) \\ &\quad \times \left(\int_0^s \bar{u}_1^j(\tau) d\tau \right)^2 \bar{u}_2^j(s) ds, \\ C_{B_1}(\bar{u}^j)(t) &= \frac{1}{3!} \int_0^t \left(\int_0^s \bar{u}_2^j(\tau) d\tau \right) \left(\int_0^s \bar{u}_1^j(\tau) d\tau \right)^3 \bar{u}_2^j(s) ds. \end{aligned}$$

To study the asymptotic behavior of these expressions as $j \rightarrow \infty$, we just expand each \bar{u}_ℓ^j as a sum of six terms (if $\ell = 1$) or four terms (if $\ell = 2$). Then C_{B_0} and C_{B_1} are expressed as sums of $6^3 2^2 = 864$ terms, which may appear to be combinatorially overwhelming. However, it is not hard to see that the only terms that make a nontrivial contribution to the asymptotic behavior are those that correspond to five frequencies that add up to zero. There are exactly 24 such terms. For C_{B_0} , a typical term of this kind is the one that corresponds to the frequencies w_1, w_4, w_2, w_3, w_5 . This term contributes $\frac{t}{2w_1 w_2 w_3 (w_1 + w_4)}$ to the limit of $C_{B_0}(\bar{u}^j)(t)$. If we symmetrize with respect to $\{w_1, w_2, w_3\}$, and with respect to $\{w_4, w_5\}$, we get 12 terms whose total contribution is precisely $g_{B_0}(w_1, \dots, w_5)$. Another contribution equal to $g_{B_0}(w_1, \dots, w_5)$ results from the 12 terms obtained by replacing the w_k by their negatives. So the limit of $C_{B_0}(\bar{u}^j)(t)$ is $2g_{B_0}(w_1, \dots, w_5)t$. Similarly, the limit of $C_{B_1}(\bar{u}^j)(t)$ is $2g_{B_1}(w_1, \dots, w_5)t$. From this it follows that $\Theta_1 = g_{B_0}(w_1, \dots, w_5)$ and $\Theta_3 = g_{B_1}(w_1, \dots, w_5)$. The formulas for Θ_2 and Θ_4 are exactly the same, except that the w_k are replaced by the w'_k .

The proof of Theorem 1 is now complete. ■

Remark 9. Using the appropriate algebraic formalism, one can get simple expressions for the functions g_B associated to every P. Hall bracket. One can then prove the following in complete generality. Let ν_1, ν_2 be integers > 0 . Write $\nu = \nu_1 + \nu_2$, and let $\mu = \dim L(\mathbf{X})_{\nu_1, \nu_2}$, so that there are exactly μ P. Hall brackets B such that $\delta_1(B) = \nu_1$ and $\delta_2(B) = \nu_2$. Then the corresponding functions g_B are linearly independent on the hyperplane $H = \{(\omega_1, \dots, \omega_\nu) : \omega_1 + \dots + \omega_\nu = 0\}$. This fact, that was verified directly here for $\nu_1 = 3, \nu_2 = 2, \mu = 2$, guarantees that the analogue of the linear system (17), (18) always has a solution. ■

SOME REMARKS ON THE GENERAL CASE

The analysis carried out in this paper applies almost without change to a general single-bracket extension of a general m -input driftless system. The only new technical difficulty is the problem discussed in Remark 9, that is, the linear independence of the functions g_B for different P. Hall brackets of the same degree with respect to each X_ℓ .

General multibracket extensions can then be handled by superposition, as explained above. (We have already encountered one example of such superposition: the limit input \mathbf{u}^∞ is the sum of two parts, namely, the one arising from the w_ℓ and that resulting from the w'_ℓ . This was a consequence of the independence of the two sets of frequencies. It turns out that, provided one takes care of some algebraic complications, this generalizes. One can produce a desired limiting extended input \mathbf{v} by taking an HOS which is a sum of parts, one for each totally homogeneous component, and making sure that the corresponding sets of frequencies are independent.)

DRIFT AND FEEDBACK

A trivial particular case of the above is the seemingly more general situation where our original system is of the form

$$\dot{x} = f_0(x) + \sum_{k=1}^m u_k(t) f_k(x), \quad (45)$$

(i.e. has a drift term), but the extended system

$$\dot{x} = f_0(x) + \sum_{k=1}^r v_k(t) f_k(x), \quad (46)$$

has the special property that the new brackets f_k , $k > m$, do not involve f_0 . In this case we can apply our general construction thinking of (45) as a system without drift, and of the extended input that we want to approximate as one whose X_0 component

happens to be 1. Since the new brackets do not involve f_0 , our construction yields approximating inputs u^j whose zeroth component is just 1, so we end up producing inputs for (45) that T-converge to the desired limit.

It has been suggested by L. Gurvits (personal communication) that the approximation algorithm can be used to produce *feedback laws*, as follows. Suppose that for a suitable Lie bracket extension (46) of (45) there is a feedback control law $v_k = \varphi_k(x)$, $k = 1, \dots, r$, that has some desired properties. Then one may try to produce time-dependent feedback laws $u_k = \psi_{k,\epsilon}(x, t)$ that, as $\epsilon \rightarrow 0$, give rise to trajectories that converge to those of the closed-loop system

$$\dot{x} = f_0(x) + \sum_{k=1}^r \varphi_k(x) f_k(x), \quad (47)$$

at least on some fixed time-interval $[0, T]$. Our algorithm makes it possible to produce such an approximation as follows. Assuming that the functions φ_k are sufficiently smooth, then we can regard the control system

$$\dot{x} = f_0(x) + \sum_{k=1}^r w_k \varphi_k(x) f_k(x), \quad (48)$$

as arising from a control specialization of a Lie bracket extension of a system

$$\dot{x} = f_0(x) + \sum_{k=1}^{\rho} \hat{w}_k g_k(x), \quad (49)$$

where the vector fields g_k are linear combinations of f_1, \dots, f_m with sufficiently smooth coefficients. For example: suppose $m = 2$, and (47) is

$$\dot{x} = f_0(x) + \varphi_1(x) f_1(x) + \varphi_2(x) f_2(x) + \varphi_3(x) [f_1, f_2](x).$$

We rewrite $\varphi_3[f_1, f_2](x)$ as $[f_1, \varphi_3 f_2] - (L_{f_1} \varphi_3) f_2$, and let $g_1 = f_1$, $g_2 = \varphi_1 f_1$, $g_3 = \varphi_2 f_2$, $g_4 = (L_{f_1} \varphi_3) f_2$, $g_5 = \varphi_3 f_2$. Then (48) is just given by

$$\dot{x} = f_0(x) + w_1 g_2(x) + w_2 g_3(x) + w_3 g_4(x) + w_3 [g_1, g_5](x),$$

which arises from the extension

$$\Sigma_e : \dot{x} = f_0(x) + \sum_{k=1}^5 \bar{w}_k g_k(x) + \bar{w}_6[g_1, g_5](x)$$

of

$$\Sigma : \dot{x} = f_0(x) + \sum_{k=1}^5 \hat{w}_k g_k(x)$$

by specializing the control to $\bar{w}_1 = \bar{w}_5 = 0$, $\bar{w}_2 = w_1$, $\bar{w}_3 = w_2$, $\bar{w}_4 = \bar{w}_6 = w_3$. If we use our algorithm to produce a sequence $\{\hat{w}^j\}$ of inputs for Σ that T-converge to the extended input $X_2 + X_3 + X_4 + [X_1, X_5]$, then the time-dependent feedbacks

$$\begin{aligned} u_1^j(x, t) &= \hat{w}_1^j(t) + \hat{w}_2^j(t)\varphi_1(x), \\ u_2^j(x, t) &= \hat{w}_3^j(t)\varphi_2(x) + \hat{w}_4^j(t)(L_{f_1}\varphi_3)(x) + \hat{w}_5^j(t)\varphi_3(x) \end{aligned}$$

have the desired properties.

References

- [1] Bourbaki, N., *Elements of Mathematics: Lie Groups and Lie Algebras*, Hermann (1975) Paris.
- [2] Brockett, R. W., "Control Theory and Singular Riemannian Geometry," in *New Directions in Applied Mathematics*, Springer-Verlag, New York (1981), pp. 11-27.
- [3] Fernandes, C., L. Gurvits, and Z.X. Li, "Foundations of Nonholonomic Motion Planning," Technical Report, Robotics Research Laboratory, Courant Institute of Mathematical Sciences, 1991.
- [4] Gurvits, L., and Z.X. Li, "Theory and Application of Nonholonomic Motion Planning," Technical Report, Courant Institute of Mathematical Sciences, 1990.
- [5] Haynes, G. W., and H. Hermes, "Nonlinear Controllability via Lie Theory," *SIAM J. Control and Optimization* 8, No. 4, Nov. 1970, pp. 450-460.

- [6] Hauser, J., S. S. Sastry, and P. V. Kokotovic, "Nonlinear Control via Approximate Input-output Linearization: The Ball and Beam Example," *Proc. 28th IEEE CDC*, Tampa, Florida, Dec. 1989, pp. 1987-1993.
- [7] Kurzweil, J., and J. Jarnik, "Limit Process in Ordinary Differential Equations," *Journal of Applied Mathematics and Physics* **38**, March 1987, pp. 241-256.
- [8] Kurzweil, J., and J. Jarnik, "Iterated Lie Brackets in Limit Processes in Ordinary Differential Equations," *Results in Mathematics* **14**, 1988, pp. 125-137.
- [9] Kurzweil, J., and J. Jarnik, "A Convergence Effect in Ordinary Differential Equations," in *Asymptotic Methods of the Mathematical Physics*, (Russian), 301, 'Naukova Pumka,' Kiev, 1989.
- [10] Lafferriere, G., "A General Strategy for Computing Steering Controls of Systems without Drift," Submitted to the 1991 CDC.
- [11] Lafferriere, G., and H. J. Sussmann, "Motion Planning for Controllable Systems without Drift: A Preliminary Report," Rutgers University Systems and Control Center Report SYCON-90-04.
- [12] Lafferriere, G., and H. J. Sussmann, "Motion Planning for Controllable Systems without Drift," *Proc. IEEE Int. Conf. Robotics and Automation*, Sacramento, CA, April 1991.
- [13] Murray, R. M., and S. S. Sastry, "Grasping and Manipulations Using Multifingered Robot Hands," Memorandum No. UCB/ERL M90/24, Electronics Research Laboratory, Univ. of California, Berkeley, March 1990.
- [14] Murray, R. M., and S. S. Sastry, "Steering Controllable Systems," *Proc. 29th IEEE CDC*, Honolulu, Hawaii, Dec. 1990.
- [15] Sastry, S. S., and Z. Li, "Robot Motion Planning with Nonholonomic Constraints," *Proc. 28th IEEE CDC*, Tampa, Florida, Dec. 1989, pp. 211-216.

- [16] Sussmann, H. J., "A Product Expansion for the Chen Series," in *Theory and Applications of Nonlinear Control Systems*, C. Byrnes and A. Lindquist Eds., North-Holland, 1986, pp. 323-335.
- [17] Sussmann, H. J., "A General Theorem on Local Controllability," *SIAM J. Control and Optimization* **25** No. 1, January 1987, pp. 158-194.
- [18] Sussmann, H. J., and W. Liu, "Limiting Behavior of Trajectories for Highly Oscillating Controls," in preparation.
- [19] Sussmann, H. J., and W. Liu, "Limits of Highly Oscillatory Controls and Approximation of General Paths by Admissible Trajectories." To appear in *Proc. 30th IEEE CDC*, Brighton, UK, Dec. 1991.

5

Singularities and Topological Aspects in Nonholonomic Motion Planning

JEAN-PAUL LAUMOND *

LAAS/CNRS
7 AVENUE DU COLONEL ROCHE
31077 TOULOUSE, FRANCE

Abstract

Motion planning is an already old and classical problem in Robotics. A few years ago a new instance of this problem has appeared in the literature : *motion planning for nonholonomic systems*. While useful tools in motion planning come from Computer Science and Mathematics (Computational Geometry, Real Algebraic Geometry), nonholonomic motion planning needs some Control Theory and more Mathematics (Differential Geometry).

*A preliminary version of this paper appears as the technical report STAN-CS 90-1345 written during a stay of the author at Stanford in the Robotics Laboratory of the Department of Computer Science. His stay was funded by DARPA/Army contract DAAA21-89-C0002.

First of all, this paper tries to give a computational reading of the tools from Differential Geometric Control Theory required by planning. Then it shows that the presence of obstacles in the real world of a real robot challenges Mathematics with some difficult questions which are topological in nature. Some of them have been solved only recently, within the framework of Sub-Riemannian Geometry. Many other ones constitute open problems at this time.

Introduction

The motion planning problem (MP) is certainly one of the best formulated ones in Robotics. It raises two questions :

- Can a robot reach a given goal while avoiding the obstacles of its environment ? This is the *decision* problem.
- If the answer to the previous question is yes, *what path* may it follow ? This is the *complete* problem.

The geometric formulation of this problem is known as the piano mover problem. This formulation considers the motion of rigid bodies amidst obstacles in the 3-dimensional Euclidean space. The placement (translation and rotation) of a body in the Euclidean space is given by a point in a 6-dimensional space. Some geometric relationship between the bodies may appear for a given robotic system (as is typical for a robot arm). They are translated into equations between the placement parameters of the bodies. These are called the *holonomic links*. They restrict the space of the allowed placements to a subspace of the placement spaces of all the bodies. This subspace is called the *configuration space*. Finally, a configuration of the robot is represented by a point of the configuration space that defines precisely the domain occupied by the robot in Euclidean space. A point-path in the configuration space

corresponds to a motion of the robot. For a holonomic system, we have as many degrees of freedom as is needed to follow *any* path.

Therefore, for holonomic systems, the existence of a collision-free path is characterized by the existence of a connected component in the admissible (i.e., collision-free) configuration space. To solve MP, it is enough to compute the admissible configuration space (i.e., transform the obstacles in the Euclidean space into “obstacles” in the configuration space), and then explore its connected components.

Since the seventies this problem has attracted many researchers working in Robotics and beyond, in Computational Geometry and Real Algebraic Geometry. See [60] for a recent overview, [76] for a review of the various approaches of the problem, and [37] as the first genuine book on the subject.

However, there are cases in which this formulation of motion planning is not sufficient. In the last four years, an example of the limitation of the piano mover formulation has been investigated : planning constrained motions where constraints are nonholonomic in nature. A *nonholonomic link* is expressed as a non-integrable equation involving derivatives of the configuration parameters. Such constraints are expressed in the tangent space at each configuration; they define the allowable velocities of the system, and they *cannot* be eliminated by defining a more restricted configuration space manifold. Thus, the main consequence of a nonholonomic constraint is that an arbitrary path in the admissible configuration space does not necessarily correspond to a feasible path for the robot. Therefore, the existence of a collision-free path is not *a priori* characterized by the existence of a connected component in the admissible configuration space.

Planning motions for nonholonomic systems is not as new in other communities as in the community working on obstacle avoidance for robots¹. This problem is well known in Nonlinear Control

¹Notice that nonholonomic motion planning appears also in some spatial applications, for systems (like space-stations or satellite) using internal motion and submitted to conservation laws (see [52,54] for instance, and [19] for the

and in Differential Geometry. Important results have been obtained over the last two decades, while the first results seem dated from the thirties with Chow's work [16].

Notwithstanding, Robotics brings to the front an important constraint which is not usually taken into consideration : the planner has to produce paths that *avoid the obstacles*. Moreover, by its applications to the real world, it requires effective and efficient computational tools, rather than just proofs of existence.

Results useful to our problem can be found in publications—often difficult ones—from other communities than the Robotics one. Because the viewpoints are different, they attack only some aspects of the problem, and use different terminologies. The goal of this paper is to enlighten these points of view by a computational one (the right point of view for the planning problem) and to stress the connections between motion planning and differential geometric control theory. This study has to be viewed as an informal statement of these connections, hopefully readable by a non-specialist in control theory or in differential geometry (as the author is). A spotlight will be focused on some concepts from these theories, using a minimal formalism while trying to understand where and why these are pertinent as far as the motion planning problem is concerned. It is clear that an in-depth study of these concepts needs the precision given by the mathematical formalism : for each concept there will be references introducing or using it.

The decision problem of planning for nonholonomic systems is related to their controllability² : more precisely the existence of a

amusing—and complicated—case of a falling cat).

²The use of the term “controllability” in this context is fuzzy in the community. Indeed, the meaning we use here is related to the reachability concept. A nonholonomic system may be controllable by *open* loops, while, by Brockett's necessary conditions for stability, one may demonstrate that it cannot be stabilized to a point with smooth state feedback [10] (see [59,13,1] for studies of feedback controls for nonholonomic wheeled carts). It would be better to use the notion of a *completely nonholonomic system* related to the concept of a distribution (see [73]). This paper adheres to Sussmann's terminology [66], which seems to have reached some state of general acceptance.

collision-free path for a controllable nonholonomic robot is characterized by the existence of an *open* connected component of the admissible configuration space. The decision problem is then similar to that of the piano mover problem. This result constitutes the first main link; it has been studied simultaneously in several research groups of the Robotics community : [42,46,4,49] (see Murray's PhD Thesis [51] for a survey of the very last progress). It is based upon the Lie algebra rank condition, and will be recalled in Section 2. Then we consider the computational aspects of the problem : is a system controllable ? We give a semi-decidable procedure for solving this problem *locally*. Then we will see that the *global* point of view of the planning deals with the existence of *singularities*. Finally Section 2 introduces the *well-controllability* notion, in relation to the planning problem.

Nevertheless, at this stage the complete problem of nonholonomic motion planning remains unsolved.

Section 3 considers the complete problem. We will see that the key question is *topological* in nature. While heuristic or specific methods for nonholonomic systems have blossomed through the last three years [39,4,71,75,42,21], very recent contributions pursue a deep study of the differential geometric tools available for solving the complete problem. Lafferriere and Sussmann present the first general planner based upon a general constructive proof of controllability [36]. At the same time, Murray and Sastry show how to solve the problem for some "canonical" systems [50]. Nevertheless, both references do not address the obstacle avoidance (Section 3.1). This point is attacked by Sussmann and Liu [67]. Finally, in [32], Jacobs, Laumond and Taïx present a complete and efficient planner for mobile robots and show that its strategy can be generalized. These results (surviewed in Section 3.3) let us stress the main difficulty for building efficient planners : while obstacle avoidance requires us to consider the "natural" Riemannian topology of the configuration space (i.e., induced by the natural Hausdorff metric working in the robot environment), the paths allowed by the nonholonomic constraints compel us to consider another topology in this space : the topology induced by the length of the shortest *allowed* paths between two points. Such

a metric is known as a sub-Riemannian (or singular, or Carnot-Caratheodory) metric. In fact, using sub-Riemannian geometry (see [9,63,73,48]), it is possible to show that both topologies are the same (Section 3.2). This result enables us to conclude on the generality of the approaches presented in [32]. Nevertheless, because we are interested in the computational point of view, we have to study more deeply the shape of the sub-Riemannian metrics in order to estimate the combinatorial complexity of the planners. This study has been done in [32] for the case of the car-like robot. This section points at the reference [73] that gives precisely the general and finest form of the sub-Riemannian metric we need to conclude on the complexity of the complete problem.

Other results related to nonholonomic motion do not use intensively the tools we present in this paper, but they are interesting nonetheless from either a theoretical or a practical point of view. We just give here a list of references :

- Car-like and Trailer-like Robots [39,4,71,75,42,21]
- Smooth Paths [18,40,20,29]
- Shortest Paths [18,29,28,71,56]
- Time-Optimal Paths [53,11,31,12]
- Control-Oriented Approaches [17,34,58,35]

The decision problem

Until a very recent period, the main contribution to nonholonomic motion planning (independently developed in [42] and [46]) has been to solve the decision part of the nonholonomic motion planning problem, via differential geometry and control theory.

What is the problem ?

While the constraints due to the obstacles are expressed directly in the manifold of configurations, nonholonomic constraints deal with the tangent space. In the presence of a link between the robot's parameters and their derivatives, two questions arise :

- Is this link holonomic ? (i.e., does it reduce the dimension of the configuration space ?)
- If not, does it reduce the accessible configuration space ?

In the case of r links corresponding to r equations linear in the derivatives of the n parameters, these equations determine what is called an $(n - r)$ -distribution Δ on the manifold of configurations. The answer to the first question is then given by Frobenius' theorem (see for instance [61]) : the equations are integrable if and only if the distribution Δ is closed under the Lie bracket operation. Let us recall that the Lie bracket of two vector fields X and Y is defined as $[X, Y] = \partial X.Y - \partial Y.X$. A sample of computation examples appears in Appendix 2.

From a control theory perspective, a control is a function which allows us to choose the system state velocity at each instant by a careful weighting of smooth vector fields. The control Lie algebra associated with Δ , denoted by $LA(\Delta)$, is the smallest distribution which contains Δ and is closed under the Lie bracket operation. The answer to the second question is then given by the non-linear system controllability theorem (see for instance [65,47,24]) : if the rank of the Lie algebra is full at a given configuration c , then there exists a neighborhood \mathcal{N} of c whose points represent configurations reachable by the system moving from c along an admissible path. Moreover, this path stays in \mathcal{N} . This condition is known as the “rank condition”; it is a *local* condition³.

³See [4] for a more detailed introduction.

If the rank condition holds everywhere in the configuration space, then the robot is termed *controllable*⁴. From our planning point of view, the main consequence is that the existence of a collision-free path is characterized by the existence of a connected component in the *free* (i.e., with neither collision nor contact) configuration space.

Therefore, apart from topological subtleties dealing with motions in contact, **the decision problem of motion planning for controllable systems is the same as for holonomic ones**. The following section highlights the difficulty for proving the controllability of a system. Since we are interested in the computational point of view, we are looking for a *procedure* that allows us to conclude.

Remark : The difference is more involved with the complete problem. The previous result answers the question of the existence of a feasible path, that is, the decision problem, but does not solve the problem of efficiently producing an admissible path. To the extent of the author's knowledge, there was no general *constructive* proof of the result mentioned above until the recent contribution of Lafferriere and Sussmann [36]. Specific constructive proofs appear in [39] for the car-like robot and in [42] for the car-like robot pulling a trailer. At this stage we can just hope that the search for a solution for a nonholonomic system can be guided by a collision-free path for the associated holonomic system. Indeed, thanks to the local property above, a controllable robot can be steered close to any path as long as there is a "small gap" between the reference path and the obstacles. This idea is precisely the basis of the two strategies defined in [67] and [32] that we will resume in Section 3. It appears clearly that the key question for developing such a strategy deals with the size of the "gap", i.e., with the *topology* induced by the nonholonomic constraints.

⁴In fact a controllable system requires only that the rank condition holds nearly everywhere, that is on a dense subset of the manifold. Various precise controllability concepts appear involving the "size" of reachable sets and the "size" of sets where the rank condition holds (see [66]). Omitting them does not affect our purpose.

Proving the controllability of an n -dimensional system using the rank condition involves showing that, for any point c in the manifold, there exists a family of n vectors fields in the Control Lie Algebra that spans \mathbf{R}^n when applied to c . This stirs two difficulties due to the *local* and *global* characteristics of the problem :

- At any specific point c , finding such a family enables us to conclude. Not being able to find a suitable family does not imply that there is none. An exhaustive enumeration of possible families is impossible since there is an infinity of potential choices. We will see that this number can be reduced to a countable one, but not further, leading to the design of some semi-decidable procedures. We will proceed then to giving an estimate of the complexity of procedures testing the controllability of a system at a point.
- One may succeed in finding bases that work somewhere, but not necessarily *everywhere*. There may be some singularities. An interesting problem is to know whether such singularities have an intrinsic nature, or depend upon the choice we make.

As far as we know, testing controllability is not a decidable problem.

Distributions and Filtrations

The material of this section uses the concepts of a distribution, also known as a Pfaffian system (see for instance [73]), and of the Free Lie Algebra (see [8]).

Let us recall that every Lie operator has to verify skew-symmetry $[X, Y] = -[Y, X]$ and the Jacobi identity $[X, [Y, Z]] + [Y, [Z, X]] + [Z, [X, Y]] = 0$.

Consider the $(n - r)$ -distribution Δ associated with a robotic system. We want to define an *algorithm* for testing the controllability of that system at a point. Precisely, we are interested in

the rank of $LA(\Delta)$ (i.e., the distribution spanned by all the combinations of Lie brackets of vector fields in Δ). We can consider a basis \mathcal{X} of Δ together with all the combinations of Lie brackets built upon that basis.

To do this, one may consider a brute force strategy consisting in building iteratively the following increasing sequence of distributions : $\Delta_i = \Delta_{i-1} + [\Delta_{i-1}, \Delta_{i-1}]$ where $[\Delta_{i-1}, \Delta_{i-1}]$ is the linear space spanned by all the brackets $[X, Y]$ for X and Y in Δ_{i-1} . By putting $\Delta = \Delta_1$, the Control Lie Algebra $LA(\Delta)$ is precisely defined as $\bigcup_i \Delta_i$. But in fact, a more efficient strategy can be used. First of all, let us define a parameter estimating the complexity of a combination of Lie brackets. The *degree* of a combination is the number of elements in \mathcal{X} defining the combination. For example the degree of $[., [., [., .]], [., [., .]]]$ is 7. Now, our strategy will consist in building all the brackets of a given degree, step by step. This strategy is founded on the following iterative construction. We denote Δ by Δ_1 . Then Δ_i is defined by :

$$\Delta_i = \Delta_{i-1} + \sum_{j+k=i} [\Delta_j, \Delta_k].$$

It verifies :

$$\Delta_1 \subset \Delta_2 \subset \Delta_3 \subset \cdots \subset LA(\Delta) \quad \text{and} \quad LA(\Delta) = \bigcup_i \Delta_i.$$

The set of all the Δ_i s is called a *filtration* associated with Δ .

Remark : Such a construction can be viewed as a “breadth-first” construction. Some authors [73, 72] use another construction. $\tilde{\Delta}$ is denoted by $\tilde{\Delta}_1$. Then $\tilde{\Delta}_i$ is defined by :

$$\tilde{\Delta}_i = \tilde{\Delta}_{i-1} + [\tilde{\Delta}_1, \tilde{\Delta}_{i-1}].$$

Again :

$$\tilde{\Delta}_1 \subset \tilde{\Delta}_2 \subset \tilde{\Delta}_3 \subset \cdots \subset LA(\Delta).$$

Such a construction can be viewed as a “depth-first” construction. Using skew-symmetry and the Jacobi identity, we may verify that

both constructions are the same⁵. We will prefer the first presentation that corresponds exactly to the concept of Phillip Hall families introduced below.

On the other hand, at a point c of our manifold (the configuration space), $(n - r) \leq \text{rank } \Delta_i(c) \leq n$ ⁶. Moreover, if $\Delta_i(c) \neq \Delta_{i-1}(c)$, then $\text{rank } \Delta_i(c) > \text{rank } \Delta_{i-1}(c)$. Hence, if we consider the construction *locally* (i.e., by applying the distributions at a point), we can conclude that there exists an index p_c such that $\Delta_{p_c-1}(c) \neq \Delta_{p_c}(c) = \Delta_{p_c+1}(c) = \dots$. The construction always stabilizes. The index p_c is the *degree of nonholonomy* of the system at c . Therefore a system is controllable at c if and only if $\text{rank } \Delta_{p_c} = n$ (if $p_c = 1$ we are locally in the holonomic situation). Notice that, from a global point of view, this stabilization property is not true, since the degree of nonholonomy may change from point to point. A close analysis of possible singularities shows that this degree may be arbitrarily high at singular points—even when we start with a regular distribution, the filtration we build may acquire some weird singularities. So, the degree of nonholonomy may be unbounded when c varies.

Remark : It is possible to define a *global* degree of nonholonomy of a nonholonomic system, as the maximum of pointwise degrees of nonholonomy. There are no obvious applications of this notion. Also, keep in mind that this global degree can be infinite, though it will stay bounded in the particular cases we consider.

⁵For example, take $[[X, Y], [X, [X, Z]]]$, an element of Δ_5 :

$$\begin{aligned} [[X, Y], [X, [X, Z]]] &= -[X, [X, [Y, [X, Z]]]] + [X, [Y, [X, [X, Z]]]] + \\ &\quad [X, [Z, [X, [X, Y]]]] - [Z, [X, [X, [X, Y]]]]. \end{aligned}$$

Hence, it belongs to $\tilde{\Delta}_5$ too.

⁶We denote by $\Delta_i(c)$ the linear subspace of the tangent space in c , obtained by applying the distribution Δ_i at c .

A Controllability Algorithm

In this section we define an algorithm for testing the controllability of a given system at a point based upon the previous construction. We have to use a basis \mathcal{X} of Δ . According to that construction, we build :

$$\begin{aligned}\mathcal{X}_1 &= \mathcal{X} \\ \mathcal{X}_i &= \mathcal{X}_{i-1} \bigcup_{j+k=i} [\mathcal{X}_j, \mathcal{X}_k]\end{aligned}$$

where, now, $[\mathcal{X}_j, \mathcal{X}_k]$ is no longer viewed as a linear space, but as a *finite* family of brackets. Each \mathcal{X}_i contains of course a basis of Δ_i . Again, we can define the union $\mathcal{LA}(\mathcal{X})$ of all these families and we have :

$$\mathcal{X}_1 \subset \mathcal{X}_2 \subset \mathcal{X}_3 \subset \cdots \subset \mathcal{LA}(\mathcal{X})$$

This is clearly an infinite family, but, during the real process, we can check out the added elements if they happen to be linearly dependent on the previous ones.

Even if we know only about the relations pertinent to the concept of a Lie Algebra, we can take advantage of these to compute only relevant elements of what is called the *Free Lie Algebra*.

Phillip Hall Families

In this section⁷ the elements of $\mathcal{LA}(\mathcal{X})$ are considered as formal expressions produced by the construction above, i.e., they are not actually evaluated as vector fields belonging to a distribution. From this point of view, $\mathcal{LA}(\mathcal{X})$ is considered as a *Free Lie Algebra*. Our current problem is to enumerate a basis of this algebra, i.e., to get rid of redundant elements using only skew-symmetry and

⁷The material used in this section comes from [8]. We want just to give a rough idea of the concept and of its pertinence with respect to our problem. Interested readers will find a more rigorous presentation in this reference.

the Jacobi identity. Such a basis can be found via a Phillip Hall family.

The degree of an element X in $\mathcal{LA}(\mathcal{X})$ is denoted by $\text{degree}(X)$: this is the degree of the monomial defining X ⁸. According to our notations, a *Phillip Hall family* (PH-family for short) of $\mathcal{LA}(\mathcal{X})$, is any totally ordered subset (\mathcal{PH}, \prec) such that :

- If $X \in \mathcal{PH}$, $Y \in \mathcal{PH}$ and $\text{degree}(X) < \text{degree}(Y)$ then $X \prec Y$;
- $\mathcal{X} \subset \mathcal{PH}$;
- $\mathcal{PH} \cap \mathcal{X}_2 = \{[X, Y], \quad X \prec Y\}$;
- An element $X \in \mathcal{LA}(\mathcal{X})$ with $\text{degree}(X) \geq 3$ belongs to \mathcal{PH} if and only if $X = [U, [V, W]]$ with U, V, W in \mathcal{PH} , $[V, W]$ in \mathcal{PH} , $V \preceq U \prec [V, W]$ and $V \prec W$.

The main property of a PH-family is that, taking skew-symmetry and the Jacobi identity into account, it yields a *basis* of the free Lie algebra $\mathcal{LA}(\mathcal{X})$ [8].

The proof of existence of such a family is easy; it is an iterative one. In the context of our control problem, it can be extended into the following algorithm.

The Algorithm

The idea is to build a PH-family, based upon a graded family of sets \mathcal{H}_i , where \mathcal{H}_i is a part of \mathcal{X}_i . We will also build a total order \prec on the union \mathcal{H}_i . Assume first that \mathcal{X} is totally ordered by $<$ and set $\mathcal{H}_1 = \mathcal{X}$. The order \prec on \mathcal{H}_1 is the same as the order $<$ on

⁸We use the word “degree” with two different meanings, according to whether we speak of a bracket or of a nonholonomic system. This may introduce some confusion, but both terms are already used in the literature (see for instance [72]).

\mathcal{X} . The next set \mathcal{H}_2 is defined as the set of all $[X, Y]$, with X, Y elements of \mathcal{H}_1 and $X \prec Y$. Endow \mathcal{H}_2 with any total order, and define \prec on $\mathcal{H}_1 \cup \mathcal{H}_2$ by setting $X \prec Y$, for X in \mathcal{H}_1 and Y in \mathcal{H}_2 .

The rest of the algorithm consists in building the sets \mathcal{H}_i iteratively. Suppose the family $\mathcal{H}_1, \mathcal{H}_2, \dots, \mathcal{H}_{i-1}$ is given. Denote it as \mathcal{H} . Define \mathcal{H}_i according to the definition of a PH-family. That is, \mathcal{H}_i is the set of all $X = [U, [V, W]]$ verifying : $U \in \mathcal{H}_j$, $[V, W] \in \mathcal{H}_{i-j}$, $V \preceq U \prec [V, W]$ and $V \prec W$. Choose a total order on \mathcal{H}_i and extend it to $\mathcal{H} \cup \mathcal{H}_i$: $X \prec Y$ if $X \in \mathcal{H}$ and $Y \in \mathcal{H}_i$. It is almost obvious that the family \mathcal{H}_i is a PH-family and, furthermore, that the degree of an element of \mathcal{H}_i is precisely i .

We can use this construction to design an algorithm for testing the controllability of a system at a point c of the manifold. Our algorithm adds new brackets to the PH-family step by step, but now, we check further the value of each new bracket as a potential member of a basis at c . If we ultimately obtain a basis, the system is controllable at c .

In the following procedure, \mathcal{B} denotes the free family that will eventually become a basis, cnt is the current number of element of that basis. The initial distribution is $(n - r)$ -dimensional at the point c . For an order on \mathcal{H} , we assume that we have an initial order on \mathcal{X} ; then we simply take the order of chronological computation. Finally $\lfloor x \rfloor$ is the integer part (floor function) of the real x .

Procedure *Controllability*(c)

(initialize \mathcal{H}_1)

$\mathcal{H}_1 \leftarrow \mathcal{X}$

$\mathcal{B} \leftarrow \mathcal{X}$

$cnt \leftarrow n - r$

(build \mathcal{H}_2)

For X, Y in \mathcal{H}_1 , $X \prec Y$ **do**

add $[X, Y]$ in \mathcal{H}_2 ;

If $\{[X, Y](c)\} \cup \mathcal{B}(c)$ is a free family ⁽¹⁾

then

add $[X, Y]$ in \mathcal{B}

```

 $cnt \leftarrow cnt + 1$ 
 $i \leftarrow 2$ 
While  $cnt < n$  do
   $i \leftarrow i + 1$ 
  (build  $\mathcal{H}_i$ )
  For  $1 \leq j \leq \lfloor i/2 \rfloor$  do
    For  $X \in \mathcal{H}_j, Y = [U, V] \in \mathcal{H}_{i-j}$  do
      If  $U \preceq X$  (2)
        then
          add  $[X, Y]$  in  $\mathcal{H}_i$ 
          If  $\{[X, Y](c)\} \cup \mathcal{B}(c)$  is a free family (3)
            then
              add  $[X, Y]$  in  $\mathcal{B}$ 
               $cnt \leftarrow cnt + 1$ 

```

One can verify that the procedure builds sets \mathcal{H}_i defining a PH-family. Therefore, it appears clearly that the system is controllable at c if and only if *Controllability*(c) terminates. This also means that the procedure never stops otherwise.

Example Part 1 : For a classical example [8,36], take $\mathcal{X} = \{X_1, X_2\}$. The first 14 elements of the PH-family generated by the procedure (if it does not stop before) are :

\mathcal{H}_1 :	X_1	X_2
\mathcal{H}_2 :	$X_3 = [X_1, X_2]$	
\mathcal{H}_3 :	$X_4 = [X_1, [X_1, X_2]]$	$X_5 = [X_2, [X_1, X_2]]$
\mathcal{H}_4 :	$X_6 = [X_1, [X_1, [X_1, X_2]]]$	$X_7 = [X_2, [X_1, [X_1, X_2]]]$
	$X_8 = [X_2, [X_2, [X_1, X_2]]]$	
\mathcal{H}_5 :	$X_9 = [X_1, [X_1, [X_1, [X_1, X_2]]]]$	$X_{10} = [X_2, [X_1, [X_1, [X_1, X_2]]]]$
	$X_{11} = [X_2, [X_2, [X_1, [X_1, X_2]]]]$	$X_{12} = [X_2, [X_2, [X_2, [X_1, X_2]]]]$
	$X_{13} = [[X_1, X_2], [X_1, [X_1, X_2]]]$	$X_{14} = [[X_1, X_2], [X_2, [X_1, X_2]]]$

Example Part 2 : Consider now a 3-body mobile robot (i.e. a classical two degree of freedom mobile robot with two trailers shown in Figure 1). The configuration space is a 5-dimensional manifold. Let c be a point of coordinates $(x, y, \theta, \varphi_1, \varphi_2)$. The rolling constraints of the three bodies provide 3 nonholonomic links. We can prove (see [5] [44] for details) that the 2-dimensional

associated distribution has the following vector fields as basis⁹ :

$$X_1 = \begin{pmatrix} \cos \theta \\ \sin \theta \\ 0 \\ -\sin \varphi_1 \\ \sin \varphi_1 - \cos \varphi_1 \sin \varphi_2 \end{pmatrix} \quad X_2 = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{pmatrix}.$$

The first elements of a PH-family are displayed in Example Part 1. We can verify that the algorithm stops with $\{X_1, X_2, X_3, X_4, X_6\}$ as a basis for every point c verifying $\varphi_1 \not\equiv \frac{\pi}{2} \pmod{\pi}$. The algorithm stops with $\{X_1, X_2, X_3, X_4, X_9\}$ for the remaining hyperplane¹⁰.

Remark : Finally, the rank condition holds everywhere and we can conclude that the corresponding system is controllable.

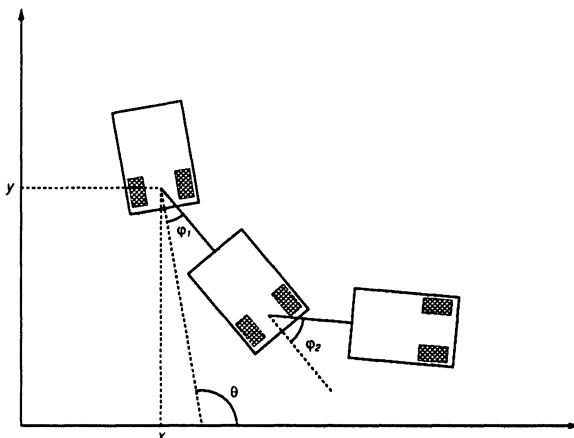


Figure 1: The 3-body mobile robot

In this example, notice that the algorithm checks $6 - 2 = 4$ “candidates” in the first case, and $9 - 2 = 7$ in the second one. What happens in the general case ?

⁹This distribution is computed without reference to any control system. It is built just from the non-sliding hypothesis applied to each body.

¹⁰More precisely : $X_5 = X_1$, $\det\{X_1, X_2, X_3, X_4, X_6\} = -\cos(\varphi_1)$, $X_7 = 0$, $X_8 = -X_3$ and finally $\det\{X_1, X_2, X_3, X_4, X_9\} = -1 - \cos^2(\varphi_1) \cos(\varphi_2)$.

The core of the algorithm is the construction of a PH-family. The dimension n of the manifold being a constant integer of our problem, the only tests needing a subroutine depending on n are (1) and (3). Their complexity is asymptotically negligible. Therefore the worst-case complexity of the algorithm is dominated by the complexity of building a PH-family. The relevant parameter is the value of i when the algorithm stops. Because of the test (2), our procedure for building a PH-family is not optimal¹¹. But, here, we just want to find the minimal complexity of any algorithm that builds a PH-family. Now, the complexity of computing all the elements of a set \mathcal{H}_i is bounded below by the number of all the elements in \mathcal{H}_j , $j \leq i$, and it has been proven that this number is

$$\sum_{1 \leq j \leq i} \alpha(j) \text{ with } \alpha(j) = \frac{1}{j} \sum_{d|j} \mu(d)(n - r)^{\frac{j}{d}}$$

where μ designates the Möbius function.

$$\begin{aligned} \mu : \mathbb{N}^* &\longrightarrow \{-1, 0, 1\} \\ m &\longmapsto \mu(m) = \begin{cases} 0 & \text{if } m \text{ is sq. of a prime integ.} \\ (-1)^k & \text{otherwise, where } k \text{ is the numb.} \\ & \text{of primes dividing } m. \end{cases} \end{aligned}$$

For example, setting $(n - r) = m$, we have $\alpha(1) = m$, $\alpha(2) = \frac{1}{2}(m^2 - m)$, $\alpha(3) = \frac{1}{3}(m^3 - m)$, $\alpha(4) = \frac{1}{4}(m^4 - m^2)$, $\alpha(5) = \frac{1}{5}(m^5 - m)$, $\alpha(6) = \frac{1}{6}(m^6 - m^3 - m^2 + m)$. One may verify the first 5 values on the current example.

If the algorithm runs for a point c and stops with a family \mathcal{H}_i , the system is said to be *completely nonholonomic* at c (i.e., all missing dimensions can be recovered; its degree of nonholonomy is maximal; it is controllable). Besides, its degree of nonholonomy at c is i .

We have to prove this latter result. Indeed, the algorithm above clearly depends upon the basis \mathcal{X} we chose for the distribution Δ . However, the concept of degree of nonholonomy does not. Now, it is a general result from the Lie Algebra Theory that $\bigcup_{j \leq i} \mathcal{H}_j$

¹¹It seems possible to define an optimal one. Maybe such a procedure already exists in the literature ?

constitutes a basis of the nilpotent free Lie algebra $\mathcal{LA}_i(\mathcal{X})$ defined by taking all the brackets of degrees less than i and by killing all the brackets of greater degrees. See [8] for details. Therefore, i does not depend on our choice of a basis \mathcal{X} of Δ . It truly is the degree of nonholonomy that has been previously defined.

Example Part 3 : The degree of nonholonomy of the 3-body mobile robot is 4 at points whose coordinates $(x, y, \theta, \varphi_1, \varphi_2)$ verify $\varphi_1 \not\equiv \frac{\pi}{2} \bmod \pi$. It is 5 elsewhere.

Summing up the results of this section :

The method we use for testing the controllability of a nonholonomic system at a point is at least exponential in the degree of nonholonomy at this point.

Next two sections present classes of systems for which one can stop the procedure *Controllability* after a finite and *known* number of steps.

Regular systems

Let us define the growth vector of a controllable nonholonomic system at a point. This concept appears in [73]. This is a key one for the topological viewpoint that we will adopt in Section 3.

Suppose that the distribution associated to our system is $(n - r)$ -dimensional. Consider a point c and its degree of nonholonomy q_c . The *growth vector* at c is defined as the sequence (n_1, \dots, n_{q_c}) , where $n_1 = (n - r) \leq n_2 \leq \dots \leq n_{q_c} = n$ and n_i is the dimension at c of the linear space generated by combinations of brackets of degree less than i .

Example Part 4 : Let us recall that $\{X_1, X_2, X_3, X_4, X_6\}$ constitutes a basis for points whose coordinates $(x, y, \theta, \varphi_1, \varphi_2)$ verify $\varphi_1 \not\equiv \frac{\pi}{2} \bmod \pi$, while $\{X_1, X_2, X_3, X_4, X_9\}$ works elsewhere. One can verify (by computing the dimension of the linear spaces for

each level) that the growth vector at points verifying $\varphi_1 \not\equiv \frac{\pi}{2} \bmod \pi$ is $(2, 3, 4, 5)$, while it is $(2, 3, 4, 4, 5)$ elsewhere.

All the above tools work only locally. For instance, we have just seen that the growth vector is not the same everywhere in the manifold. The global viewpoint is not easy to reach. A first step is to study what happens in a neighborhood of a point.

A filtration $\{\Delta_i\}$ is *regular* at a point c if the growth vector is constant in a neighborhood of c [73] [72]. This means that all the ranks of $\Delta_i(\cdot)$ are constant in the neighborhood. Otherwise, the filtration is *singular* and the corresponding point is a *singularity*. Most of the problems we encounter when we try to define growth vectors and degrees of nonholonomy derive from the presence of singularities. By extension, we will say that a system is *regular* if the corresponding filtration is regular everywhere.

Example Part 5 : The 3-body system is not regular; more precisely the corresponding filtration is regular at points verifying $\varphi_1 \not\equiv \frac{\pi}{2} \bmod \pi$. It is singular at the remaining points. Remark that the growth vector is strictly increasing for regular points.

For regular systems, the degree of nonholonomy is a constant. It can also be shown (see [63]) that the growth vector is strictly increasing, so the procedure we designed always stops in that particular case.

Nilpotent and Nilpotentizable Systems

Now, let us consider the following case.

Suppose that all the Lie brackets of degree greater than k vanish. In this case, the sequence \mathcal{X}_i stabilizes :

$$\mathcal{X}_1 \subset \mathcal{X}_2 \subset \cdots \subset \mathcal{X}_k = \mathcal{LA}(\mathcal{X}).$$

Such systems are called *nilpotent of order k* (see [8] for a general

definition of the concept in the Lie Algebra framework).

For nilpotent systems of order k , we can stop the procedure *Controllability* as soon as all the Lie brackets of degree k or less are generated. If the procedure does not yield a basis, then the system is not controllable.

Example Part 6 : In our example, we may verify that $[X_2, [X_2, X_1]] = -X_1$. Set $ad_X(Y) = [X, Y]$. Then¹²: $ad_{X_2}^{2m}(X_1) = (-1)^m X_1$. The system is not nilpotent.

In some cases, a non-nilpotent system can be transformed into a nilpotent one via a linear change of controls called a *feedback transformation*. Quite logically, such systems are called *feedback nilpotentizable*. [36] gives some examples of feedback nilpotentizations (e.g., the unicycle, a car-like system and a car-like system with a trailer). See also [25] for sufficient conditions for a system to be nilpotentizable.

Well-Controllability

At this stage of the presentation, let us return to the planning problem. This section introduces the concept of well-controllability. As the regularity concept, it deals with the existence of singularities, but this is a more global one.

As we have seen in Section 2.1, a general idea for devising a nonholonomic motion planner for controllable systems is to define a procedure that searches for an admissible collision-free path, taking any collision-free path as a seed for the search¹³.

Recently Lafferiere and Sussmann proved that this principle is a general one. A collision-free path is first computed without

¹²This example appears in [36] for the unicycle and the car-like robot, i.e., systems equivalent to our current system without trailers (see Section 4).

¹³[39] pinpointed this method for the car-like robot, while [43] presents a planner using this principle for this case.

taking the nonholonomic constraints into account. Lafferriere and Sussmann's method [36] (see Section 3.1.2 for more details) roughly consists of expressing the first holonomic path into some "local coordinate system" (a more precise definition will be given in Section 3.1.1); from these coordinates, because the system is controllable, the authors show that it is possible to explicitly define an admissible control (and then an admissible path) that locally steers the system from a given point (on the first path) to any other on the first path inside a given neighborhood. Because the planner has to work *a priori* everywhere, one has to define a procedure that guarantees to find a local coordinate system *everywhere*. The existence of such a coordinate system is a technical point essential for the method. It is solved by considering an *extended system* associated with the original one; this new system is obtained by adding virtual controls working on vector fields defined from a PH-family of the original system. Since the nonholonomic distribution Δ is $(n - k)$ -dimensional, it seems *a priori* that k additional controls would suffice to make the system holonomic. In fact, in order to avoid singularities (understood as points where the transformation matrix would be non invertible), one has generally to add more controls. Lafferriere and Sussmann note also that additional controls make easier the choice of a transformation matrix with a good condition number.

Let us illustrate this point using our example.

Example Part 7 : Recalling Example Part 2, a local coordinate system defined from $\{X_1, X_2, X_3, X_4, X_6\}$ will encounter singularities. Following Lafferriere and Sussmann's method, a possible extended control system is defined by $\{X_1, X_2, X_3, X_4, X_6, X_9\}$; in the process, four controls are added to the original ones. The previous results show that it is everywhere possible to choose in this family a basis that spans \mathbf{R}^5 .

Now, consider the following family :

$$\begin{array}{lll} U_0 = X_1 & V_0 = X_2 \\ Y_1 = [X_1, X_2] & U_1 = \cos \varphi_1 X_1 + \sin \varphi_1 Y_1 & V_1 = \sin \varphi_1 X_1 - \cos \varphi_1 Y_1 \\ Y_2 = [U_1, V_1] & U_2 = \cos \varphi_2 U_1 + \sin \varphi_2 Y_2 & V_2 = \sin \varphi_2 U_1 - \cos \varphi_2 Y_2 \\ Y_3 = [U_2, V_2] & & \end{array}$$

Using some dedicated software (like Mathematica), it is easy to check that the determinant of $\{V_0, V_1, V_2, U_2, Y_3\}$ is equal to 1. Therefore $\{V_0, V_1, V_2, U_2, Y_3\}$ spans \mathbf{R}^5 *everywhere*¹⁴. According to the previous comments, we can define a *minimal* extended system that never meets with any singularity. Moreover, the transformation matrix has a good condition number. We introduce the concept of a *well-controllable* system.

Definition 1 : An n -dimensional nonholonomic system defined by a distribution Δ is *well-controllable*, if there exists a basis of n vectors fields in the Control Lie Algebra $LA(\Delta)$ such that the determinant of the basis is constant.

Obviously well-controllability implies controllability. The converse does not hold. Indeed, as we mentionned above, a system can be controllable while the local degrees of nonholonomy are unbounded. This means that the filtration $\{\Delta_i\}$ stabilizes locally, but not globally. In this case, it is impossible to define a basis verifying the conditions of our definition.

The well-controllability concept is a global one and it is related to the planning problem. Indeed, for well-controllable systems, the same “local” coordinate system can work everywhere. This simplifies Lafferriere and Sussmann’s planning method. But, though we have a general procedure for testing controllability, we have no general procedure for testing well-controllability. For instance, there is no obvious argument leading to reducing the search of a good basis to a small family, like a PH-Hall family.

Definition 2 : Let \mathcal{B} be a basis of the control Lie Algebra verifying the conditions of Definition 1. The *degree of well-*

¹⁴Be careful : the degree of Y_3 equals 8, when it is viewed as a polynomial function with indeterminates X_1 and X_2 in $\mathcal{LA}(\{X_1, X_2\})$.

controllability of the system is the maximum degree of all the elements of \mathcal{B} .

Remark : If the system is well-controllable, it is obvious that the global degree of nonholonomy is finite.

Example Part 8 : The 3-body mobile robot is a well-controllable nonholonomic system. Its degree of nonholonomy is 5, while its degree of well-controllability is at most 8. [45] gives a proof in the general case of a n -body mobile robot and shows that the degree of well-controllability is at most 2^n in this case.

Geometric Models and Singularities : some Examples

Singularities typically come from the geometry of the system, though in some cases a bad choice of basis might create artificial singularities. Let us illustrate this point with examples.

Two kinds of 2-body systems

Consider the case of the 2-body system. We can verify (see [44]) that this is a regular, well-controllable system of degree 3; its growth vector is (2,3,4) everywhere. Now consider the same convoy, but with another hooking system. Instead of hooking the trailer exactly at the middle of the rear wheels, we hook it to a point behind the rear axle (see Figure 2)¹⁵. Solving the equations yields the basis :

$$\begin{pmatrix} a' \cos \theta \\ a' \sin \theta \\ 0 \\ -\sin \varphi \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} (a' + a \cdot \cos \varphi) \cos \theta \\ (a' + a \cdot \cos \varphi) \sin \theta \\ \sin \varphi \\ 0 \end{pmatrix}.$$

¹⁵[42] gives a constructive proof of controllability for this system.

If we consider this basis only, we find a growth vector of $(2, 3, 4)$ at points verifying $\sin \varphi \neq 0$. The points verifying $\sin \varphi = 0$ seem to constitute a singularity. But, checking our provisional basis, we find that the vector fields are collinear at these points, though the distribution is regular at these points too. It is just awkward to find two vector fields which stay independent everywhere. For the record, any vector field of the form

$$X = k \begin{pmatrix} a' \cos \theta \\ a' \sin \theta \\ 0 \\ -\sin \varphi \end{pmatrix} + l \begin{pmatrix} (a' + a \cdot \cos \varphi) \cos \theta \\ (a' + a \cdot \cos \varphi) \sin \theta \\ \sin \varphi \\ 0 \end{pmatrix} + m \begin{pmatrix} 0 \\ 0 \\ a' \\ a \cos \varphi + a' \end{pmatrix}$$

is a valid vector field. With some ingenuity, for any value of a and a' , it is possible to obtain a global basis.

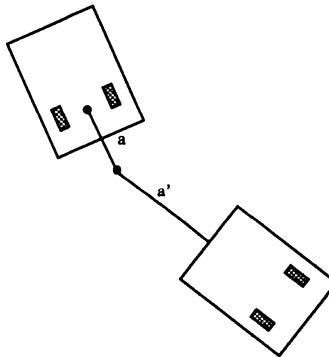


Figure 2: Another 2-body system

A stranger case

Now let us pinpoint a stranger case. We have seen that the 3-body convoy we study is controllable with degree 5. We can verify that the 2-body convoy is controllable with degree 3 (see [44]). The 1-body convoy (i.e., the unicycle) is controllable with degree 2. Because of the singularity we mentioned, there is no 4th degree. What is the fundamental difference between one trailer and two trailers ?

The n -body system

It is proved in [44] that a n -body convoy is well-controllable and that its nonholonomy degree is at most 2^n . What is its precise nonholonomy degree ? What kind of singularities do we stumble upon ?

Answering these questions requires a general framework dealing with the study of the dimension and the number of connected components of singular subsets. To the extent of the author's knowledge, this study is still open.

An example with unbounded degree of nonholonomy

Finally, our last example does not model any robotics systems whatsoever¹⁶. Rather, it tries to capture the flavor of problems encountered in practice without any tedious computations: for instance, the example of the n -body system is quite similar to this one. Let us work in Euclidean space (x, y, z) . Choose an integer n and consider the following vector fields :

$$X = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}, \quad Y_0 = \begin{pmatrix} 1 + z^n \\ 1 + z^{2n} \\ 0 \end{pmatrix}$$

and the distribution they engender. Computing the associated filtration obviously sums up to computing

$$Y_k = \begin{pmatrix} (z^n)^{(k)} \\ (z^{2n})^{(k)} \\ 0 \end{pmatrix}.$$

Now, this system is regular almost everywhere but not everywhere : the point $(0, 0, 0)$ is a major inconvenience. At that point, for any k less than n , the vector field Y_k vanishes altogether, so that the growth vector at $(0, 0, 0)$ is $(2, 2, \dots, 3)$, giving a generic

¹⁶This example was suggested by Marc Espie.

case where the nonholonomy degree is arbitrarily high. Furthermore, using standard techniques (Partitions of Unity), it is easy to piece together a denumerable infinity of such singular patches, so that the resulting distribution has an unbounded degree.

This is the typical case where our algorithm will not terminate. A finer study of the problem would be a tremendous help.

The Complete Problem

In Section 2 we saw that the existence of an admissible collision-free path for a controllable nonholonomic system is characterized by the existence of a collision-free path for the associated holonomic system. Planning a path can thus be achieved through the following steps :

- using a geometric planner for finding a path without taking into account nonholonomic constraints¹⁷, and then
- steering the system along an admissible path as close to the first one as required by the obstacles.

Such a general strategy has been refined into two different approaches that we will examine in Sections 3.3. The first one [67] uses a constructive proof of local controllability founded on fundamental tools of differential geometry introduced in the next section (Section 3.1). The second one [32] uses an explicit form for canonical feasible paths (e.g., shortest paths).

Both approaches come up against the following key problem : how to guarantee that the admissible path lies as close to the

¹⁷Finding such a path corresponds to the classical Piano Mover problem. This problem is decidable. From a theoretical point of view, general algorithms exist in the literature. Notwithstanding, at this time, there is no general software that runs efficiently in practice. This is due to the intrinsic combinatorial complexity of the problem.

steering path as the obstacles require ? The topological nature of this question is discussed in Section 3.2.

Finally, we show in Section 3.4 how the complexity analysis of the nonholonomic path finding problem has to be revisited by reference to the classical piano mover problem.

Motion Planning without Obstacles

From Vector Fields to Paths

In this subsection we will investigate the delicate problem of finding paths that are compatible with our nonholonomic constraints. We need some precise concepts of differential geometry, which are thoroughly studied in [62] [72] [63] [64].

Choose a point p in our manifold and a vector field X defined around this point. There is exactly one path $\gamma(t)$ starting at this point and following X . In a formal notation, it verifies $\gamma(0) = p$ and $\dot{\gamma}(t) = X_{\gamma(t)}$.

One defines the exponential of X (denoted by e^X) to be the point $\gamma(1)$. This gives a correspondence between the space of vector fields and a neighborhood of p . It is obvious that one has $e^{tX} = \gamma(t)$, namely this definition doesn't describe a peculiar point of a path but, more accurately, links every point of the path to a specific vector field. Let us translate our previous example to this formalism. Following $aX + bY$ for a given time (the unit time) simply means to take e^{aX+bY} . Following X for a time a amounts to following aX for the unit time, that is taking e^{aX} , and following Y for a time b is the same as taking e^{bY} . This is still a slightly different point of view : instead of considering the exponential to define a specific point of a path with regard to an origin point p , we understand it as describing a motion from a point to another on a given path. Thus, starting at the origin o , following aX for a given time, then bY leaves us at the point $e^{bY} \cdot e^{aX} \cdot o$. Therefore

the exponential of a vector field X appears as an operation on the manifold, meaning “slide from the given point along the vector field X for unit time.”

In that setting, everything works nearly as smoothly as in the Euclidean case, at least locally. The main difference is that, whenever $[X, Y] \neq 0$, following directly $aX + bY$ or following first aX then bY are no longer equivalent. Intuitively, $[X, Y]$ measures the variation of Y along the paths of X ; in other words, the field Y we follow in $aX + bY$ has not the same value as the field Y we follow after having followed aX (indeed Y is not evaluated at the same points in both cases). The main result is the following :

Assume that X_1, \dots, X_n are vector fields defined in a neighborhood U of a point p such that at each point of U , X_1, \dots, X_n constitutes a basis of the tangent space. Then there is a smaller neighborhood V of p on which the correspondences $(a_1, \dots, a_n) \mapsto e^{a_1 X_1 + \dots + a_n X_n} \cdot p$ and $(a_1, \dots, a_n) \mapsto e^{a_n X_n} \cdots e^{a_1 X_1} \cdot p$ are two coordinate systems, called the first and the second normal coordinate system associated to $\{X_1, \dots, X_n\}$.

The Campbell-Haussdorf-Baker-Dynkin formula states precisely the difference between the two systems :

For a sufficiently small t , one has : $e^{tX} \cdot e^{tY} = e^{tX + tY - \frac{1}{2}t^2[X, Y] + t^2\epsilon(t)}$, where $\epsilon(t) \rightarrow 0$ when $t \rightarrow 0$.

Actually, the whole Campbell-Haussdorf-Baker-Dynkin formula as proved in [72] gives an explicit form for the ϵ function. More precisely, ϵ yields a formal series whose coefficients lie in $\mathcal{LA}(\{X, Y\})$: the coefficient of t^k is a combination of brackets of degree k .

Roughly speaking, the Campbell-Haussdorf-Baker-Dynkin formula tells us how a nonholonomic system can reach *any* point in a neighborhood of a starting point. This formula is the hard core of the local controllability concept. It yields a method for *explicitly computing paths* in a neighborhood of a point.

On the basis choice

In the case of a nilpotent system of order k , since brackets of degrees greater than k vanish, the Campbell-Haussdorf-Baker-Dynkin gives an *exact* development of the exponential. This property is used by Lafferiere and Sussmann in [36].

For nilpotent systems we have seen that it is possible to compute a basis \mathcal{B} of the Control Lie Algebra $LA(\Delta)$ from a Philipp Hall family. Lafferiere and Sussmann's method assumes that a holonomic path γ is given. If we express locally this path on \mathcal{B} , i.e., if we write the tangent vector $\dot{\gamma}(t)$ as a linear combination of vectors in $\mathcal{B}(\gamma(t))$, the resulting coefficients define a control that steers the holonomic system along γ . Using the Campbell-Haussdorf-Baker-Dynkin formula, it is then possible to compute an admissible and piecewise constant control u for the nonholonomic system that steers the system *exactly* to the goal (indeed, since all the brackets vanish after a given level k , the Campbell-Haussdorf-Baker-Dynkin formula gives an *exact* development of the exponential on brackets of degree less than k , so the synthetized path ends exactly at the same point).

For a general system, Lafferiere and Sussmann reason as if the system were nilpotent of order k . In this case, the synthetized path deviates from the goal. Nevertheless, thanks to a topological property, this basic method is used in an iterated algorithm that produces a path ending as close to the goal as wanted.

In [26], Jacob give an account of Lafferiere and Sussmann's strategy by using another coordinate system. This system is built from a Lyndon basis of the free Lie algebra [74] instead of a P. Hall basis. This choice reduces the number of pieces of the solution.

Remark : Lafferiere and Sussmann's strategy can be extended to well-controllable systems by replacing the Philip Hall local coordinate system by another one built from any basis working everywhere. In this case the method could solve the singularity problem which has not been considered at this time. Neverthe-

less, because the degree of vector fields of a “good” basis is at least higher than the degree of the vector fields of a PHall-family, the computational cost due to the development of the Campbell-Haussdorf-Baker-Dynkin formula will be more expensive. This extension is currently under study.

Steering by sinusoids

At the same time as Lafferiere and Sussmann work, Murray and Sastry explored in [50,51] the use of sinusoidal inputs in steering certain classes of nonholonomic systems : the class of systems which can be converted into a triangular form. A triangular form is :

$$\begin{aligned}\dot{x}_1 &= v \\ \dot{x}_2 &= f_2(x_1)v \\ \dot{x}_3 &= f_3(x_1, x_2)v \\ &\vdots \\ \dot{x}_p &= f_p(x_1, \dots, x_p)v\end{aligned}$$

with $x_i \in \mathbf{R}^{m_i}$ and $\sum_i m_i = n$.

Because of this special form, there exists simple sinusoidal control that may be used for generating motions affecting the i^{th} set of coordinates while leaving the previous sets of coordinates unchanged. The algorithm then is :

1. Steer x_1 to the desired value using any input and ignoring the evolutions of the x_i 's ($1 < i$),
2. Using sinusoids at integrally related frequencies, iteratively find the inputs steering the x_i 's without changing the x_j 's, $j < i$.

Even if a system is not triangular, it may be possible to transform it into a triangular one by feedback transformations (see [51] for details and [70] for recent extensions).

Obstacle Avoidance : the Topological Question

Now we take a closer look at the problem of obstacle avoidance.

What Kinds of Topologies ? An Informal Statement

Let us come back to the sources. Motion planning in Robotics deals with obstacle avoidance. Real obstacles get transformed into “obstacles” in the configuration space. The Hausdorff metric¹⁸ on the bodies, which are closed compact subsets, in the environment (that is to say, 3-dimensional Euclidean space) induces a metric in the configuration space [76]. Therefore, the reference open sets in the configuration space are the open sets in the topology induced by the Hausdorff metric in Euclidean space. This is the topology needed for solving placement problems¹⁹. A path appears as a continuous function from a closed interval of \mathbf{R} to the configuration space equipped with this topology.

In some cases, the very act of considering motions can lead to a finer topology. If we introduce the distance induced by the best path(ies) between two points with respect to a given cost (length, energy, time taken, etc), differential considerations take the scene. Consider the case of energy for instance. For holonomic systems, since every smooth path in the configuration space is an

¹⁸A Hausdorff metric d_H can be defined for any space equipped with a distance d . It yields the following topology on compact subsets :

$$d_H(A, B) = \inf_{a \in A} (\sup_{b \in B} d(a, b)).$$

¹⁹See the problem of cloth or leather cutting as an example of such a problem in the context of Computational Geometry.

admissible path, this cost induces a natural Riemannian distance. In that case, the induced topology remains the same.

For general nonholonomic systems, there may exists points at an infinite distance of each other. The non-holonomic constraints partition the configuration space into disconnected submanifolds, and the resulting topology has little resemblance to the natural one. However, for controllable nonholonomic systems, any two points can be joined by an admissible path; considering the best path leads once again to a well-defined metric. This is the origin of sub-Riemannian geometry. Recent contributions show that Riemannian and sub-Riemannian metrics are equivalent. Therefore both topologies are the same. The next section state these results more thoroughly.

Of Sub-Riemannian Metrics, Shortest Paths and Geodesics

This section makes use of the ideas given in [22] [73] for the case of regular systems. Consider a controllable nonholonomic system (i.e., a completely nonholonomic one) defined on a n -dimensional manifold CM (for “Configuration Manifold”) by a distribution Δ . The nonholonomic metric²⁰[73] is defined by

$$\rho_\Delta(c, c') = \inf_{\gamma \in S(c, c')} \int_0^1 \langle \dot{\gamma}(t), \dot{\gamma}(t) \rangle^{\frac{1}{2}} dt,$$

where

$$S(c, c') = \{\gamma \mid [0, 1] \rightarrow CM, \gamma(0) = c, \gamma(1) = c', \dot{\gamma} \in \Delta\}.$$

In that setting, *geodesics* are admissible paths that locally solve the variational problem.

The proof of the equivalence of the Riemannian and the sub-Riemannian metric resides in a two-sided estimate of the size of sub-Riemannian balls. Denote by $B_\epsilon(c)$ the sub-Riemannian ϵ -ball, i.e., the set of points reachable from c by an admissible path

²⁰This metric is also known as a *singular* [9], a *Carnot-Caratheodory* [48], or a *sub-Riemannian* [63] metric.

of length less than ϵ . Assume that the system is *regular* at c (see Section 4.8). Let $\{c_i\}$ be the local coordinate system defined in Section 6.1. Now consider the parallelepiped $P_{\alpha,\epsilon}(c) = \{c' \in CM \mid |c_i(c')| \leq \alpha\epsilon^{\varphi(i)}\}$ where $\varphi(i)$ is derived from the growth vector $\{n_1, \dots, n_{q_c}\}$ of Δ at c as $\varphi(i) = j$ for $n_{j-1} < i \leq n_j$ ²¹. A two sided-estimate of $B_\epsilon(c)$ is given by the *parallelepiped theorem* [73]²² : there are positive constants $\alpha_1, \alpha_2, \epsilon_0$, such that for $\epsilon < \epsilon_0$,

$$P_{\alpha_1,\epsilon}(c) \subset B_\epsilon(c) \subset P_{\alpha_2,\epsilon}(c).$$

Therefore, for regular systems, Riemannian and sub-Riemannian topologies are the same.

Remark : The parallelepiped theorem holds only at regular points. There are some technical problems at singular points (e.g., as at the singularities appearing in the 2-body system). It seems possible to extend the proofs (using a local coordinate system, i.e., valid everywhere in a neighborhood of a regular point) to well-controllable systems by using a local coordinate system holding for singular points as well as for regular points (such a system exists by definition). Nevertheless, as always with singularities, some care will have to be taken.

Strange phenomena appear in this sub-Riemannian geometry framework. We know that, in general, if shortest paths are geodesics, geodesics are not necessarily shortest paths—indeed, they only minimize length *locally*. One of the main features of Riemannian geometry is that, locally, geodesics *are* shortest paths. Consider the ϵ -ball $B_\epsilon(c)$ above, and $S_\epsilon(c)$ its boundary sphere. In Riemannian geometry, for ϵ sufficiently small, the sphere S_ϵ is in one to one correspondence with the ends of geodesics of length ϵ (so-called

²¹**Example Part 9 :** For the 2-trailer convoy at $c = (0, 0, 0, 0, 0)$, the growth vector is $(2, 3, 4, 5)$ (see Example Part 4). Therefore :

$$P_{\alpha,\epsilon}(c) = \{(x, y, \theta, \varphi_1, \varphi_2) \mid |x| \leq \alpha\epsilon, |y| \leq \alpha\epsilon, |\theta| \leq \alpha\epsilon^2, |\varphi_1| \leq \alpha\epsilon^3, |\varphi_2| \leq \alpha\epsilon^4\}$$

²²A proof using different terminology appears in [63].

wave front). No similar property does hold for nonholonomic systems. [73] holds two drawings illustrating the strange relationship between the spheres and the wave fronts in the Heisenberg group case.

Another strange phenomenon can be illustrated by the study of the car-like system. In this particular case, the shortest paths have an explicit form (see the following section). Figure 3 shows pictures of one of the corresponding spheres (it appears clearly that the sphere is not smooth).

Geodesics and Shortest Paths : Elementary Computational Aspects

The classical way for computing geodesics is to use the maximum principle of Pontryagin [55]. This is a powerful tool from classical Optimal Control Theory, which provides necessary conditions for the existence of an optimal control. The bang-bang principle is its direct consequence [2]. Under some hypotheses, this principle gives the form of the optimal controls, if they exist. It has been applied in [27] for mobile robots with two driving wheels : in this case, geodesics are piecewise clothoids or anticrothoids. Using the same ideas, we may verify that the geodesics for car-like robots are piecewise arcs of circle or straight lines.

How to compute shortest paths is a much more difficult problem, even when an explicit form for the geodesics (local shortest paths) is known. The problem is basically a combinatorial one : how to piece smooth geodesic parts together in order to produce a shortest path ? As far as the author knows, this question has only been answered for the car-like robot by Reeds and Shepp [56]. They extend the work of Dubins on the form of smooth shortest paths [18] and they establish precisely which combinations of arcs of circle and straight line segments can produce shortest paths. Since the number of used combinations is finite, this gives birth to an efficient method to compute shortest paths.

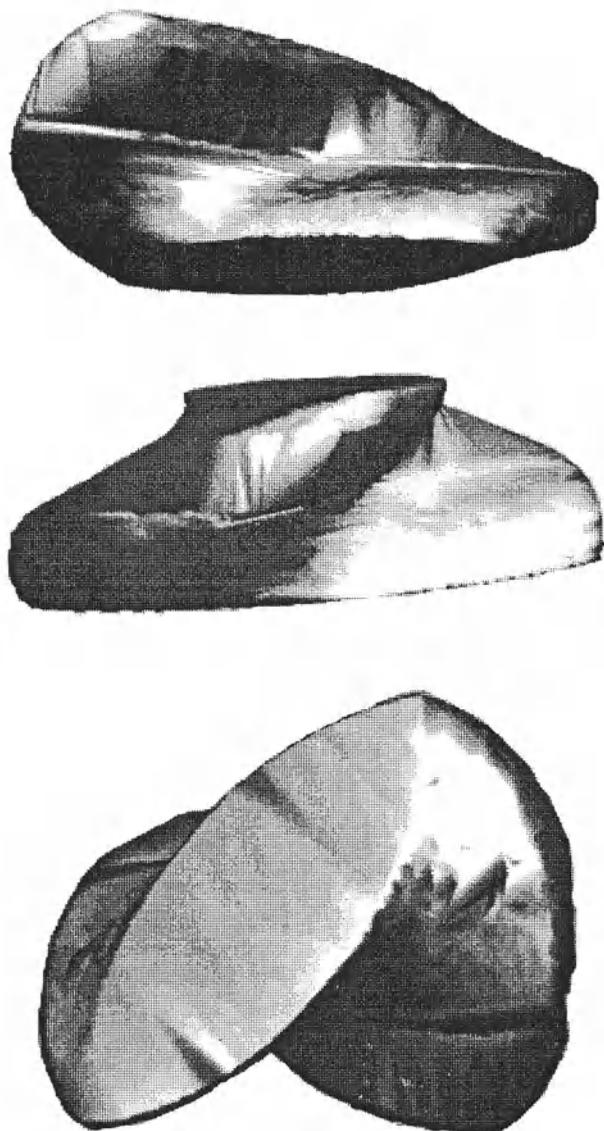


Figure 3: A sphere for the car-like system.

Very recently Sussmann and Tang [68] and Boissonnat, Cerezo and Leblong [7] give independently a new proof of Reeds and Shepp's result by using the maximum principle. It appears that the necessary conditions of the maximum principle, together with few geometric arguments dedicated to this special case, permit to compute not only the form of the geodesics, but also the switches on the controls. [68] shows how the proof can be developed in a general framework which opens a promising way for computing shortest paths.

Holonomic Path Approximation Methods

All the needed material being introduced, we now survey two strategies for planning nonholonomic paths that avoid obstacles. Both strategies consist in computing a collision-free holonomic path and then approximating this path by an admissible one.

A Strategy Using Philipp Hall Coordinate Systems and Oscillatory Controls

The strategy developed by Sussmann and Liu [67] starts with the previous work by Lafferriere and Sussmann [36] who did not address the obstacle avoidance problem (see above). For any nonholonomic system

$$\dot{x} = \sum_{k=1}^m u_k f_k(x)$$

the authors consider an extended system

$$\dot{x} = \sum_{k=1}^r v_k f_k(x)$$

where the $(r - m)$ new vector fields are elements of a Philipp Hall family built from the original f_i 's. They prove sufficient conditions for a sequence of inputs u^j to be such that the corresponding paths converge to those of the extended system. Using these conditions, it is possible to solve the inverse problem : find paths

of the original system that converge to any path of the extended one. The extended system being holonomic, one first computes a holonomic collision-free path. Then one computes a converging sequence of nonholonomic paths. Since this sequence converges to a collision-free path, it contains nonholonomic collision-free paths.

The basic idea for building the sequence u^j is to use sinusoids, like in Murray and Sastry's strategy that works for triangular systems. The triangular form allows sinusoid controls to iteratively steer each variable without changing the previous ones (see above). In the general context addressed by Sussmann and Liu, the difficulty is to treat separately each bracket appearing in the extended system. The solution is obtained by considering for each bracket a sinusoidal input with a frequency which must not interfere with the others. The combinatorial study of the non-interfering conditions is clever (and hard to understand...).

In [70], simulation results obtained from that kind of strategy are presented for the case a 3-body car-like system. This reference pinpoints the merits of the basis, relative to the convergence properties (indeed, the general method can work with other basis than the P. Hall basis).

Finally, this strategy requires to integrate a collision checker in order to evaluate the allowed “gap” between the holonomic path and the nonholonomic one extracted from the sequence. This gap is related to the size of the free-space around the reference path. The study of this integration has still to be done.

A Planner Using Shortest Paths

The planner developed by Jacobs, Laumond and Taïx [32] has been developed in the framework of the Hilare mobile robot project of LAAS (Figure 4; see [23,14]). It uses a shortest paths approach. The three steps of the algorithm are as follows :

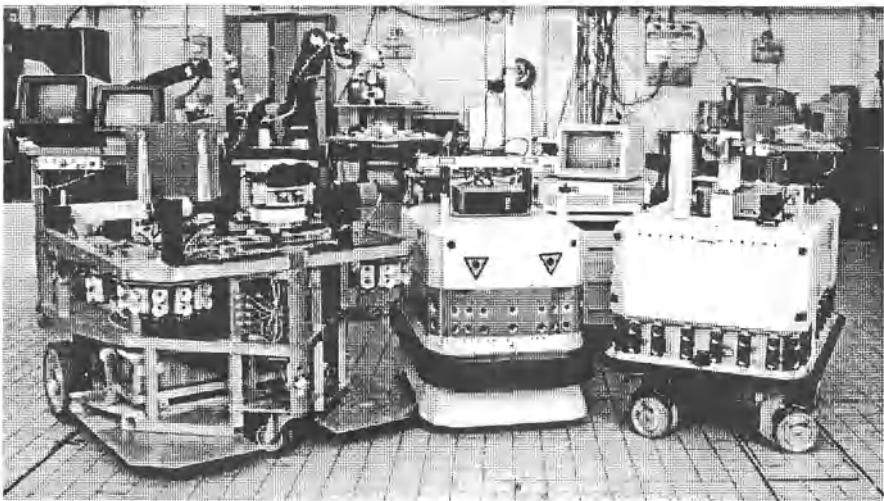


Figure 4: The Hilare Family

- Plan a path γ for the corresponding holonomic system. If one does not exist, then no feasible path exists.
- Subdivide γ until all endpoints can be linked by a minimal length *collision-free* feasible path.
- Run through an “optimization” routine to reduce the length of the path.

The convergence of the algorithm is a consequence of the parallelepiped theorem. Indeed, consider a point c on γ and \mathcal{N}_c a neighborhood of c included in the *collision-free* configuration space. The parallelepiped theorem guarantees that there is a neighborhood \mathcal{N}'_c of c , such that for each point $c' \in \mathcal{N}'_c$ the shortest path between c and c' lies in \mathcal{N}_c . Therefore since γ can be covered by a finite number of such neighborhoods \mathcal{N}_c , the subdivision procedure will stop.

This planner has been totally implemented in an exact version for the special case of a polygonal car-like robot. This means we had to implement a geometric planner for computing an exact representation of the collision-free configuration space. To do this, we used Avnaim and Boissonnat’s algorithm [3]. We also

implemented a procedure for computing the shortest paths in the absence of obstacles based upon Reeds and Shepp's work. Finally, we designed a fast collision checking procedure. Figure 5, an excerpt from [69], shows how the algorithm responds to the classical example of the parking problem. The drawings give the paths produced by the three steps of the algorithm²³.

An advantage of such a strategy is that it optimizes locally the path (in terms of the path length). Indeed, the third step finds a quasi-optimal solution. Notice that finding the optimal one is a problem known to be very difficult, and computationally very complex.

The main drawback of this general strategy is that we need to compute the shortest paths (see Section 3.2). We are working on the Hilare-like system : at this time we know an explicit form for the geodesics [27], but not for the shortest paths.

Notwithstanding, using general mathematical techniques for proofs of controllability and complexity analysis, we have laid a theoretical basis for a study of systems of greater complexity. Essentially, the method consists in establishing a catalogue of canonical paths having the necessary topological properties, then in using them together with the subdivision motion planning technique. Clearly, the main question is : how to compute a sufficient set of canonical paths ? In the general case, the only way out seems to be the use of discretization techniques, but even in this case, the theoretical background is required to find a discretization fine enough to solve the problem. This latter aspect is currently under investigation.

Complexity of the Complete Problem

As discussed in Section 2, the *decision* part of the motion planning problem for controllable systems is equivalent to the decision

²³In [38], Latombe presents another implementation of this planner based on a "bitmap" representation of the configuration space.

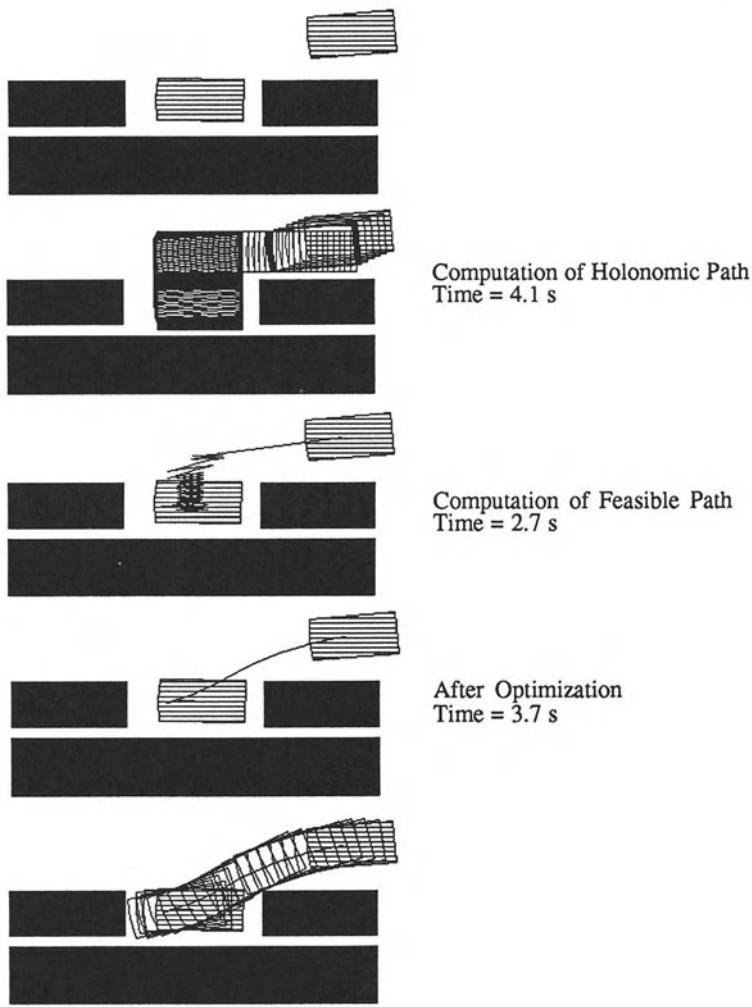


Figure 5: The three steps of the algorithm.

problem for the associated holonomic system. Hence the complexity of this problem is a polynomial function of the *complexity of the environment* (i.e., the number of geometric primitives required to describe it), and the classical algorithms for the piano movers problem can be applied.

Notwithstanding, the complexity of the *complete* problem (i.e., actually producing a path) is more difficult to grasp. As a rule, any measure of complexity for this problem ought to be bounded from below by the complexity of the solution, which in turn ought to account for the number of singular points (cusps, loops,...) it contains. Unfortunately, this number bears no relationship to the customary description of the input data used for the piano movers problem. It actually depends on the inner size of the free space, measured in the sub-Riemannian setting adapted to the problem.

Before giving a formal definition of the complexity of a path, as it appears in [6], we will first introduce the concept for a continuous function $f: [a, b] \rightarrow \mathbf{R}$. A good measure of what we call the *geometric complexity* of such a function is the number of changes of variation. More precisely, this complexity is defined as the quotient of the total variation of the function (see [57]) by the amplitude of the function.

In the current context, our paths depend on the associated controls, which are continuous real valued functions. Consequently, we define the complexity of a path according to the complexity of the associated controls. If a path γ is defined by a control function u , its geometric complexity is defined by :

$$\sum_i \#(\text{sign changes of } u_i)$$

(see [6] for details).

This definition grasps every critical point of a path, and so matches closely the output complexity of any motion planner. In the current context, it also includes the number of maneuvers as well as the number inflexion points for instance.

This definition hints at some lower bounds for the complexity of complete motion planners. For instance [6] solves the case of the car-like robot. Given c and c' two configurations in the same connected component of configuration space, we proceed to define ϵ . Choose an admissible path between c and c' . At a point q of that path, find the Euclidean diameter of the largest ball centered at q and wholly contained in the free configuration space. Take the infimum of this diameter for all points of the path. Then ϵ is defined as the supremum of this quantity over all possible paths linking c and c' .

Then the complexity of a path of a car-like robot going between c and c' is proportional to $O(\epsilon^{-2})$.

Roughly speaking, this means that, in the car parking problem (as shown in Figure 5), the number of necessary maneuvers asymptotically varies as the square of the inverse of the margin of maneuver.

The proof of this result is based upon the Campbell-Haussdorff-Baker-Dynkin development of the exponential. A detailed study of the general case exhibits very close relationship between this complexity model and the degree of nonholonomy of the system. Everything depend upon the shape of the parallelepipeds derived from the growth vectors. A general proof, currently under study, should lead to the following property :

Property : Given c, c' two configurations in the same arcwise-connected component of the configuration space. Define ϵ as the supremum, over all continuous paths from c to c' , of the infimum, over all configurations q along that path, of the Euclidean diameter of the largest ball centered at q and wholly contained in the free configuration space. The problem of finding the path for a *regular* controllable nonholonomic system of degree k between c and c' is in $\Omega(\epsilon^{-k})$.

If this conjecture holds, this would mean that **a luggage carrier driver would have to do $\Omega(\epsilon^{-n})$ maneuvers in order**

to park his n -body machine in the neighborhood of regular points in an airport.... How many maneuvers are required in the neighborhood of singular points ?...

This last question deals with the shape of the sub-Riemannian balls around singularities. This is an open one.

Acknowledgments

In this study, I have benefited from an advanced preliminary collaboration with Paul Jacobs and Berkeley's group. I thank Marc Espie for rummaging with me through Stanford's libraries, and for a critical reading of the initial report [44]. I also thank Jean-Claude Latombe for several useful discussions on the controllability of a multibody system and for his invitation to stay in his laboratory. I wish also to thank André Bellaïche for introducing me to the sub-Riemannian Geometry, Shankar S. Sastry and Hector Sussmann for many discussions, and Richard Murray for pinpointing the reference [73]. Finally, thanks to Georges Giralt who has felt the fecundity of the nonholonomic motion planning problem a few years ago, and presented it to me.

References

- [1] B. d'Andréa-Novel, G. Bastin and G. Campion. Modelling and control on non-holonomic wheeled mobile robots. In *Proceedings of the 1991 IEEE Int. Conf. on Robotics and Automation*, Sacramento, April 1991.
- [2] M. Athans, and P.L. Falb. *Optimal Control*. McGraw-Hill, 1966.
- [3] F. Avnaim, J.D. Boissonnat, and B. Faverjon. A practical exact motion planning algorithm for polygonal objects amidst polygonal obstacles. In *IEEE International Conference on Robotics and Automation, Philadelphia (USA)*, pages 1656–1661, April 1988.
- [4] J. Barraquand and J.C. Latombe. On non-holonomic mobile robots and optimal maneuvering. *Revue d'Intelligence Artificielle*, 3(2), 1989.
- [5] J. Barraquand and J.C. Latombe. Controllability of mobile robots with kinematic constraints. *Technical Report STAN-CS-90-1317*, Stanford University, June 1990.

- [6] A. Bellaiche, J.P. Laumond and P. Jacobs Controllability of car-like robots and complexity of the motion planning problem with kinematic constraints. *Int. Symp. on Intelligent Robotics*, Bengalore, January 1991.
- [7] J.D. Boissonnat, A. Cerezo and J. Leblong. Shortest paths of bounded curvature in the plane. INRIA Report, Sophia Antipolis, France, July 1991.
- [8] N. Bourbaki. *Groupes et Algèbres de Lie : chapitres 2 et 3* CCLS Diffusion, Paris, 1972.
- [9] R.W. Brockett. Control theory and singular Riemannian geometry. In *New Directions in Applied Mathematics*, Springer-Verlag, 1981.
- [10] R.W. Brockett. Asymptotic stability and feedback stabilization. In *Differential Geometric Control Theory*, Brockett, Millman, Sussmann Eds, Birkhäuser, 1983.
- [11] John Canny, Bruce Donald, John Reif, and Patrick Xavier. On the complexity of kinodynamic planning. In *29th Symposium on the Foundations of Computer Science*, pages 306–316, White Plains, NY., October 1988.
- [12] J. Canny, A. Rege, and J. Reif. An exact algorithm for kinodynamic planning in the plane. *6th ACM Symp. on Computational Geometry*, Berkeley, 1990.
- [13] C. Canudas de Wit and R. Roskam. Path following of a 2-DOF wheeled mobile robot under path and input torque constraints. In *Proceedings of the 1991 IEEE Int. Conf. on Robotics and Automation*, Sacramento, April 1991.
- [14] R. Chatila. Mobile Robot navigation: space modeling and decisional processes. In *Robotics Research : The Third International Symposium, O. Faugeras and G. Giralt (Eds), MIT Press, Cambridge, Massachusetts*, 1986.
- [15] P. Calou, W. Choi, J.C. Latombe, C. Lepape and M. Yim. Indoor automation with many mobile robots. In *IEEE IROS*, 1989.

- [16] W.L. Chow. Über systeme von linearen partiellen Differentialgleichungen erster ordnung. *Math. Ann.*, 117, pp 98-105, 1939.
- [17] C. A. Desoer and Y. T. Wang. Linear-time-invariant servomechanism problem: A self-contained exposition. In C. T. Leondes, editor, *Control and Dynamic Systems*, pages 81–129. Academic, New York, 1980.
- [18] L. E. Dubins. On curves of minimal length with a constraint on average curvature and with prescribed initial and terminal positions and tangents. *American Journal of Mathematics*, 79:497–516, 1957.
- [19] C. Fernandes, L. Gurvits and Z.X. Li. A variational approach to optimal nonholonomic motion planning. In *Proceedings of the 1991 IEEE Int. Conf. on Robotics and Automation*, Sacramento, April 1991.
- [20] S.J. Fortune and G.T. Wilfong. Planning constrained motions. In *ACM STOCS*, pages 445–459, Chicago IL, May 1988.
- [21] T. Fraichard, C. Laugier, G. Lievin. Robot motion planning : the case of nonholonomic mobiles in a dynamic world. '90 IROS, Tsuchiura, Japan, July 1990.
- [22] V.Y. Gershkovich. Two-sided estimates of metrics generated by absolutely nonholonomic distributions on Riemannian manifolds. *Soviet. Math. Dokl.*, 30 (2), 1984.
- [23] G. Giralt, R. Chatila, and M. Vaissset. An integrated navigation and motion control system for autonomous multisensory mobile robots. In *Robotics Research : The First International Symposium, M. Brady and R. P. Paul (Eds), MIT Press, Cambridge, Massachusetts*, 1984.
- [24] Hermann and Krener. Nonlinear controllability and observability. *IEEE Transactions on Automatic Control*, 22(5), 1977.

- [25] H. Hermes, A. Lundell, and D. Sullivan. Nilpotent bases for distributions and control systems. *J. of Differential Equations*, 55, 385-400, 1984.
- [26] G. Jacob. Lyndon discretization and exact motion planning. In *Proceedings of European Control Conference*, Grenoble, France, 1991.
- [27] P. Jacobs, A. Rege, and J.P. Laumond. Non-holonomic motion planning for Hilare-like mobile robots. In *Int. Symp. on Intelligent Robotics*, Bengalore, January 1991.
- [28] P. Jacobs. Minimal length curvature constrained paths in the presence of obstacles. LAAS/CNRS Report 90042, Laboratoire d'Automatique et d'Analyse des Systemes, February 1990.
- [29] P. Jacobs and J. Canny. Planning smooth paths for mobile robots. In *Proceedings of the 1989 International Conference on Robotics and Automation*, pages 2–7. IEEE, May 1989.
- [30] P. Jacobs and J. Canny. Robust motion planning for mobile robots. In *Proceedings of the 1990 International Conference on Robotics and Automation*, Cincinnati, Ohio, May 1990. IEEE.
- [31] P. Jacobs, G. Heinzinger, J. Canny and B. Paden. Planning guaranteed near-time-optimal trajectories for a manipulator in a cluttered workspace. Tech. Rep. RAMP 89-15, University of California, Berkeley, Engineering Systems Research Center, September 1989.
- [32] P. Jacobs, J.P. Laumond and M. Taïx. Efficient motion planners for nonholonomic mobile robots. In *Proceedings of '91 IEEE IROS*, Osaka, November 1991.
- [33] K. Kedem and M. Sharir. An efficient motion-planning algorithm for a convex polygonal object in two-dimensional polygonal space. *Discrete and Computational Geometry*, 5:43–75, 1990.

- [34] O. Khatib. Real-time obstacle avoidance for manipulators and mobile robots. *International Journal of Robotics Research*, 5(1):90–99, Spring 1986.
- [35] B.H. Krogh and D. Feng. Dynamic generation of subgoals for autonomous mobile robots using local feedback information. *Transactions on Automatic Control*, 34(5):483–493, May 1989.
- [36] G. Lafferriere and H.J. Sussmann. Motion planning for controllable systems without drift : a preliminary report. *Technical Report SYCON-90-04*, Rutgers Univ., June 1990.
- [37] J.C Latombe. Robot Motion Planning. *Kluwer Pub.*, 1991.
- [38] J.C. Latombe. A fast path planner for a car-like indoor mobile robot. In *Proceedings of 9th National Conf. on Artificial Intelligence, AAAI*, Anaheim (CA), July 1991.
- [39] J.P. Laumond. Feasible trajectories for mobile robots with kinematic and environment constraints. In *Intelligent Autonomous Systems*, North Holland, 1987.
- [40] J.P. Laumond. Finding collision-free smooth trajectories for a non-holonomic mobile robot. In *10th International Joint Conference on Artificial Intelligence*, Milano, Italy, 1987.
- [41] J.P. Laumond, M. Taix, and P. Jacobs. A motion planner for car-like robots based on a mixed global/local approach. In *IEEE IROS*, July 1990.
- [42] J.P. Laumond and T. Siméon. Motion planning for a two degrees of freedom mobile robot with towing. *LAAS/CNRS Report 89148* , Toulouse, April 1989.
- [43] J.P. Laumond, T. Siméon, R. Chatila, and G. Giralt. Trajectory planning and motion control for mobile robots. In *Geometry and Robotics, J.D. Boissonnat and J.P. Laumond (Eds)*, pages 133–149. Lecture Notes in Computer Science, Vol 391, Springer Verlag, 1989.

- [44] J.P. Laumond. Nonholonomic motion planning versus controllability via the multibody car system example. Technical Report STAN-CS 90-1345, Stanford Univ., 1990.
- [45] J.P. Laumond. Controllability of a multibody mobile robot. In *Proceedings of '91 ICAR*, Pisa, Italy, June 1991.
- [46] Z. Li and J. Canny. Motion of two rigid bodies with rolling constraint. *IEEE Trans. on Robotics and Automation*, 6, 1990.
- [47] C. Lobry. Controlabilité des systèmes non linéaires. *SIAM J. of Control*, 8:573–605, 1979.
- [48] J. Mitchell On Carnot-Carathéodory metrics. *J. Differential Geometry*, 21, pp 35-45, 1985.
- [49] R.M. Murray and S.S. Sastry. *Mathematical Questions in Robotics*, chapter Mathematical Problems in Grasping and Manipulation by Multifingered Robot Hands. American Mathematical Society, January 1990.
- [50] R.M. Murray and S.S. Sastry. Steering nonholonomic systems using sinuoids. *29th C.D.C., Honolulu, Dec. 1990*
- [51] R.M. Murray. Robotics control and nonholonomic motion planning. PhD Thesis UCB/ERL M90/117, Berkeley, December 1990.
- [52] Y. Nakamura and R. Mukherjee. Nonholonomic path planning of space robots via bi-directional approach. *IEEE International Conference on Robotics and Automation*, 1990.
- [53] Colm Ó'Dúnlait. Motion planning with inertial constraints. *Robotics Report 73*, New York University, July 1986.
- [54] E. Papadopoulos and S. Dubowsky. On the nature of control algorithms for space manipulators. *IEEE International Conference on Robotics and Automation*, 1990.
- [55] L. Pontryagin et al. *The Mathematical Theory of Optimal Processes*. L.S. Pontryagin Selected Works, Vol. 4, Gordon and Breach Sc. Pub., 1986.

- [56] J.A. Reeds and L.A. Shepp. Optimal paths for a car that goes both forwards and backwards. *Pacific J. Mathematics*, 145 (2), 1990.
- [57] H.L. Royden. *Real Analysis*. The Macmillan Company, New York, 1968.
- [58] E. Rimon and D.E. Koditschek. The construction of analytic diffeomorphisms for exact robot navigation on star worlds. In *Proceedings of the 1989 IEEE Int. Conference on Robotics and Automation*, pages 21–26, Scottsdale, May 1989.
- [59] C. Samson and K. Ait-Abderrahim. Feedback Control of a Nonholonomic Wheeled Cart in Cartesian Space. In *Proceedings of the 1991 IEEE Int. Conf. on Robotics and Automation*, Sacramento, April 1991.
- [60] M. Sharir. Algorithmic motion planning in robotics. Technical Report 392, Courant Institute, New-York University, 1988.
- [61] M. Spivak. *A Comprehensive Introduction to Differential Geometry*, volume 1. Publish or Perish, Berkeley, CA, 1979.
- [62] S. Sternberg *Lectures on Differential Geometry* Chelsea Pub., 1983.
- [63] R.S. Strichartz. Sub-riemannian geometry. *J. Differential Geometry*, 24, 221–263, 1986.
- [64] R.S. Strichartz. The Campbell-Baker-Hausdorff-Dynkin formula and solutions of differential equations. *J. of Functional Analysis*, 72, 320–345, 1987.
- [65] H.J. Sussmann and Jurdjevic. Controllability of nonlinear systems. *J. of Differential Equations*, 12, 1972.
- [66] H.J. Sussmann. Lie brackets, real analyticity and geometric control. In *Differential Geometric Control Theory*, Brockett, Millman, Sussmann Eds, Birkhäuser, 1982.

- [67] H.J. Sussmann and W. Liu. Limits of highly oscillatory controls and the approximation of general paths by admissible trajectories. Report SYCON-91-02, Rutgers University, 1991.
- [68] H.J. Sussmann and W. Tang. Shortest paths for the Reeds-Shepp car : a worked out example of the use of geometric techniques in nonlinear optimal control. Report SYCON-91-10, Rutgers University, 1991.
- [69] M. Taïx. Planification de mouvements pour robot mobile non holonome. PhD Thesis 824, P. Sabatier Univ., Toulouse, Jan. 1991.
- [70] D. Tilbury, J.P. Laumond, R. Murray, S. Sastry and G. Walsh. Steering car-like systems with trailers using sinusoids. Technical Report, Electronics Research Lab., Univ. of California, Berkeley, September 1991.
- [71] P. Tournassoud. Motion planning for a mobile robot with a kinematic constraint. In *Geometry and Robotics*, J.D. Boissonnat and J.P. Laumond (Eds), pages 150–171. Lecture Notes in Computer Science, Vol 391, Springer Verlag, 1989.
- [72] V.S. Varadarajan. *Lie Groups, Lie Algebra, and their Representations*, Springer-Verlag, 1984.
- [73] A.M. Vershik and V.Ya. Gershkovich. Nonholonomic problems and the theory of distributions. *Acta Applicandae Mathematicae*, 12, pp 181-209, 1988.
- [74] *Algèbres de Lie libres et monoïdes libres*. Lecture Notes in Mathematics, 691, Springer Verlag, 1978.
- [75] G.T. Wilfong. Motion planning for an autonomous vehicle. In *IEEE Int. Conf. on Robotics and Automation*, 1988.
- [76] C.K. Yap. *Algorithmic and geometric aspects of Robotics*, chapter Algorithmic motion planning. J.T. Schwartz and C.K. Yap (eds), Lawrence Erlbaum Associates, London, 1987.

6

MOTION PLANNING FOR NONHOLONOMIC DYNAMIC SYSTEMS

Mahmut Reyhanoglu N. Harris McClamroch

Department of Aerospace Engineering

The University of Michigan

Ann Arbor, Michigan 48109

Anthony M. Bloch¹

Department of Mathematics

The Ohio State University

Columbus, Ohio 43210

Abstract

The problem of motion planning for nonholonomic dynamic systems is studied. A model for nonholonomic dynamic systems is first presented in terms of differential-algebraic equations defined on a phase space. A nonlinear control system in a normal form is introduced to completely describe the dynamics. The assumptions guarantee that the resulting normal form equations necessarily contain a nontrivial drift vector field. We show that the linearized control system always has uncontrollable eigenvalues at the origin. However, any equilibrium is shown to be strongly accessible and small time locally controllable. A motion planning approach using holonomy is developed for nonholonomic Caplygin dynamical systems, i.e. nonholonomic systems with certain symmetry properties which can be expressed by the fact that the constraints are cyclic in certain of the variables. The theoretical development is applied to physical examples of systems that we have studied in detail elsewhere: the control of motion of a knife edge moving on a plane surface and the control of motion of a wheel rolling without slipping on a plane surface. The results of the paper are also applied to the control of motion of a planar multibody system using angular momentum preserving joint torques. Results of simulations are included to illustrate the effectiveness of the proposed motion planning approach.

¹Supported by in part by NSF Grant DMS-9002136
and NSF PYI Grant DMS-9157556

1. Introduction

Numerous papers have been published in recent years on the control of nonholonomic systems. Analytical techniques for studying nonholonomic motion planning have recently received considerable attention. Li and Canny [10] studied the motion of a fingertip rolling on an object without slipping. Li and Montgomery [11] studied a hopping robot flipping in mid-air by recasting the problem into a nonholonomic motion planning problem. Similar techniques have also been used for studying the motion of coupled rigid bodies and space manipulators [7], [8], [19]. Laffarriere and Sussmann [9] have recently proposed a motion planning method using tools from geometric control theory and Lie algebra. Murray and Sastry [14] have extended the work by Brockett [6] on motion planning using sinusoids. The above work mostly has considered nonholonomic systems defined solely by kinematic relations, disregarding the system dynamics. The resulting “nonholonomic kinematic systems” are characterized by the fact that there are no drift vector fields in their control representations. In this paper we focus on the control of “nonholonomic dynamic systems”, so that control representations include nontrivial drift vector fields. Nonholonomic dynamic systems have been considered by Bloch and McClamroch [2]. Our recent work in [3], [4], [5], [16], [17] has demonstrated a common theoretical framework for a large class of control problems for nonholonomic dynamic systems. Our development is based on the formulation of nonholonomic dynamics by Neimark and Fufaev [15] and on the modern formulation of nonlinear geometric control.

2. Nonholonomic Dynamic Systems

We consider the class of nonholonomic systems described by the equations

$$M(q)\ddot{q} + F(q, \dot{q}) = J'(q)\lambda + B(q)u , \quad (1)$$

$$J(q)\dot{q} = 0 . \quad (2)$$

Note that a “prime” denotes transpose. We refer to q as an n -vector of generalized configuration variables, \dot{q} as an n -vector of generalized

velocity variables, and \ddot{q} as an n -vector of generalized acceleration variables; in addition, u is an r -vector of control input variables and λ is an m -vector of constraint multipliers. The $n \times n$ matrix function $M(q)$ is assumed to be symmetric and positive definite, $F(q, \dot{q})$ is an n -vector function, $J(q)$ denotes an $m \times n$ matrix function which is assumed to have full rank and $B(q)$ is a full rank $n \times r$ matrix function. All of these functions are assumed to be smooth (C^∞ and defined on an appropriate open subset of the (q, \dot{q}) phase space). The formulation could be given in terms of a system defined on the tangent bundle of a C^∞ manifold; we have not made such a generalization since it is direct. Various assumptions about the control input variables are indicated subsequently.

Differential-algebraic equations of the above form are known to arise for (uncontrolled) nonholonomic systems; see Arnold [1] and Neimark and Fufaev [15] for many examples. We note that here the classical approach for the formulation of constrained dynamics as described in [15] is used. This is in contrast to the variational approach, or “vaconomic” theory, which is considered in [1]. We also note that a Hamiltonian formulation can be developed.

We have assumed that the $m \times n$ matrix $J(q)$ has full rank; hence there is no loss of generality in assuming that the configuration variables are ordered so that the last m columns of the matrix $J(q)$ constitute an $m \times m$ locally invertible matrix function, i.e. the matrix $J(q)$ can be expressed as $[J_1(q) \ J_2(q)]$, where $J_1(q)$ is an $m \times (n-m)$ matrix function and $J_2(q)$ is an $m \times m$ locally nonsingular matrix function. The columns of the $n \times (n-m)$ matrix function

$$C(q) = \begin{bmatrix} I \\ -\bar{J}(q) \end{bmatrix}, \quad (3)$$

where I is the $(n-m) \times (n-m)$ identity matrix and $\bar{J}(q) = J_2^{-1}(q)J_1(q)$ is a locally smooth $m \times (n-m)$ matrix function, span the null space of $J(q)$. Formally, the rows of $J(q)$ constitute m linearly independent smooth covector fields defined on the configuration space; these covector fields span a codistribution Ω and the annihilator of the codistribution Ω , denoted Ω^\perp , is spanned by $n-m$ linearly independent smooth vector fields

$$\tau_j = \sum_{i=1}^n C_{ij}(q) \frac{\partial}{\partial q_i}, \quad j = 1, \dots, n-m. \quad (4)$$

We present the following definition.

Definition 1 [22]: Consider the following nondecreasing sequence of locally defined distributions

$$N_1 = \Omega^\perp,$$

$$N_k = N_{k-1} + \text{span}\{[X, Y] \mid X \in N_1, Y \in N_{k-1}\}.$$

There exists an integer k^* such that

$$N_k = N_{k^*}$$

for all $k > k^*$. If $\dim N_{k^*} = n$ and $k^* > 1$ then the constraints (2) are called completely nonholonomic and the smallest (finite) number k^* is called the degree of nonholonomy.

In this paper it is assumed that constraint equations (2) are completely nonholonomic with nonholonomy degree k^* . Note that for this to hold $n - m$ must be strictly greater than one. Note also that since the constraints are nonholonomic, there is in fact no explicit restriction on the values of the configuration variables.

The constraints (2) define a $(2n - m)$ -dimensional smooth submanifold

$$M = \{(q, \dot{q}) \mid J(q)\dot{q} = 0\} \quad (5)$$

of the phase space. This manifold M plays a critical role in the concept of solutions and the formulation of control and stabilization problems associated with equations (1)-(2).

We begin by making clear that equations (1)-(2) do represent well posed models in the sense that the associated initial value problem has a unique solution, at least locally.

Definition 2 : A pair of vector functions $(q(t), \lambda(t))$ defined on an interval $[0, T]$ is a solution of the initial value problem defined by

equations (1)-(2) and the initial data (q_0, \dot{q}_0) if $q(t)$ is at least twice differentiable, $\lambda(t)$ is integrable, the vector functions $(q(t), \lambda(t))$ satisfy the differential-algebraic equations (1)-(2) almost everywhere on their domain of definition and the initial conditions satisfy $(q(0), \dot{q}(0)) = (q_0, \dot{q}_0)$.

The following existence and uniqueness result has been obtained.

Theorem 1 [2]: *Assume that the control input function $u : [0, T] \rightarrow R^r$ is a given bounded and measurable function for some $T > 0$. If the initial condition data satisfy $(q_0, \dot{q}_0) \in M$, then there exists a unique solution (at least locally defined) of the initial value problem corresponding to equations (1)-(2) which satisfies $(q(t), \dot{q}(t)) \in M$ for each t for which the solution is defined.*

Since the differential-algebraic equations (1)-(2) define a smooth vector field on M , a number of other results could be stated, including conditions for continuous dependence of the solution on initial conditions and parameters, conditions for non-existence of finite escape times, etc. Such results are important, but they are not given here since they are easily obtained. We subsequently use the notation $(Q(t, q_0, \dot{q}_0), \Lambda(t, q_0, \dot{q}_0))$ to denote the solution of equations (1)-(2) at time $t \geq 0$ corresponding to the initial conditions (q_0, \dot{q}_0) . Thus for each initial condition $(q_0, \dot{q}_0) \in M$ and each bounded, measurable input function $u : [0, T] \rightarrow R^r$, $(Q(t, q_0, \dot{q}_0), \dot{Q}(t, q_0, \dot{q}_0)) \in M$ holds for all $t \geq 0$ where the solution is defined.

A particularly important class of solutions are the equilibrium solutions of (1)-(2). A solution is an equilibrium solution if it is a constant solution; note that if (q^e, λ^e) is an equilibrium solution we refer to q^e as an equilibrium configuration. The following result should be clear.

Theorem 2. *Suppose that $u(t) = 0$, $t \geq 0$. The set of equilibrium configurations of equations (1)-(2) is given by*

$$\{q \mid F(q, 0) - J'(q)\lambda = 0 \text{ for some } \lambda \in R^m\},$$

An equivalent expression for the set of equilibrium configurations is

$$\{q \mid C'(q)F(q, 0) = 0\}.$$

We remark that generically the equilibrium manifold has dimension at least m . On the other hand, for certain cases, there may not be even a single equilibrium configuration (e.g. the dynamics of a ball on an inclined plane). However, since we have assumed that $C'(q)B(q)$ is full rank, we can always introduce an equilibrium manifold of dimension at least m by appropriate choice of input.

A number of approaches have been suggested for eliminating the constraint multipliers so that a minimum set of differential equations is obtained: the reduced differential equations characterize the control dependent motion on the constraint manifold.

We first emphasize that the reduced state space is $2n - m$ dimensional. The state of the system can be specified by the n -vector of configuration variables and an $(n - m)$ -vector of kinematic variables. Let $q = (q_1, q_2)$ be a partition of the configuration variables corresponding to the partitioning of the matrix function $J(q)$ introduced previously. Then consider the following relation

$$\dot{q} = C(q)\dot{q}_1$$

where $C(q)$ is defined by equation (3). Taking time derivatives yields

$$\ddot{q} = C(q)\ddot{q}_1 + \dot{C}(q)\dot{q}_1$$

where $\dot{C}(q)$ denotes the time derivative of $C(q)$. Substituting this into equation (1) and multiplying both sides of the resulting equation by $C'(q)$ gives

$$C'(q)M(q)C(q)\ddot{q}_1 = C'(q)[B(q)u - F(q, C(q)\dot{q}_1) - M(q)\dot{C}(q)\dot{q}_1]. \quad (6)$$

Note that $C'(q)M(q)C(q)$ is an $(n - m) \times (n - m)$ symmetric positive definite matrix function.

We also assume that $r = n - m$ (for simplicity). Then the matrix product $C'(q)B(q)$ is locally invertible. Consequently, for any $u \in R^r$ there is unique $v \in R^{n-m}$ which satisfies

$$C'(q)[B(q)u - F(q, C(q)\dot{q}_1) - M(q)\dot{C}(q)\dot{q}_1] = C'(q)M(q)C(q)v. \quad (7)$$

(Note that if $r > n - m$ then v can be chosen to depend smoothly on the variables (q, \dot{q}_1, u)). This assumption guarantees that the reduced configuration variables satisfy the linear equations

$$\ddot{q}_1 = v .$$

Define the following state variables

$$x_1 = q_1 ,$$

$$x_2 = q_2 ,$$

$$x_3 = \dot{q}_1 .$$

Then the normal form equations are given by

$$\dot{x}_1 = x_3 , \quad (8)$$

$$\dot{x}_2 = -\bar{J}(x_1, x_2)x_3 , \quad (9)$$

$$\dot{x}_3 = v . \quad (10)$$

Equations (8)-(10) define a drift field $f(x) = (x_3, -\bar{J}(x_1, x_2)x_3, 0)$ and control vector fields $g_i(x) = (0, 0, e_i)$, where e_i is the i 'th standard basis vector in R^{n-m} , $i = 1, \dots, n - m$, according to the standard control system form

$$\dot{x} = f(x) + \sum_{i=1}^{n-m} g_i(x)v_i . \quad (11)$$

We now describe a special class of dynamical systems with classical nonholonomic constraints. If the functions used in defining equations (1)-(2) do not depend explicitly on the configuration variables q_2 , so that the system is locally described by

$$M(q_1)\ddot{q} + F(q_1, \dot{q}) = [\bar{J}(q_1) \ I]' \lambda + B(q_1)u , \quad (12)$$

$$\bar{J}(q_1)\dot{q}_1 + \dot{q}_2 = 0 , \quad (13)$$

where $\bar{J}(q_1)$ is an $m \times (n - m)$ matrix function and I is the $m \times m$ identity matrix, then the uncontrolled system is called a "nonholonomic Caplygin system" by Neimark and Fufaev [15]. In terms of the Lagrangian formalism for the problem this corresponds to the Lagrangian of the free problem being cyclic in (i.e. independent of) the variables q_2 while the constraints are also independent of q_2 . The cyclic property is an expression of symmetries in the problem, such symmetries occurring naturally in many physical problems. More generally, if a system can be expressed in the form (12)-(13) using feedback, then we refer to it as a "controlled nonholonomic Caplygin system".

For the nonholonomic Caplygin system described by equations (12)-(13), equation (7) becomes

$$\begin{aligned} C'(q_1)M(q_1)C(q_1)\ddot{q}_1 = \\ C'(q_1)[B(q_1)u - F(q_1, C(q_1)\dot{q}_1) - M(q_1)\dot{C}(q_1)\dot{q}_1] , \end{aligned} \quad (14)$$

which is an equation in the phase variables (q_1, \dot{q}_1) only. As a consequence, q_1 constitutes a reduced configuration space for the system (12)-(13). This reduced configuration space is also referred to as the "base space" (or "shape space") of the system. The term shape space (see [7], [8], [10], [11]) arises from the theory of coupled mechanical systems, where it refers to the internal degrees of freedom of the system. It is possible to consider control theoretic problems which can be expressed solely in the base space, which can be solved using classical methods. However, in our work, we are interested in the more general control problems associated with the complete dynamics defined by equations (12)-(13), which are reflected in terms of the equations (13)-(14), or (equivalently) by the following normal form equations

$$\dot{x}_1 = x_3 , \quad (15)$$

$$\dot{x}_2 = -\bar{J}(x_1)x_3 , \quad (16)$$

$$\dot{x}_3 = v . \quad (17)$$

where $x_1 = q_1$, $x_2 = \dot{q}_1$, $x_3 = q_2$ and v satisfies

$$\begin{aligned} C'(q_1)[B(q_1)u - F(q_1, C(q_1)\dot{q}_1) - M(q_1)\dot{C}(q_1)\dot{q}_1] = \\ C'(q_1)M(q_1)C(q_1)v. \end{aligned} \quad (18)$$

We remark that the dimension of the base space is unique, equal to the number of degrees of freedom; however the identity of the base space variables is not unique.

3. Controllability of Nonholonomic Dynamic Systems

The simplest approach to studying controllability of the nonlinear system described by (11) is to consider its linearization. The linearization of (11) about $x = x^e$ and $u = 0$ can be written as

$$\delta\dot{x} = A\delta x + B\delta u \quad (19)$$

where

$$A = \begin{bmatrix} 0 & 0 & I_{N-1} \\ 0 & 0 & -\bar{J}(x_1^e, x_2^e) \\ 0 & 0 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 0 \\ I \end{bmatrix}.$$

Obviously, the linearization is not controllable since the Kalman rank condition is not satisfied. Thus the first order test for controllability is inconclusive.

We consider the nonlinear control system (11) and employ certain results of nonlinear control theory. We first recall some relevant definitions and results from the nonlinear control literature.

Let $\mathcal{R}(p,t)$ denote the set of reachable states from initial state p in time exactly t , for equation (11). The following definitions are now standard.

Accessibility [21]: *The system (11) is accessible if for all $p \in M$ and any $T > 0$, $\bigcup_{t \leq T} \mathcal{R}(p,t)$ has a nonvoid interior with respect to M .*

Strong Accessibility [21]: *The system (11) is strongly accessible if for all $p \in M$ and any $T > 0$, $\mathcal{R}(p,T)$ has a nonvoid interior with respect to M .*

Small-time Local Controllability (STLC) [20]: *The system (11) is STLC from p if for any $T > 0$, p is an interior point of*

$$\cup_{t \leq T} \mathcal{R}(p, t).$$

We now demonstrate that the normal form equations (8)-(10), and hence the nonholonomic control system defined by equations (1)-(2), does indeed satisfy certain strong local controllability properties. In particular, we show that the system is strongly accessible and that the system is small time locally controllable to any equilibrium. These results not only provide a theoretical basis for the use of inherently nonlinear control strategies but they also suggest constructive procedures for the desired control strategies.

Theorem 3. *Let $m \geq 1$ and let $x^e = (x_1^e, x_2^e, 0)$ denote an equilibrium solution in \mathbf{M} . The nonholonomic control system, defined by equations (1)-(2), is strongly accessible at x_e .*

Proof: It suffices to prove that system (8)-(10) is strongly accessible at the origin. Let I denote the set $\{1, \dots, n-m\}$. The drift and control vector fields can be expressed as

$$f = \sum_{j=1}^{n-m} x_{3,j} \tau_j ,$$

$$g_i = \frac{\partial}{\partial x_{3,i}} , \quad i \in I ,$$

where

$$\tau_j = \frac{\partial}{\partial x_{1,j}} - \sum_{i=1}^{n-m} \bar{J}_{ij}(x_1, x_2) \frac{\partial}{\partial x_{2,i}} , \quad j \in I$$

are considered as vector fields on the (x_1, x_2, x_3) state space. It can be verified that

$$[g_{i_1}, f] = \tau_{i_1} , \quad i_1 \in I ;$$

$$[g_{i_2}, [f, [g_{i_1}, f]]] = [\tau_{i_2}, \tau_{i_1}] , \quad i_1, i_2 \in I ;$$

⋮

$$[g_{i_{k^*}}, [f, \dots, [g_{i_2}, [f, [g_{i_1}, f]]] \dots]] = \\ [\tau_{i_{k^*}}, \dots, [\tau_{i_2}, \tau_{i_1}] \dots], \quad i_k \in I, 1 \leq k \leq k^* ,$$

hold locally, where k^* denotes the nonholonomy degree at the origin. Let

$$\mathcal{G} = \text{span}\{g_i, i \in I\},$$

$$\begin{aligned} \mathcal{H} = & \text{span}\{[g_{i_1}, f], \dots, [g_{i_{k^*}}, [f, \dots, [g_{i_2}, [f, [g_{i_1}, f]]] \dots]] ; \\ & i_k \in I, 1 \leq k \leq k^* \}. \end{aligned}$$

Note that $\dim \mathcal{G}(0) = n - m$ and $\dim \mathcal{H}(0) = n$ since the distribution defined by the constraints is completely nonholonomic; moreover $\dim \{\mathcal{G}(0) \cap \mathcal{H}(0)\} = 0$. It follows that the strong accessibility distribution

$$\mathcal{L}_0 = \text{span}\{X : X \in \mathcal{G} \cup \mathcal{H}\}$$

has dimension $2n - m$ at the origin. Hence the strong accessibility rank condition is satisfied at the origin. Thus system (8)-(10) is strongly accessible at the origin. Consequently, the nonholonomic control system, defined by equations (1)-(2), is strongly accessible at x_e .

Theorem 4. *Let $m \geq 1$ and let $x_e = (x_1^e, x_2^e, 0)$ denote an equilibrium solution in \mathbf{M} . The nonholonomic control system, defined by equations (1)-(2), is small time locally controllable at x_e .*

Proof: A precise and detailed proof in terms of indeterminates is given in [4]. The essential ingredients are as follows.

It suffices to prove that system (8)-(10) is small time locally controllable at the origin. By Theorem 1, the system is accessible at the origin.

Now, following Sussmann [20], let $Br(X)$ denote the smallest Lie algebra of vector fields containing $\{f, g_1, \dots, g_{n-m}\}$ and let B denote a bracket in $Br(X)$. Let $\delta^0(B), \delta^1(B), \dots, \delta^{n-m}(B)$ denote the number of times f, g_1, \dots, g_{n-m} , respectively, occur in the bracket B and let $\delta(B) = \sum_{i=0}^{n-m} \delta^i(B)$ denote its degree. The bracket B is called "bad" if $\delta^0(B)$ is odd and $\delta^i(B)$ is even for each i

in I. The Sussmann condition is that "bad" brackets may be written as a linear combination of brackets of lower degree at the origin.

The brackets in \mathcal{G} are obviously "good" (not of the type defined as "bad") and $\delta^0(h) = \sum_{j=1}^{n-m} \delta^j(h) \forall h \in \mathcal{H}$; thus $\delta(h)$ is even for all h in \mathcal{H} , i.e. \mathcal{H} contains "good" brackets only. It follows that the tangent space $T_0 M$ to M at the origin is spanned by the brackets that are all "good". Next we show that the brackets that might be "bad" vanish at the origin. First note that f vanishes at the origin. Let B denote a bracket satisfying $\delta(B) > 1$. If B is a "bad" bracket then necessarily $\delta^0(B) \neq \sum_{j=1}^{n-m} \delta^j(B)$, i.e. $\delta(B)$ must be odd. It can be verified that if $\delta^0(B) < \sum_{j=1}^{n-m} \delta^j(B)$ then B is identically zero and if $\delta^0(B) > \sum_{j=1}^{n-m} \delta^j(B)$ then B is of the form $\sum_{i=1}^{n-m} r_i(x_3) Y_i(x_1, x_2)$, for some vector fields $Y_i(x_1, x_2)$, $i \in I$, where $r_i(x_3)$, $i \in I$, are homogeneous functions of degree $(\delta^0(B) - \sum_{j=1}^{n-m} \delta^j(B))$ in x_3 ; thus B vanishes at the origin. Consequently the Sussmann condition is satisfied. Hence system (8)-(10) is small time locally controllable at the origin. It follows that, the nonholonomic control system, defined by equations (1)-(2), is small time locally controllable at x_e .

4. Nonholonomic Motion Planning Framework

The nonholonomic motion planning problem has been studied by a number of authors. Murray and Sastry [14] have proposed procedures that work well in a number of special cases. These procedures use an optimal control approach by making special assumptions on certain Lie algebras. The strategy in Laffarriere and Sussmann [9] does not in principle require special Lie algebraic assumptions and does not use optimal control. Their strategy works exactly for nilpotent and nilpotentizable systems; it constitutes a successive approximations algorithm that converges to a solution for non-nilpotentizable systems. We point out that these methods work because of the absence of drift. Here we remove this restriction by studying the motion planning problem for nonholonomic dynamic systems, which necessarily include nontrivial drift vector fields. Our motivation is as follows: (1) since mechanical systems are ultimately force or torque controlled, we seek to develop dynamics based motion planning procedures which explicitly include such control vari-

ables; (2) such a formulation is crucial if limits are imposed on accelerations or controls; (3) the dynamics based formulation is ultimately required for analysis of feedback issues and to account for initial condition and modeling errors, noise, external disturbances and other effects that are encountered in any real implementation.

We now define the problem of rest-to-rest motion planning for nonholonomic dynamic systems. Suppose that $(q^0, 0)$ and $(q^f, 0)$ are given points in M . The nonholonomic motion planning problem (NMPP) is the problem of determining a motion $(q(t), \dot{q}(t))$, $0 \leq t \leq t_f$, such that $(q(0), \dot{q}(0)) = (q^0, 0)$, $(q(t_f), \dot{q}(t_f)) = (q^f, 0)$ and $(q(t), \dot{q}(t))$ satisfies the reduced equations

$$\dot{q} = C(q)\dot{q}_1 ,$$

$$\ddot{q}_1 = [C'(q)M(q)C(q)]^{-1}C'(q)[B(q)u - F(q, C(q)\dot{q}_1) - M(q)\dot{C}(q)\dot{q}_1] ,$$

for some control function $t \mapsto u(t)$.

In this paper we construct a control which transfers any $(q^0, 0)$ to $(0, 0)$ (or equivalently $x^0 = (x_1^0, x_2^0, 0)$ to $x^f = (0, 0, 0)$). The controllability results proved previously guarantee that such a control exist for every x^0 which is close to the origin because local controllability guarantees only local solution to the NMPP. However, the local solutions can be patched together to yield global solutions as will be illustrated in the subsequent examples.

Note that the problem does not have a unique solution. In this paper we describe one solution, outline the theory behind it, and present some data from simulations. We will demonstrate that the theory works well for Caplygin nonholonomic dynamic systems.

We now describe the ideas that are employed to construct an open loop control which achieves the desired goal for motion planning of Caplygin nonholonomic dynamic systems. These ideas are based on the use of holonomy (geometric phase) which has proved useful in a variety of kinematics and dynamics problems (see e.g. Krishnaprasad [7], [8] and Montgomery [13]). More information concerning geometric phases can be found in the recent book [18] of Shapere and Wilczek, and a review article [12] of Marsden, Montgomery and Ratiu. Our use of holonomy is, to the best of our knowledge, its first application to nonlinear control systems of the form

(8)-(10) which contain nontrivial drift vector fields [5], [16], [17]. The key observation is that the holonomy, the extent to which a closed path in the base space fails to be closed in the configuration space, depends only on the path traversed in the base space and not on the time history of traversal of the path. Related ideas have been used for a class of path planning problems, based on kinematic relations, in Li and Canny [10], Li and Montgomery [11].

Let $x^0 = (x_1^0, x_2^0, 0)$ denote an initial state. We now describe two steps involved in construction of a control strategy which transfers the initial state to the origin.

Step 1 : Bring the state of the system to the origin of the (x_1, x_3) base phase space, i.e. find a control which transfers the initial state $(x_1^0, x_2^0, 0)$ to $(0, x_2^1, 0)$ in a finite time, for some x_2^1 .

Step 2 : Traverse a closed path (or a series of closed paths) in the x_1 base space to produce a desired holonomy in the (x_1, x_2) configuration space, i.e. find a control which transfers $(0, x_2^1, 0)$ to $(0, 0, 0)$.

The desired holonomy condition is given by

$$x_2^1 = \oint_{\gamma} \bar{J}(x_1) dx_1 , \quad (20)$$

where γ denote a closed path traversed in the base space. The holonomy in the system is reflected in the fact that traversing a closed path (or a concatenation of a series of closed paths) in the base space yields a nonclosed path in the full configuration space.

Under the weak assumptions mentioned previously, explicit procedures can be given for each of the above two steps. Step 1 is classical since the relevant dynamics are described by a set of $n - m$ decoupled double integrators; it is step 2, involving the holonomy, that requires special consideration. Explicit characterization of a closed path γ which satisfies the desired holonomy equation (20) can be given for several specific examples. In the next section, we present three such examples. However, some problems may require a general computational approach. An algorithm based on Lie algebraic methods can be employed to approximately characterize the

required closed path. Suppose the closed path γ which satisfies the desired holonomy condition is chosen. Then an open loop control which forces the base variables to track the closed path in the base space can be constructed since the base space equations (15),(17) constitute $n - m$ decoupled double integrators on the base space. This general construction procedure provides a strategy for transferring an arbitrary initial state of equations (15)-(17) to the origin.

We remark that the technique presented in this section can be generalized to some systems which are not Caplygin. For instance, this generalization is tractible to systems for which equation (16) takes the form

$$\dot{x}_2 = \rho(x_2)\bar{J}(x_1)$$

where $\rho(x_2)$ denotes certain Lie group representation (see e.g. [12]). The holonomy of a closed path for such systems is given as a path ordered exponential rather than a path integral.

5. Examples

Dynamics of Knife Edge Using Steering and Pushing Inputs

We first consider the dynamics of a knife edge moving in point contact on a plane surface [2], [3], [5]. Let x and y denote the coordinates of the point of contact of the knife edge on the plane and let ϕ denote the heading angle of the knife edge, measured from the x -axis. Then the equations of motion, with all numerical constants set to unity, are given by

$$\ddot{x} = \lambda \sin \phi + u_1 \cos \phi \quad (21)$$

$$\ddot{y} = -\lambda \cos \phi + u_1 \sin \phi \quad (22)$$

$$\ddot{\phi} = u_2 \quad (23)$$

where u_1 denotes the control force in the direction defined by the heading angle, u_2 denotes the control torque about the vertical axis through the point of contact; the components of the force of constraint arise from the scalar nonholonomic constraint

$$\dot{x} \sin \phi - \dot{y} \cos \phi = 0 \quad (24)$$

which has nonholonomy degree two at any configuration. It is clear that the constraint manifold is a five-dimensional manifold and is defined by

$$\mathbf{M} = \{(\phi, x, y, \dot{\phi}, \dot{x}, \dot{y}) | \dot{x} \sin \phi - \dot{y} \cos \phi = 0\}$$

and any configuration is an equilibrium if the controls are zero.

Define the variables

$$x_1 = x \cos \phi + y \sin \phi ,$$

$$x_2 = \phi ,$$

$$x_3 = -x \sin \phi + y \cos \phi ,$$

$$x_4 = \dot{x} \cos \phi + \dot{y} \sin \phi - \dot{\phi}(x \sin \phi - y \cos \phi) ,$$

$$x_5 = \dot{\phi} ,$$

so that the reduced differential equations are given by

$$\dot{x}_1 = x_4 ,$$

$$\dot{x}_2 = x_5 ,$$

$$\dot{x}_3 = -x_1 x_5 ,$$

$$\dot{x}_4 = u_1 + u_2 x_3 - x_1 x_5^2 ,$$

$$\dot{x}_5 = u_2 .$$

Consequently, equations (21)-(24) represent a controlled nonholonomic Caplygin system with base space equations which are feedback linearizable. Normal form equations are given by

$$\dot{x}_1 = x_4 ,$$

$$\dot{x}_2 = x_5 ,$$

$$\dot{x}_3 = -x_1 x_5 ,$$

$$\dot{x}_4 = v_1 ,$$

$$\dot{x}_5 = v_2 ,$$

where

$$v_1 = u_1 + u_2 x_3 - x_1 x_5^2 , \quad (25)$$

$$v_2 = u_2 . \quad (26)$$

The following conclusions are based on analysis of the above normal form equations.

Proposition 1. *Consider the knife edge dynamics introduced above. Let $x^e = (x_1^e, x_2^e, x_3^e, 0, 0)$ denote an equilibrium solution of the reduced differential equations corresponding to $u = 0$. The knife edge dynamics described by equations (21)-(24) have the following properties:*

1. *The system is strongly accessible at x^e since the space spanned by the vectors*

$$g_1, g_2, [g_1, f], [g_2, f], [g_2, [f, [g_1, f]]]$$

has dimension 5 at x^e .

2. *The system is small time locally controllable at x^e since the brackets satisfy sufficient conditions for small time local controllability.*

Let $\gamma(a, b)$ denote the closed path, parameterized by the pair (a, b) , obtained by starting at the origin $(0, 0)$ of the base space and then moving from $(0, 0)$ to $(a, 0)$, from $(a, 0)$ to (a, b) , from (a, b) to $(0, b)$, and finally from $(0, b)$ back to the origin. Also let $\gamma(a^k, b^k)o\cdots o\gamma(a^1, b^1)$ denote a concatenation of k closed paths parameterized by (a^i, b^i) , $i = 1, \dots, k$.

Note that the base variables are (x_1, x_2) . Consider a parameterized rectangular closed path γ in the base space with four corner points

$$(0, 0), (x_1, 0), (x_1, x_2), (0, x_2) .$$

By evaluating the integral in (20) in closed form for this case, the desired holonomy equation is

$$x_3^1 = x_1 x_2 .$$

This equation can be explicitly solved to determine a closed path $\gamma^* = \gamma(x_1^*, x_2^*)$ which achieves the desired holonomy. One solution can be given as follows

$$\gamma^* = \gamma(\operatorname{sign} x_3^1 \sqrt{|x_3^1|}, \sqrt{|x_3^1|}) .$$

A control function to accomplish Step 1 is given as

$$v_{[0,t_1)}^0 = \begin{cases} \left(\begin{array}{cc} -\frac{\pi x_1^0}{t_1} \cos\left(\frac{\pi t}{t_1}\right) & -\frac{\pi x_2^0}{t_1} \cos\left(\frac{\pi t}{t_1}\right) \\ -\frac{8\pi x_1^0}{t_1^2} \sin\left(\frac{2\pi(2t-t_1)}{t_1}\right) & -\frac{8\pi x_2^0}{t_1^2} \sin\left(\frac{2\pi(2t-t_1)}{t_1}\right) \end{array} \right) & 0 \leq t < 0.5t_1 , \\ \left(\begin{array}{cc} 0 & 0 \end{array} \right) & 0.5t_1 \leq t < t_1 . \end{cases}$$

We define the following control functions corresponding to the four segments of γ^* :

$$v_{[t_1,t_2)}^1 = \left(\begin{array}{cc} \frac{2\pi x_1^*}{(t_2-t_1)^2} \sin\left(\frac{2\pi(t-t_1)}{(t_2-t_1)}\right) & 0 \end{array} \right) ,$$

$$v_{[t_2,t_3)}^2 = \left(\begin{array}{cc} 0 & \frac{2\pi x_2^*}{(t_3-t_2)^2} \sin\left(\frac{2\pi(t-t_2)}{(t_3-t_2)}\right) \end{array} \right) ,$$

$$v_{[t_3,t_4)}^3 = \left(\begin{array}{cc} -\frac{2\pi x_1^*}{(t_4-t_3)^2} \sin\left(\frac{2\pi(t-t_3)}{(t_4-t_3)}\right) & 0 \end{array} \right) ,$$

$$v_{[t_4,t_5)}^4 = \left(\begin{array}{cc} 0 & -\frac{2\pi x_2^*}{(t_5-t_4)^2} \sin\left(\frac{2\pi(t-t_4)}{(t_5-t_4)}\right) \end{array} \right) ,$$

where $0 < t_1 \dots < t_5 = T$ are arbitrary. Combining the control functions $v_{[t_k,t_{k+1})}^k$, $k = 0, 1, 2, 3, 4$, where $t_0 = 0$, we describe

$$v(x^0, t) = \begin{cases} v^0, & t \in [0, t_1) , \\ v^1, & t \in [t_1, t_2) , \\ v^2, & t \in [t_2, t_3) , \\ v^3, & t \in [t_3, t_4) , \\ v^4, & t \in [t_4, t_5) . \end{cases} \quad (27)$$

The corresponding control u can be computed using equations (25)-(26). It is clear that the constructed control transfers any initial state of the system to the origin at time $T = t_5$. The resulting motion of the knife edge is obtained by solving an initial value problem using the control defined by (27).

We present a representative simulation example, transferring the initial state $(-1, \pi/4., 3, 0, 0)$ to the origin. The time responses for x, y, ϕ are shown in Figure 1. In Figure 2, the configuration of the knife edge is shown for a sequence of time instants.

Dynamics of Rolling Wheel Using Steering and Driving Inputs

As a second example, we consider the dynamics of a vertical wheel rolling without slipping on a plane surface [2], [5]. Let x and y denote the coordinates of the point of contact of the wheel on the plane, let ϕ denote the heading angle of the wheel, measured from the x -axis and let θ denote the rotation angle of the wheel due to rolling, measured from a fixed reference. Then the equations of motion, with all numerical constants set to unity, are given by

$$\ddot{x} = \lambda_1 \quad (28)$$

$$\ddot{y} = \lambda_2 \quad (29)$$

$$\ddot{\theta} = -\lambda_1 \cos \phi - \lambda_2 \sin \phi + u_1 \quad (30)$$

$$\ddot{\phi} = u_2 \quad (31)$$

where u_1 denotes the control torque about the rolling axis of the wheel and u_2 denotes the control torque about the vertical axis through the point of contact; the components of the force of constraint arise from the two nonholonomic constraints

$$\dot{x} = \dot{\theta} \cos \phi , \quad (32)$$

$$\dot{y} = \dot{\theta} \sin \phi \quad (33)$$

which have nonholonomy degree three at any configuration.

The constraint manifold is a six-dimensional manifold and is given by

$$\mathbf{M} = \{(\theta, \phi, x, y, \dot{\theta}, \dot{\phi}) | \dot{x} = \dot{\theta} \cos \phi, \dot{y} = \dot{\theta} \sin \phi\}$$

and any configuration is an equilibrium if the controls are zero.

Define the variables

$$x_1 = \theta, \quad x_2 = \phi, \quad x_3 = x, \quad x_4 = y, \quad x_5 = \dot{\theta}, \quad x_6 = \dot{\phi}$$

so that the reduced differential equations are given by

$$\dot{x}_1 = x_5 ,$$

$$\dot{x}_2 = x_6 ,$$

$$\dot{x}_3 = x_5 \cos x_2 ,$$

$$\dot{x}_4 = x_5 \sin x_2 ,$$

$$\dot{x}_5 = \frac{1}{2}u_1 ,$$

$$\dot{x}_6 = u_2 .$$

Consequently, equations (28)-(33) represent a controlled nonholonomic Caplygin system with base space equations which are feedback linearizable. Normal form equations are given by

$$\dot{x}_1 = x_5 ,$$

$$\dot{x}_2 = x_6 ,$$

$$\dot{x}_3 = x_5 \cos x_2 ,$$

$$\dot{x}_4 = x_5 \sin x_2 ,$$

$$\dot{x}_5 = v_1 ,$$

$$\dot{x}_6 = v_2 ,$$

where

$$v_1 = \frac{1}{2}u_1 , \tag{34}$$

$$v_2 = u_2 . \tag{35}$$

The following conclusions are based on analysis of the above normal form equations.

Proposition 2. Consider the rolling wheel dynamics introduced above. Let $x^e = (x_1^e, x_2^e, x_3^e, x_4^e, 0, 0)$ denote an equilibrium solution of the reduced differential equations corresponding to $u = 0$. The rolling wheel dynamics described by equations (28)-(33) have the following properties:

1. The system is strongly accessible at x^e since the space spanned by the vectors

$$g_1, g_2, [g_1, f], [g_2, f], [g_2, [f, [g_1, f]]], [g_2, [f, [g_1, [f, [g_2, f]]]]]$$

has dimension 6 at x^e .

2. The system is small time locally controllable at x^e since the brackets satisfy sufficient conditions for small time local controllability.

Note that the base variables are (x_1, x_2) . Consider a parameterized rectangular closed path γ in the base space with four corner points

$$(0, 0), (x_1, 0), (x_1, x_2), (0, x_2).$$

By evaluating the integral in (20) in closed form for this case, the desired holonomy equations are

$$x_3^1 = x_1(\cos x_2 - 1),$$

$$x_4^1 = x_1 \sin x_2.$$

This equations can be explicitly solved to determine a closed path (or a concatenation of closed paths) γ^* which achieves the desired holonomy. One solution can be given as follows

$$\gamma^* = \begin{cases} \gamma\left(-\frac{(x_3^1)^2 + (x_4^1)^2}{2x_3^1}, -\sin^{-1}\left(\frac{2x_3^1 x_4^1}{(x_3^1)^2 + (x_4^1)^2}\right)\right), & x_3^1 \neq 0, \\ \gamma(0.5x_4^1, 0.5\pi) or \gamma(-0.5x_4^1, -0.5\pi), & x_3^1 = 0. \end{cases}$$

A control function to accomplish Step 1 is given as

$$v_{[0,t_1]}^0 = \begin{cases} \left(\begin{array}{cc} -\frac{\pi x_5^0}{t_1} \cos(\frac{\pi t}{t_1}) & -\frac{\pi x_6^0}{t_1} \cos(\frac{\pi t}{t_1}) \\ -\frac{8\pi x_1^0}{t_1^2} \sin(\frac{2\pi(2t-t_1)}{t_1}) & -\frac{8\pi x_2^0}{t_1^2} \sin(\frac{2\pi(2t-t_1)}{t_1}) \end{array} \right) & 0 \leq t < 0.5t_1, \\ \left(\begin{array}{cc} & \\ & \end{array} \right) & 0.5t_1 \leq t < t_1. \end{cases}$$

There is no loss of generality in assuming that $x_3^1 \neq 0$ so that a single rectangular path γ^* satisfies the desired holonomy conditions. We define the following control functions corresponding to the four segments of γ^* :

$$v_{[t_1,t_2]}^1 = \left(\begin{array}{cc} \frac{2\pi x_1^*}{(t_2-t_1)^2} \sin(\frac{2\pi(t-t_1)}{(t_2-t_1)}) & 0 \end{array} \right),$$

$$v_{[t_2,t_3]}^2 = \left(\begin{array}{cc} 0 & \frac{2\pi x_2^*}{(t_3-t_2)^2} \sin(\frac{2\pi(t-t_2)}{(t_3-t_2)}) \end{array} \right),$$

$$v_{[t_3,t_4]}^3 = \left(\begin{array}{cc} -\frac{2\pi x_3^*}{(t_4-t_3)^2} \sin(\frac{2\pi(t-t_3)}{(t_4-t_3)}) & 0 \end{array} \right),$$

$$v_{[t_4,t_5]}^4 = \left(\begin{array}{cc} 0 & -\frac{2\pi x_4^*}{(t_5-t_4)^2} \sin(\frac{2\pi(t-t_4)}{(t_5-t_4)}) \end{array} \right),$$

where $0 < t_1 \dots < t_5 = T$ are arbitrary. Combining the control functions $v_{[t_k,t_{k+1}]}^k$, $k = 0, 1, 2, 3, 4$, where $t_0 = 0$, we describe

$$v(x^0, t) = \begin{cases} v^0, & t \in [0, t_1], \\ v^1, & t \in [t_1, t_2], \\ v^2, & t \in [t_2, t_3], \\ v^3, & t \in [t_3, t_4], \\ v^4, & t \in [t_4, t_5]. \end{cases} \quad (36)$$

The corresponding control u can be computed using equations (34)-(35). It is clear that the constructed control transfers any initial state of the system to the origin at time $T = t_5$. The resulting motion of the rolling wheel is obtained by solving an initial value problem using the control defined by (36).

We present a representative simulation example, transferring the initial state $(\pi/4, \pi/2, 1, 2, 0, 0)$ to the origin. The time responses for x, y, ϕ, θ are shown in Figure 3. In Figure 4, the configuration of the rolling wheel is shown for a sequence of time instants.

Dynamics of a Planar Interconnection of Three Links
Using Internal Inputs

Another interesting class of physical examples is given by the control of a planar multibody system with angular momentum preserving control torques. For more details on the origin of this problem, and references to previous work, see Krishnaprasad in [7], [8] and Sreenath in [19]. Related papers are by Reyhanoglu and McClamroch in [16], [17]. In this paper, we consider a planar interconnection of three rigid links with joint torques as control inputs as in [17]. We assume that no external moments act on the system, so that the angular momentum of the three links is necessarily a constant. The configuration space of the system is T^3 , the three-dimensional torus. Denote by $\theta = (\theta_1, \theta_2, \theta_3)$ the vector of absolute angles of the links and by $\psi = (\psi_1, \psi_2)$ the vector of relative angles (or joint angles) corresponding to the two joints. The relationship between the vectors θ and ψ is given by $\psi = P\theta$, where

$$P = \begin{bmatrix} -1 & 1 & 0 \\ 0 & -1 & 1 \end{bmatrix}.$$

The equations of motion are given by

$$J(\theta)\ddot{\theta} + F(\theta, \dot{\theta}) = P'u \quad (37)$$

where

$$J(\theta) = \begin{bmatrix} 4 & 3\cos(\theta_2 - \theta_1) & \cos(\theta_3 - \theta_1) \\ 3\cos(\theta_2 - \theta_1) & 8 & 3\cos(\theta_3 - \theta_2) \\ \cos(\theta_3 - \theta_1) & 3\cos(\theta_3 - \theta_2) & 4 \end{bmatrix},$$

$$F(\theta, \dot{\theta}) = \frac{d}{dt}[J(\theta)]\dot{\theta} - \frac{1}{2}\frac{\partial}{\partial\theta}(P'J(\theta)P),$$

and $u = (u_1, u_2)$ is the vector of joint torques. Assuming that the angular momentum is zero, it follows that

$$1'J(\theta)\dot{\theta} = 0 \quad (38)$$

holds, where $1 = (1, 1, 1)'$. Clearly, equation (38) is nonintegrable. The constraint manifold is a five-dimensional manifold and is given by

$$\mathbf{M} = \{(\theta, \dot{\theta}) | 1' J(\theta) \dot{\theta} = 0\}$$

and any configuration is an equilibrium if the controls are zero. It can be shown that equations (37)-(38) constitute a controlled nonholonomic Caplygin system with base space (or shape space) equations that are feedback linearizable.

Following the development in [18], define the variables

$$x_1 = \psi_1, x_2 = \psi_2, x_3 = \theta_1, x_4 = \dot{\psi}_1, x_5 = \dot{\psi}_2$$

so that the normal form equations are given by

$$\dot{x}_1 = x_4,$$

$$\dot{x}_2 = x_5,$$

$$\dot{x}_3 = s_1(x_1, x_2)x_4 + s_2(x_1, x_2)x_5,$$

$$\dot{x}_4 = v_1,$$

$$\dot{x}_5 = v_2,$$

where

$$s_1(x_1, x_2) = -\frac{12 + 3 \cos x_1 + 6 \cos x_2 + \cos(x_1 + x_2)}{16 + 6 \cos x_1 + 6 \cos x_2 + 2 \cos(x_1 + x_2)},$$

$$s_2(x_1, x_2) = -\frac{4 + 3 \cos x_2 + \cos(x_1 + x_2)}{16 + 6 \cos x_1 + 6 \cos x_2 + 2 \cos(x_1 + x_2)},$$

$$v = P J^{-1}(x_1, x_2)[P'u - F(x)]. \quad (39)$$

The following conclusions are based on analysis of the above normal form equations.

Proposition 3. Consider the three link dynamics introduced above. Let $x^e = (x_1^e, x_2^e, x_3^e, 0, 0)$ denote a regular equilibrium of the reduced differential equations corresponding to $u = 0$, i.e. $\frac{\partial s_1(x^e)}{\partial x_2} - \frac{\partial s_2(x^e)}{\partial x_1} \neq 0$. The dynamics of the three link interconnection described by equations (37)-(38) have the following properties:

1. The system is strongly accessible at x^e since the space spanned by the vectors

$g_1, \dots, g_{N-1}, [g_1, f], \dots, [g_{N-1}, f], [g_{j_0}, [f, [g_{j_0}, f]]]$

has dimension 5 at x^e .

2. The system is small time locally controllable at x^e since the brackets satisfy sufficient conditions for small time local controllability.

If the equilibrium solution x^e is not regular, higher order brackets are required to obtain the same conclusions.

Note that the base or shape variables are (x_1, x_2) . Consider a parameterized square closed path γ in the base space with four corner points

$$(0, 0), (z, 0), (z, z), (0, z).$$

The desired holonomy equation is

$$x_3^1 = - \int_0^z (s_1(x, 0) + s_2(z, x) - s_1(x, z) - s_2(0, x)) dx.$$

In this case the loop parameter z^* must be computed numerically, depending on the value of x_3^1 .

A control function to accomplish Step 1 is given as

$$v_{[0, t_1]}^0 = \begin{cases} \left(\begin{array}{cc} -\frac{\pi x_1^0}{t_1} \cos(\frac{\pi t}{t_1}) & -\frac{\pi x_2^0}{t_1} \cos(\frac{\pi t}{t_1}) \\ -\frac{8\pi x_1^0}{t_1^2} \sin(\frac{2\pi(2t-t_1)}{t_1}) & -\frac{8\pi x_2^0}{t_1^2} \sin(\frac{2\pi(2t-t_1)}{t_1}) \end{array} \right) & 0 \leq t < 0.5t_1, \\ \left(\begin{array}{cc} 0 & 0 \end{array} \right) & 0.5t_1 \leq t < t_1. \end{cases}$$

We define the following control functions corresponding to the four segments of γ^* :

$$v_{[t_1, t_2]}^1 = \left(\begin{array}{cc} \frac{2\pi z^*}{(t_2-t_1)^2} \sin(\frac{2\pi(t-t_1)}{(t_2-t_1)}) & 0 \end{array} \right),$$

$$v_{[t_2, t_3]}^2 = \left(\begin{array}{cc} 0 & \frac{2\pi z^*}{(t_3-t_2)^2} \sin(\frac{2\pi(t-t_2)}{(t_3-t_2)}) \end{array} \right),$$

$$v_{[t_3, t_4]}^3 = \left(\begin{array}{cc} -\frac{2\pi z^*}{(t_4-t_3)^2} \sin(\frac{2\pi(t-t_3)}{(t_4-t_3)}) & 0 \end{array} \right),$$

$$v_{[t_4, t_5]}^4 = \left(\begin{array}{cc} 0 & -\frac{2\pi z^*}{(t_5-t_4)^2} \sin(\frac{2\pi(t-t_4)}{(t_5-t_4)}) \end{array} \right),$$

where $0 < t_1 \dots < t_5 = T$ are arbitrary. Combining the control functions $v_{[t_k, t_{k+1}]}^k$, $k = 0, 1, 2, 3, 4$, where $t_0 = 0$, we describe

$$v(x^0, t) = \begin{cases} v^0, & t \in [0, t_1), \\ v^1, & t \in [t_1, t_2), \\ v^2, & t \in [t_2, t_3), \\ v^3, & t \in [t_3, t_4), \\ v^4, & t \in [t_4, t_5]. \end{cases} \quad (40)$$

The corresponding control u can be computed using equations (39). It is clear that the constructed control transfers any initial state of the system to the origin at time $T = t_5$. The resulting motion of the three links is obtained by solving an initial value problem using the control defined by (40).

We present a representative simulation example for a rest-to-rest maneuver. The maneuver is defined by an initial configuration given by the initial state $(0, 0, \pi/4, 0, 0)$ and a final configuration at the origin. In geometric terms, the initial configuration is that all links are oriented at 45° in a straight line as shown in Figure 7; the final configuration is that all links are oriented at 0° in a straight line as shown in Figure 10.

The holonomy function, expressing the integral in the desired holonomy equation as a function of z^* , is shown in Figure 5. The time responses for θ_1 , ψ_1 , ψ_2 are shown in Figure 6. In Figures 7, 8, 9, 10, the configuration of the three links is shown for a sequence of uniformly spaced time instants; the figures are scaled so that the center of mass of the three links is always at the origin. Figure 7 represents the motion along the first segment of the shape space square, Figure 8 represents the motion along the second segment of the shape space square, etc.

Conclusions

A motion planning framework has been established for nonholonomic dynamic systems which are characterized by the fact that control representations include nontrivial drift vector fields. The general approach described in this paper makes substantial use of the geometric approach to nonlinear control. From the theoretical point of view, one of the most significant features of the work presented here is that it provides a guideline for extending recent results on the motion planning problem for nonholonomic systems

without drift to nonholonomic systems with drift. The theoretical issues addressed in the paper and the effectiveness of the proposed motion planning approach have been illustrated through several example problems.

The results in this paper are preliminary. In the examples considered, specific control and motion functions have been constructed based on a specific choice of rectangular loops in the base space. Other loop selections are possible leading to different control and motion functions than what have been described here. It would be natural to formulate an optimal motion planning problem for nonholonomic dynamic systems. Such a formulation would include specification of a performance measure for the motion and it could also include specification of limits on velocities, accelerations, and control inputs. Such optimal motion planning problems would, no doubt, represent a substantial challenge; the results of this paper can be viewed as necessary background for future work in this direction.

References

- [1] V.I. Arnold, *Dynamical Systems III*, Vol.3, Springer-Verlag, 1988.
- [2] A.M. Bloch and N.H. McClamroch,"Control of Mechanical Systems with Classical Nonholonomic Constraints," Proceedings of IEEE Conference on Decision and Control, 1989, Tampa, Fla., 201-205.
- [3] A.M. Bloch, N.H. McClamroch and M. Reyhanoglu,"Controllability and Stabilizability Properties of a Nonholonomic Control System," Proceedings of IEEE Conference on Decision and Control, 1990, Honolulu, Hawaii, 1312-1314.
- [4] A.M. Bloch, M. Reyhanoglu and N.H. McClamroch,"Control and Stabilization of Nonholonomic Dynamic Systems," Submitted to IEEE Transactions on Automatic Control, 1991.
- [5] A.M. Bloch, M. Reyhanoglu and N.H. McClamroch,"Control and Stabilization of Nonholonomic Caplygin Dynamic Systems," Proceedings of IEEE Conference on Decision and Control, 1991, Brighton, UK.

- [6] R.W. Brockett, "Control Theory and Singular Riemannian Geometry", in New Directions in Applied Mathematics, ed. P.J. Hilton and G.S. Young, Springer-Verlag, 1982.
- [7] P.S. Krishnaprasad, "Eulerian Many-Body Problems," in Dynamics and Control of Multibody Systems, eds. J.E. Marsden, P.S. Krishnaprasad and J. Simo, Contemporary Math. Vol.97, AMS, 1989.
- [8] P.S. Krishnaprasad, "Geometric Phases and Optimal Reconfiguration for Multibody Systems," Proceedings of American Control Conference, 1990, San Diego, 2440-2444
- [9] G. Laffarriere and H.J. Sussmann, "Motion Planning for Completely Controllable Systems Without Drift," Proceedings of Conference on Robotics and Automation, 1991, Sacramento, California, 1148-1153.
- [10] Z. Li and J. Canny, "Motion of Two Rigid Bodies with Rolling Constraint," IEEE Transactions on Robotics and Automation, Vol. 6, No. 1, 1990, 62-71.
- [11] Z. Li and R. Montgomery, "Dynamics and Optimal Control of a Legged Robot in Flight Phase," Proceedings of Conference on Robotics and Automation, 1990, Cincinnati, Ohio, 1816-1821.
- [12] J. Marsden, R. Montgomery and T. Ratiu, Reduction, Symmetry and Berry's Phase in Mechanics, Memoirs of the A.M.S., 1990.
- [13] R. Montgomery, "Isoholonomic Problems and Some Applications," Communications in Mathematical Physics, Vol. 128, 1990, 565-592.
- [14] R. Murray and S. Sastry, "Steering Nonholonomic Systems Using Sinusoids," Proceedings of IEEE Conference on Decision and Control, 1990, Honolulu, Hawaii, 2097-2101.
- [15] Ju. I. Neimark and F.A. Fufaev, Dynamics of Nonholonomic Systems, Vol. 33, A.M.S. Translations of Mathematical Monographs, A.M.S., 1972.
- [16] M. Reyhanoglu and N. H. McClamroch, "Controllability and Stabilizability of Planar Multibody Systems with Angular Momentum Preserving Control Torques," 1991 American Control Conference, Boston, 1102-1107.

- [17] M. Reyhanoglu and N. H. McClamroch, "Reorientation of Space Multibody Systems Maintaining Zero Angular Momentum," 1991 AIAA Guidance, Navigation and Control Conference, New Orleans, 1330-1340.
- [18] A. Shapere and F. Wilczek, Geometric Phases in Physics, Wold Scientific, 1988.
- [19] N. Sreenath, "Nonlinear Control of Multibody Systems in Shape Space," Proceedings of IEEE Conference on Robotics and Automation, 1990, Cincinnati, Ohio, 1776-1781.
- [20] H.J. Sussmann, "A General Theorem on Local Controllability," SIAM Journal on Control and Optimization, Vol. 25, No. 1, 1987, 158-194.
- [21] H.J. Sussmann and V. Jurdjevic, "Controllability of Nonlinear Systems," J. Differential Equations, 12, 1972, 95-116.
- [22] A. M. Vershik and V. Ya. Gershkovich, " Nonholonomic Problems and the Theory of Distributions," Acta Applicandae Mathematicae, 12, 1988, 181-209.

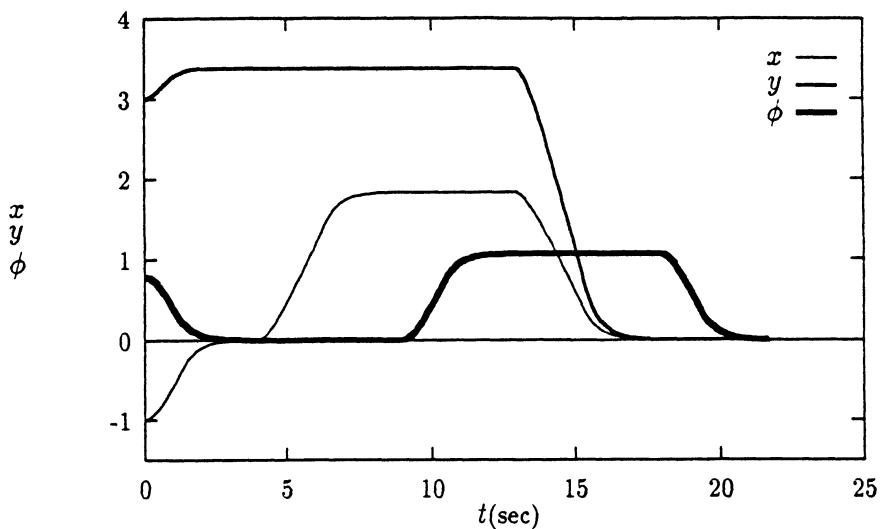


Figure 1:

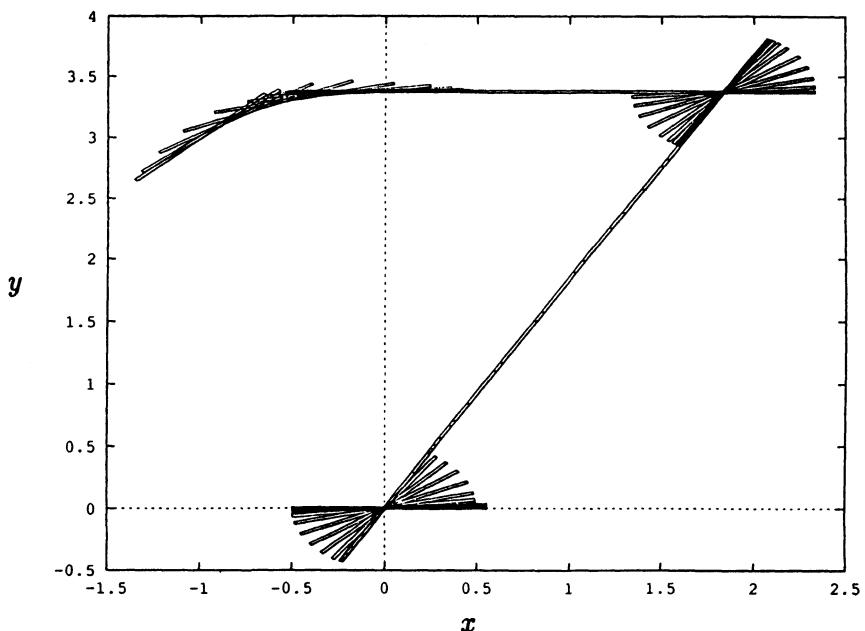


Figure 2:

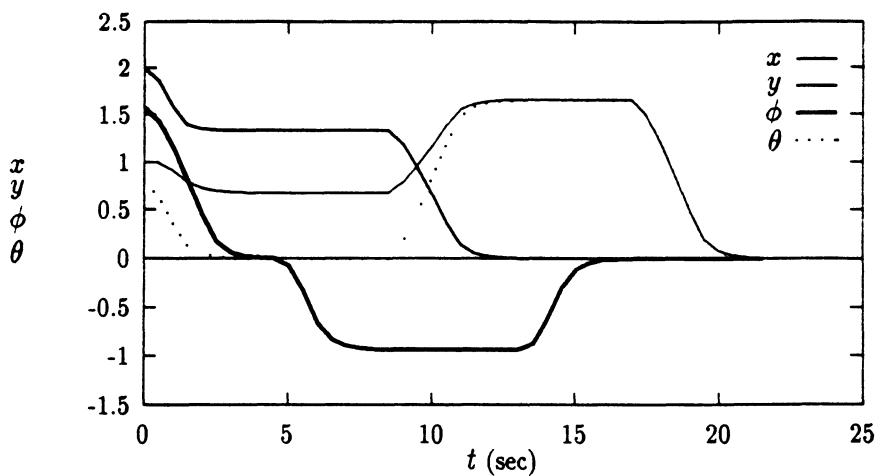


Figure 3:

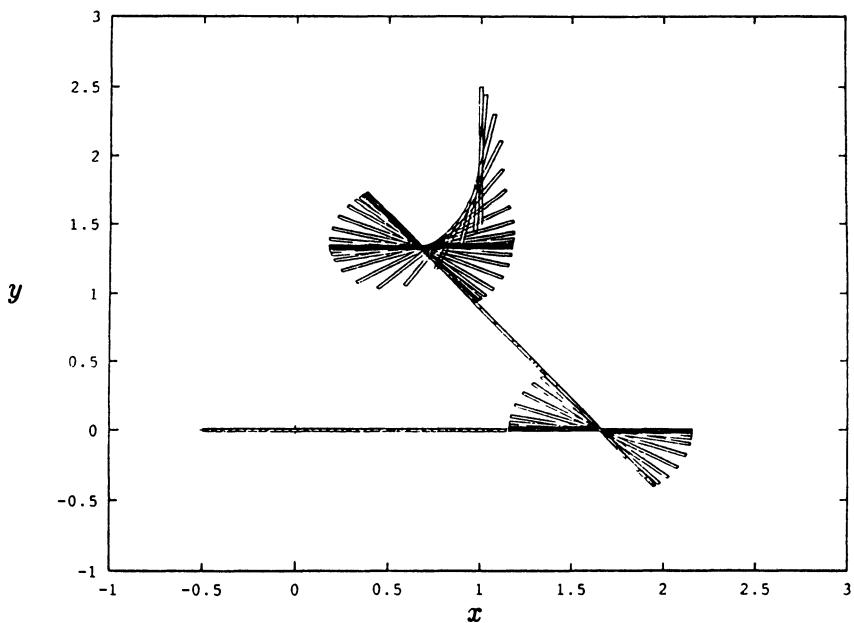


Figure 4:

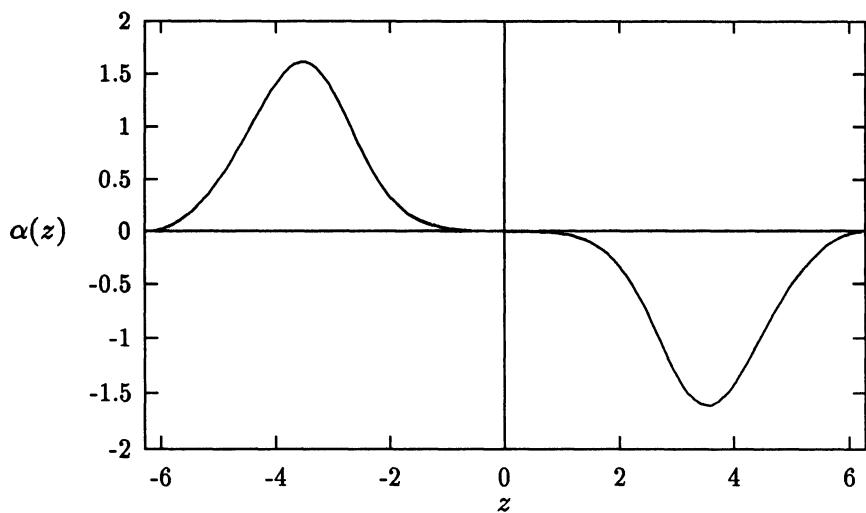


Figure 5:

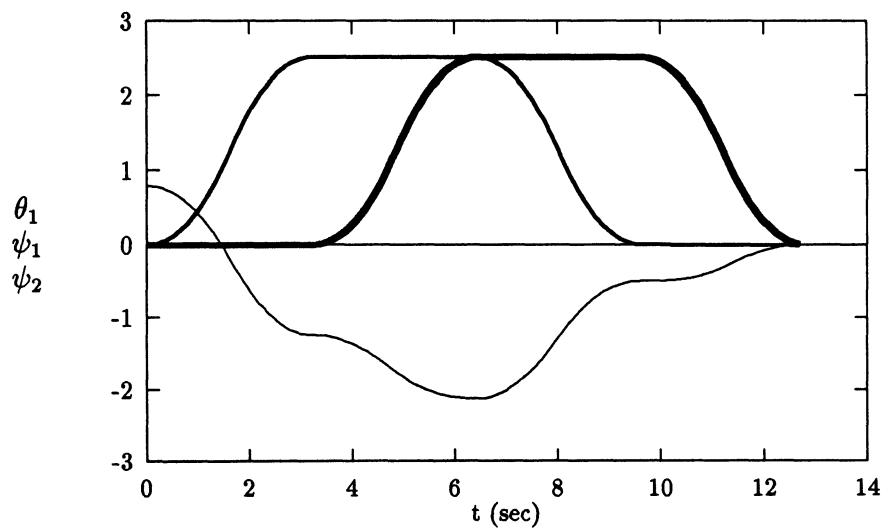


Figure 6:

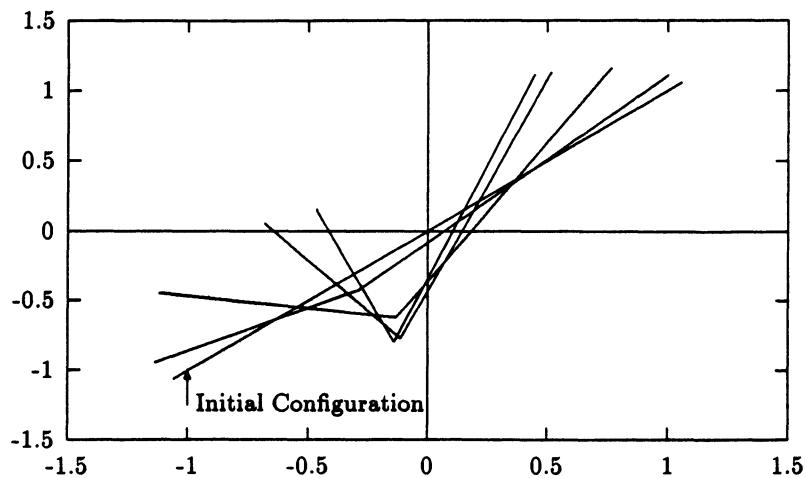


Figure 7:

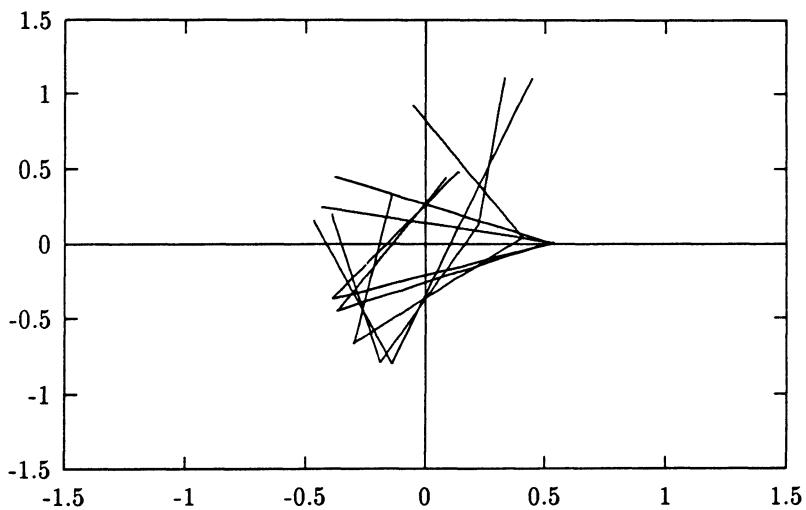


Figure 8:

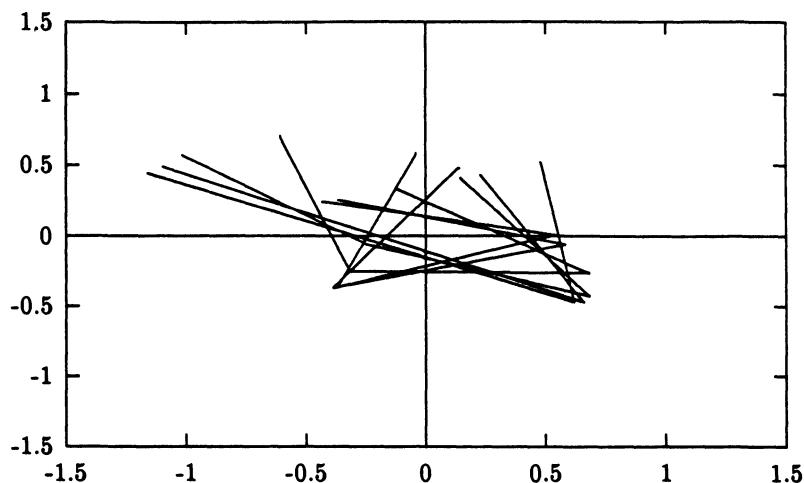


Figure 9:

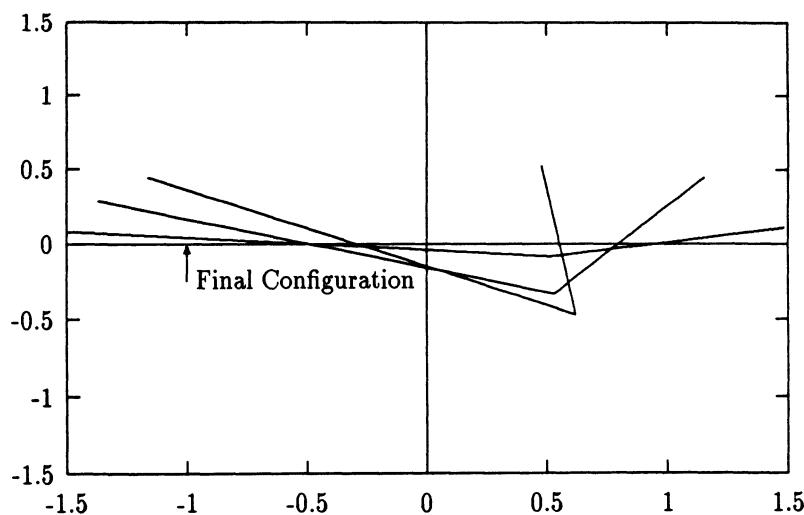


Figure 10:

7

A Differential Geometric Approach to Motion Planning

Gerardo Lafferriere

Dept. of Mathematics

Portland State University

Portland, OR 97207-0751

Héctor J. Sussmann*

Dept. of Mathematics

Rutgers University

New Brunswick, NJ 08903

Abstract

We propose a general strategy for solving the motion planning problem for real analytic, controllable systems without drift. The procedure starts by computing a control that steers the given initial point to the desired target point for an extended system, in which a number of Lie brackets of the system vector fields are added to the right-hand side. The main point then is to use formal calculations based on the product expansion relative to a P. Hall basis, to produce another control that achieves the desired result on the formal level. It then turns out that this control provides an exact solution of the original problem if the given system is nilpotent. When the system is not nilpotent, one can still produce an iterative algorithm that converges very fast to a solution. Using the theory of feedback nilpotentization, one can find classes of non-nilpotent systems for which the algorithm, in cascade with a precompensator, produces an exact solution in a finite number of steps. We also include results of simulations which illustrate the effectiveness of the procedure.

*Work supported in part by the National Science Foundation under NSF Grant DMS-8902994.

Introduction

We present a general strategy for solving the Motion Planning Problem (MPP) for controllable systems without drift. Following the line of research initiated by Brockett and Sastry (cf. e.g. [Bro81], [MS90a]), we propose an approach based on tools from “differential geometric control theory,” and making systematic use of Lie brackets of vector fields and Lie algebraic properties.

We consider control systems:

$$\Sigma : \quad \dot{x} = u_1 f_1(x) + \dots + u_m f_m(x) \quad (1)$$

where: (1) f_1, \dots, f_m are *real analytic* vector fields on \mathbb{R}^n , and (2) Σ is completely controllable. Notice that the right-hand side of (1) does not contain a term of the form $f_0(x)$, that is, we are assuming that *there is no drift*. It is well known that Condition (2) is equivalent to the LARC (Lie algebra rank condition), i.e. to the condition that $L(\mathbf{f})$ spans \mathbb{R}^n at each point, where $L(\mathbf{f})$ is the *controllability Lie algebra* of Σ , i.e. the Lie algebra of vector fields generated by $\mathbf{f} = \{f_1, \dots, f_m\}$

The MPP (Motion Planning Problem) is the problem of finding reasonable algorithms producing, for every pair p, q of points, an open-loop control

$$t \longrightarrow u(t) = (u_1(t), \dots, u_m(t)) \quad (2)$$

that steers p to q .

The controllability condition LARC guarantees that for any given p and q infinitely many such controls exist. Our goal here is to exhibit an explicit algorithm that will compute one in a reasonably simple way. Naturally, what will be considered “simple” depends on the particular situation, and many solutions are possible. Moreover, one may be interested in at least two kinds of simplicity, namely, the simplicity of the computation or that of the control it produces. Here our primary objective will be to describe a computationally simple solution and present data from simulations. The resulting

controls will be bang-bang with a simple structure but involving a possibly large number of “moves.” By refining our methods it is possible to obtain algorithms that involve more elaborate computations but yield “nicer” controls —e.g. bang-bang with fewer moves, finite superpositions of sinusoids as suggested by Murray and Sastry in [MS90b], or polynomial functions of time.

The MPP has been studied by a number of authors, notably Sastry, Hauser, Murray, Li et al., cf. [HSK89], [MS90b], [MS90a], [SL89]. They have proposed procedures that work very well in a number of special cases. Often, these procedures: (a) attempt to use optimal control; (b) require special assumptions (e.g. that for every k the span of the Lie brackets up to order k has constant dimension); (c) work in special cases.

The strategy proposed here is a modification of the approach mentioned above, in that: (a) it does not in principle require special assumptions on the spans of the Lie brackets; (b) it does not use optimal control; (c) it works exactly for *nilpotent* and *nilpotentizable systems* (defined below); (d) it works approximately for completely general systems, and can be used to produce a “successive approximations” algorithm that converges quite fast to an exact solution.

The Strategy

The main point of the strategy proposed here is to consider, in conjunction with the original system Σ , an *extended system*

$$\Sigma_e : \quad \dot{x} = v_1 f_1(x) + \dots + v_m f_m(x) + v_{m+1} f_{m+1}(x) + \dots + v_r f_r(x), \quad (3)$$

where f_{m+1}, \dots, f_r are higher-order Lie brackets of the f_i , chosen so that $f_1(x), \dots, f_r(x)$ span \mathbb{R}^n for all x , or at least for all x in some prescribed bounded region \mathcal{R} .

Remark 1 Given any bounded region \mathcal{R} , the LARC implies, by a simple compactness argument, that there exists a finite collection

$\{f_{m+1}, \dots, f_r\}$ of brackets of the f_i such that $f_1(x), \dots, f_r(x)$ span \mathbb{R}^n for all $x \in \mathcal{R}$. However, n brackets will in general not suffice, as shown by simple examples. (Consider, e.g., the vector fields in \mathbb{R}^2 given by $f_1 = (1, 0)$, $f_2 = (0, \sin x)$. Let \mathcal{R} be any region that contains a horizontal segment of length π . Then every linear combination of brackets of the f_i is parallel to f_1 at some point of \mathcal{R} , so at least three brackets are needed to span \mathbb{R}^2 at all points of \mathcal{R} .) For an unbounded region, it may even happen that there is no finite set of brackets that spans at every point. However, in many cases (e.g. for polynomial vector fields) the existence of a globally spanning finite set can be proved.

Remark 2 We prefer to allow r to be larger than n for two reasons. First because, as explained above, if we are interested in a fixed bounded region \mathcal{R} , there may not exist a fixed set of n brackets that spans \mathbb{R}^n at every point of \mathcal{R} . Second, because by taking $r > n$ it may become possible to get a better conditioned system for the computation of v , as explained below.

Once we have selected an extended system Σ_e , the solution proposed here of the MPP involves two basic steps:

STEP I: Find a control v that steers p to q for Σ_e .

STEP II: Use v to compute a control u that steers p to q for Σ .

Step I is in principle easy. In fact, to find v , we begin by choosing a C^1 curve $\gamma : [0, T] \rightarrow \mathbb{R}^n$ that goes from p to q . A straight-line segment will work perfectly well, although if we are dealing with, for example, an obstacle avoidance problem, other curves might be more suitable. Having chosen γ , we then write the tangent vector $\dot{\gamma}(t)$ as a linear combination of $f_1(\gamma(t)), \dots, f_r(\gamma(t))$. The resulting coefficients are the $v_i(t)$. The problem of expressing $\dot{\gamma}(t)$ as a linear combination of $f_1(x), \dots, f_r(x)$ involves inverting a square matrix (if $r = n$) or computing a pseudoinverse.

Step II is harder and involves some interesting algebra, that will be discussed in detail below.

Some terminology

We recall that a Lie algebra L is said to be *nilpotent* if there is an integer $k > 0$ with the property that all the Lie brackets $[v_1, [v_2, \dots, [v_k, v_{k+1}] \dots]]$ vanish. The smallest such k is the *order of nilpotency* of L , and L is said to be *nilpotent of order k* .

The system Σ will be called *nilpotent* if its controllability Lie algebra $L(\mathbf{f})$ is a nilpotent Lie algebra.

We use $A(X_1, \dots, X_m)$ to denote the algebra of noncommutative polynomials in (X_1, \dots, X_m) . With the Lie bracket defined by $[P, Q] = PQ - QP$, $A(X_1, \dots, X_m)$ is also a Lie algebra. We then define $L(X_1, \dots, X_m)$ to be the Lie subalgebra of $A(X_1, \dots, X_m)$ generated by X_1, \dots, X_m . The elements of $L(X_1, \dots, X_m)$ are known as *Lie polynomials* in X_1, \dots, X_m . We also use $\hat{A}(X_1, \dots, X_m)$ and $\hat{L}(X_1, \dots, X_m)$ to denote, respectively, the set of *noncommutative formal power series* and that of *Lie series* in the X_i . That is, $\hat{A}(X_1, \dots, X_m)$ consists of all (not necessarily finite) linear combinations $\sum_M c_M M$ of monomials, and $\hat{L}(X_1, \dots, X_m)$ is the set of those $S \in \hat{A}(X_1, \dots, X_m)$ such that, for each j , the j -th homogeneous component of S is in $L(X_1, \dots, X_m)$.

The nilpotent versions $A_k(X_1, \dots, X_m)$, $L_k(X_1, \dots, X_m)$ (also denoted A_k^m , L_k^m) are then defined by just killing all the monomials of degree $k + 1$. Then A_k^m and L_k^m are known, respectively, as the *free nilpotent associative algebra of order k* and the *free nilpotent Lie algebra of order k* .

Writing down a basis of A_k^m is always very easy, for all one has to do is list all the possible monomials of degree $\leq k$. Writing down a basis of L_k^m is harder because the formal brackets are not linearly independent, due to the linear relations among them arising from skew-symmetry and the Jacobi identity.

The concept of a *P. Hall basis* is a particular way to get around this difficulty and select a basis. For a definition of a P. Hall basis see [Bat78], [Sus87], or [Sus83]. Here we just give an example.

Example 1. Let $m = 2$ as before, and let the indeterminates be X, Y . Let $k = 5$. Then, writing $[AB]$ rather than $[A, B]$, the

following brackets form a P. Hall basis of $L(X, Y)$:

$X, Y, [XY], [X[XY]], [Y[XY]], [X[X[XY]]], [Y[X[XY]]], [Y[Y[XY]]], [X[X[X[XY]]]], [Y[X[X[XY]]]], [Y[Y[X[XY]]]], [Y[Y[Y[XY]]]], [[XY][X[XY]]], \text{ and } [[XY][Y[XY]]]$.

Step II

It will be important to restrict our choice of the extended system a bit further, by requiring the “new” f_i to be brackets arising from a P. Hall basis. Step II then proceeds as follows. We choose k such that all the $f_i, i > m$ have degree $\leq k$. Then:

- II. We do *formal calculations* on $L_k(X_1, \dots, X_m)$. The X_i are purely formal *noncommuting indeterminates*. The calculations are done in two steps:
 - IIa: Solve a simple differential equation (called the “formal extended equation”) on the Lie group $G_k(X_1, \dots, X_m)$ (see below), using P. Hall coordinates.
 - IIb: Find u from the P. Hall coordinates obtained in IIa.

P. Hall Coordinates

If P is any element of $\hat{A}(X_1, \dots, X_m)$ or $A_k(X_1, \dots, X_m)$ that has no constant term, then the exponential e^P is well defined by means of the usual power series. We define

$$\begin{aligned}\hat{G}(X_1, \dots, X_m) &= \{e^Z : Z \in \hat{L}(X_1, \dots, X_m)\}, \\ G_k(X_1, \dots, X_m) &= \{e^Z : Z \in L_k(X_1, \dots, X_m)\},\end{aligned}\quad (4)$$

and use the shorter notation G_k^m for $G_k(X_1, \dots, X_m)$. (Examples of elements of $\hat{G}(X, Y)$: $e^X = 1 + X + \frac{1}{2}X^2 + 1/6X^3 + \dots, e^{[X,Y]} = 1 + [X, Y] + \frac{1}{2}[X, Y]^2 + \dots, e^{X-Y+3[X,Y]} = 1 + X - Y + 3[X, Y] + \dots$. Examples of elements of $G_2(X, Y)$: $e^X = 1 + X + \frac{1}{2}X^2, e^{[X,Y]} =$

$1 + [X, Y]$, $e^{X-Y+3[X,Y]} = 1 + X - Y + 3[X, Y] + \frac{1}{2}(X - Y)^2$. (Notice that the meaning of e^P depends on whether P is regarded as an element of $\hat{G}(X_1, \dots, X_m)$ or of $G_k(X_1, \dots, X_m)$.) Then both $\hat{G}(X_1, \dots, X_m)$ and G_k^m are closed under multiplication, thanks to the Campbell-Hausdorff Formula. Moreover, since $e^{-Z}e^Z = 1$, they are both groups. In fact, G_k^m is the connected simply connected Lie group with Lie algebra L_k^m . We call G_k^m the *free nilpotent Lie group of order k with m infinitesimal generators*.

We now let B_1, B_2, \dots, B_s be a P. Hall basis of $L_k(X_1, \dots, X_m)$. Then it is well known that every $S \in G_k(X_1, \dots, X_m)$ has unique expressions

$$S = e^{h_s B_s} e^{h_{s-1} B_{s-1}} \dots e^{h_2 B_2} e^{h_1 B_1}, \quad (5)$$

$$S = e^{\tilde{h}_1 B_1} e^{\tilde{h}_2 B_2} \dots e^{\tilde{h}_{s-1} B_{s-1}} e^{\tilde{h}_s B_s}. \quad (6)$$

The maps $S \rightarrow (h_1, \dots, h_s)$ and $S \rightarrow (\tilde{h}_1, \dots, \tilde{h}_s)$ are global coordinate charts on G_k^m , and establish global diffeomorphisms between G_k^m and \mathbb{R}^s . The h_i, \tilde{h}_i are, respectively, the *backward* and *forward P. Hall coordinates* of S .

The formal extended equation

We are now ready to give more details about Step IIa. We fix a P. Hall basis of the Lie algebra $L_k(X_1, \dots, X_m)$, where k is chosen as explained above. We let \mathbf{E}_f be the “evaluation map” that assigns to each $P \in L(X_1, \dots, X_m)$ the vector field obtained by plugging in the $f_i, i = 1, \dots, m$, for the corresponding X_i . (For example, $\mathbf{E}_f([X_3, [X_1, [X_2, X_3]]]) = [f_3, [f_1, [f_2, f_3]]]$.) We assume that the vector fields f_{m+1}, \dots, f_r are given by $f_j = \mathbf{E}_f(X_j)$ for $j = m + 1, \dots, r$, where X_{m+1}, \dots, X_r are such that X_1, \dots, X_r is a P. Hall basis of L_k^m .

We then consider the initial value problem

$$\begin{aligned} \Sigma_{fe} : \quad \dot{S}(t) &= S(t)(v_1(t)X_1 + \dots + v_r(t)X_r), \\ S(0) &= 1. \end{aligned} \quad (7)$$

We will refer to the first equation of (7) as the “formal extended equation.” We regard it as evolving in the algebra A_k^m but observe that if the initial condition is in the group G_k^m —as is the case for the problem (7)— then the solution will also evolve in G_k^m .

We express the solution of (7) in *backward P. Hall coordinates*. That is, we write the solution as a product

$$S(t) = e^{h_s(t)B_s}e^{h_{s-1}(t)B_{s-1}} \dots e^{h_2(t)B_2}e^{h_1(t)B_1}. \quad (8)$$

The advantage of using backward P. Hall coordinates is that *the functions $h_j(t)$ are easily computed by solving a system of O.D.E.’s with input v* . This only requires successive integrations and algebraic operations. The proof that the computation of the h_i can be done by integrations was given in [Sus86] for the simpler case in which the system Σ_{fe} is not extended. An almost equally simple proof works for the extended case. (The details will be given in [SL].) The following example illustrates what happens in general.

Example 2. Say $k = 3$, $m = 2$, $r = 4$, $f_3 = [f_1, f_2]$, $f_4 = [f_1, [f_1, f_2]]$. We can take $B_1 = X$, $B_2 = Y$, $B_3 = [X, Y]$, $B_4 = [X, [X, Y]]$, $B_5 = [Y, [X, Y]]$. Then $X_3 = [X, Y]$, $X_4 = [X, [X, Y]]$, $X_5 = [Y, [X, Y]]$. Since $r = 4$, we solve (7) with $v_5 \equiv 0$. Then $h_1(t), \dots, h_5(t)$ are computed by solving

$$\begin{aligned} \dot{h}_1 &= v_1, \\ \dot{h}_2 &= v_2, \\ \dot{h}_3 &= h_1 v_2 + v_3, \\ \dot{h}_4 &= \frac{1}{2} h_1^2 v_2 + h_1 v_3 + v_4, \\ \dot{h}_5 &= h_2 v_3 + h_1 h_2 v_2, \\ h_1(0) &= h_2(0) = h_3(0) = h_4(0) = h_5(0) = 0. \end{aligned} \quad (9)$$

Remark 3 Equation (7) is formally analogous to the equation that defines the “Chen series,” introduced in control theory by M. Fliess (cf. [Fli83]), although it differs from it in two ways: (a) in the definition of the Chen series one does not have the extra brackets X_{m+1}, \dots, X_r , and (b) (7) is interpreted as evolving in A_k^m rather than in the algebra of formal power series.

Inverting the non-extended system

We now proceed to Step IIb. We consider the problem

$$\begin{aligned}\dot{S}^*(t) &= S^*(t)(u_1(t)X_1 + \dots + u_m(t)X_m) , \\ S^*(0) &= 1 ,\end{aligned}\quad (10)$$

which involves a differential equation (called the “formal equation”) similar to the formal extended equation, except that it is *not* extended, i.e. only the indeterminates X_i , $i = 1, \dots, m$, but no high-order brackets, appear in the right-hand side.

The goal of Step IIb is to solve the *inverse problem* of finding a control $u : [0, T] \rightarrow \mathbb{R}^m$ such that $S^*(T) = S(T)$. That is:

P given P . Hall coordinates h_1, h_2, \dots, h_s of an element S of $G_k(X_1, \dots, X_m)$, find a control $t \mapsto u(t)$ that gives rise to an $S^*(T)$ that has these coordinates.

Problem **P** can be solved in a number of ways. Here we describe one of several possible solutions. The main idea is to solve separately for each exponential factor, and then concatenate the results. It turns out that this is quite easy to do, if we only ask to do it “modulo factors of higher order.” Moreover, it is somewhat simpler to do the calculations using forward rather than backward P. Hall coordinates. So a full description of the algorithm would require a step intermediate between IIa and IIb, in which the expression of $S(T)$ in terms of backward P. Hall coordinates—which is much easier to derive—is transformed into an expression in forward coordinates. This intermediate step amounts to a simple algebraic transformation, and we will omit the details.

We illustrate the calculation of u by means of the following
Example 3. Find u for $S(T) = e^{\alpha X} e^{\beta Y} e^{\gamma [X, Y]} e^{\delta [X, [X, Y]]} e^{\varepsilon [Y, [X, Y]]}$, assuming that $k = 3$.

To solve this, fix once and for all open-loop controls A, B that give rise to e^X, e^Y . Use xA for “ x times A ,” and so on. Use $\#$ for concatenation. (So $A \# B$ means ‘ A followed by B .’) Then:

- $\alpha A \# \beta B$ gives rise to $e^{\alpha X} e^{\beta Y}$
- $C = \sqrt{\gamma} A \# \sqrt{\gamma} B \# (-\sqrt{\gamma} A) \# (-\sqrt{\gamma} B)$ “almost” gives rise to $e^{\gamma[X,Y]}$. (We are assuming that $\gamma > 0$. If $\gamma < 0$ we just interchange X and Y .) Precisely, C gives rise to the product $e^{\gamma[X,Y]} e^{\tilde{\gamma}[X[X,Y]]} e^{-\tilde{\gamma}[Y[X,Y]]}$, where $\tilde{\gamma} = \frac{1}{2}\gamma^{\frac{3}{2}}$.

This implies that $\alpha A \# \beta Y \# \sqrt{\gamma} A \# \sqrt{\gamma} B \# (-\sqrt{\gamma} A) \# (-\sqrt{\gamma} B)$ gives rise to the product $e^{\alpha X} e^{\beta Y} e^{\gamma[X,Y]} e^{\tilde{\gamma}[X,[X,Y]]} e^{-\tilde{\gamma}[Y,[X,Y]]}$. But what we really want is $e^{\alpha X} e^{\beta Y} e^{\gamma[X,Y]} e^{\delta[X,[X,Y]]} e^{\varepsilon[Y,[X,Y]]}$. To achieve this, it suffices to find a control that gives rise to $e^{\tilde{\delta}[X,[X,Y]]} e^{\tilde{\varepsilon}[Y,[X,Y]]}$, where $\tilde{\delta} = \delta - \tilde{\gamma}$, $\tilde{\varepsilon} = \varepsilon + \tilde{\gamma}$. For this we could use

$$2\rho A \# \rho B \# (-\rho A) \# (-\rho B) \# (-\rho A) \# \rho B \# \rho A \# (-\rho B) \# \\ (-\rho A) \# \sigma B \# \sigma A \# \sigma B \# (-\sigma A) \# (-\sigma B) \# \sigma A \# (-\sigma B) \# (-\sigma A),$$

where $\rho = \tilde{\delta}^{\frac{1}{3}}$, $\sigma = \tilde{\varepsilon}^{\frac{1}{3}}$. Since we are working in the group $G_3(X, Y)$, the higher order terms vanish.

The end result is a control u made of 26 pieces that solves our problem exactly. In other words, $S(T)$ is realized in 26 moves.

The main theorems

The following theorems show that the control u computed as above is in fact what we need. Recall that a vector field is *complete* if its integral curves are defined for all times.

Theorem 1 *If $L(f)$ is nilpotent of order k and the vector fields f_1, \dots, f_m are complete, then the u computed by the above method steers p to q exactly.*

Remark 4 If the f_i are not complete then it may happen that the trajectory for a given control u is not defined for all t and, even if it is defined, the control may fail to steer p to q .

Theorem 2 For a general system (1), let $\sigma(p, q, u)$ be the point to which u steers p . Then, on any bounded region \mathcal{R} , the error $E(p, q) = \|\sigma(p, q, u) - q\|$ satisfies a bound

$$E(p, q) \leq F(\Delta)\Delta^\theta, \quad (11)$$

for all $p, q \in \mathcal{R}$, where $\Delta = \|p - q\|$, $\theta = 1 + \frac{1}{k}$, and $F : [0, \infty) \rightarrow [0, \infty]$ is a function which is finite and bounded for Δ near 0.

Remark 5 The function F is allowed to be infinite if Δ is sufficiently large. This is because in Theorem 2 we are not assuming completeness, so it may in fact happen that the trajectory for u is not even defined for all t .

Corollary 1 If \mathcal{R} is a bounded region, then there exists $\Delta > 0$ such that, if $p, q \in \mathcal{R}$ and $\|p - q\| \leq \Delta$, then $E(p, q) \leq \frac{1}{2}\|p - q\|$.

Definition 1 The supremum of all the numbers Δ that satisfy the condition of Corollary 1 is called the *critical distance* for \mathcal{R} , and is denoted by $\Delta_c(\mathcal{R})$. Any number $\bar{\Delta}$ such that $0 < \bar{\Delta} \leq \Delta_c(\mathcal{R})$ will be called an *admissible step length* for \mathcal{R} .

The iterated algorithm

So far, we have explained how to solve the MPP if the system is exactly nilpotent. For a general system, the control u obtained by the prescription of the preceding sections is only an approximate solution. It turns out, however, that we can use our method to obtain a control that steers p to q within any arbitrary prescribed error. This is done by *iterating* the algorithm, and using always distances not greater than the critical distance $\Delta_c(\mathcal{R})$.

The iterated algorithm works as follows. We work in a fixed bounded region \mathcal{R} , and fix once and for all an admissible step length $\bar{\Delta}$ for \mathcal{R} . Suppose we are given p, q , and an upper bound ε for the permissible error. Then:

1. set $p_0 = p$, $i = 0$, and go to **2**.
2. Apply the algorithm to go from p_i to q_i , where $q_i = p_i + \min\left(1, \frac{\bar{\Delta}}{\|q - p_i\|}\right)(q - p_i)$. Let p_{i+1} be the resulting point.
3. If $\|p_{i+1} - q\| \leq \varepsilon$ stop. Else $i := i + 1$ and go to **2**.

We refer to the above as the *iterated algorithm*.

Theorem 3 *Let \mathcal{R} be a bounded region. Let Δ_c be the critical distance for \mathcal{R} , and let $\bar{\Delta}$ be such that $0 < \bar{\Delta} \leq \Delta_c$. Assume $\|p - q\| = \nu \bar{\Delta}$. Then the iterated algorithm stops in at most $2[\nu] + \log_2(\frac{\bar{\Delta}}{\varepsilon})$ steps.*

Feedback Nilpotentization

Recall that a *feedback transformation* of Σ consists of a linear change of controls $u_i = \sum_{j=1}^m \beta_{ij}(x)v_j$, such that $\beta(x)$ is a nonsingular matrix for each x , and $\beta(x)$ is smooth as a function of x .

Let us call a system Σ *feedback nilpotentizable* (FN) if it can be made nilpotent by a feedback transformation.

Feedback nilpotentization was studied by H. Hermes, A. Lundell and D. Sullivan in [Her89] and [HLS84]. They proved, for instance, that any system

$$\dot{x} = u_1 f_1(x) + u_2 f_2(x), \quad x \in \mathbb{R}^3 \quad (12)$$

for which $f_1(\bar{x}), f_2(\bar{x}), [f_1, f_2](\bar{x})$ are linearly independent is feedback nilpotentizable on a neighborhood of \bar{x} . In many cases, the nilpotentization can be carried out explicitly. Examples of several nilpotentizable systems will be shown below.

Results from simulations

We consider first the problem of steering a unicycle (Figure 1).

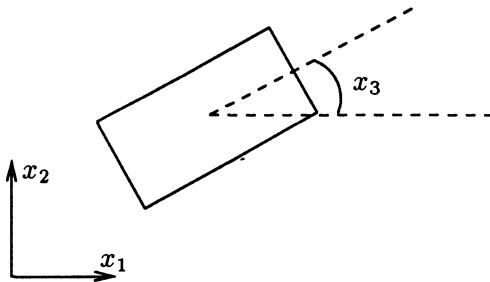


Figure 1: Unicycle.

The controls are the driving speed and the steering speed. The equations for this system are

$$\begin{aligned}\dot{x}_1 &= \cos(x_3)u_1, \\ \dot{x}_2 &= \sin(x_3)u_1, \\ \dot{x}_3 &= u_2.\end{aligned}\tag{13}$$

where (x_1, x_2) are the Cartesian coordinates of the center of the unicycle and x_3 is the angle its main axis makes with the x_1 -axis. We can rewrite this system as $\dot{x} = u_1 f_1(x) + u_2 f_2(x)$ where

$$f_1(x) = (\cos(x_3), \sin(x_3), 0) \text{ and } f_2(x) = (0, 0, 1).$$

Clearly the vectors $f_1(x), f_2(x), [f_1, f_2](x)$ span \mathbb{R}^3 near $x = 0$, so (13) is locally nilpotentizable as mentioned earlier in § . However, (13) is not nilpotent. In fact, $\text{ad}_{f_2}^{2n}(f_1) = (-1)^n(\cos(x_3), \sin(x_3), 0)$ for $n \geq 0$.

We generated first an approximate trajectory for the system using the nilpotent formal system of order 2. In this case $e^{[X,Y]} = e^X e^Y e^{-X} e^{-Y}$, and the formal solution can be written as $S(T) = e^{\alpha_5 X} e^{\alpha_4 Y} e^{\alpha_3 X} e^{\alpha_2 Y} e^{\alpha_1 X}$. We can realize this product of exponentials by choosing a piecewise constant control made up of 5 pieces, each piece generating a move corresponding to a trajectory of either f_1 or f_2 . However, the controls need not be piecewise constant. We could use instead controls $u_1(t) = -6\alpha_5(t-1)$, $u_2 = 0$ for time 1,

etc.. After concatenation we would then get a continuous control. Clearly using higher order polynomials we can obtain as smooth a control as desired.

Specifically, we tried to drive the system from $x_0 = (0, 0, 0)$ to $x_f = (2, 1, 0)$. As reference trajectory to compute the v 's we used the straight-line segment between the two points parametrized from 0 to 1. The resulting controls are plotted in Figure 2 (bang-bang on top and continuous, piecewise-polynomial on the bottom).

After two iterations the trajectory reaches the target within an error $\varepsilon \leq 0.04$. (The bottom plot represents the unicycle drawn at discrete times along the trajectory.) The backward P. Hall coordinates for our choice of v 's are $(-1, 0, 2)$ for the first iteration and $(-0.16, 0, -0.46)$ for the second. (We are using an order 2 approximation.)

This procedure corresponds to guessing that the critical distance is larger than $\|x_0 - x_f\|$. If, however, we start with a smaller guess $\Delta_g = 0.5$ for the critical distance, then the final trajectory stays closer to the segment from x_0 to x_f but it takes 5 steps to reach the desired target. This is still much smaller than the upper bound of 11 predicted by Theorem 2. Figure 4 shows the result of such a simulation.

Since this system is nilpotentizable we can actually get to the target point in one step exactly. More precisely, this system can be made nilpotent using the following feedback

$$\begin{aligned} u_1 &= \frac{1}{\cos(x_3)} w_1 , \\ u_2 &= \cos^2(x_3) w_2 . \end{aligned} \tag{14}$$

The system then becomes

$$\begin{aligned} \dot{x}_1 &= w_1 , \\ \dot{x}_2 &= \tan(x_3) w_1 , \\ \dot{x}_3 &= \cos^2(x_3) w_2 . \end{aligned} \tag{15}$$

which is nilpotent of order 2. We therefore compute the desired controls as follows. First apply the procedure to the nilpotent system

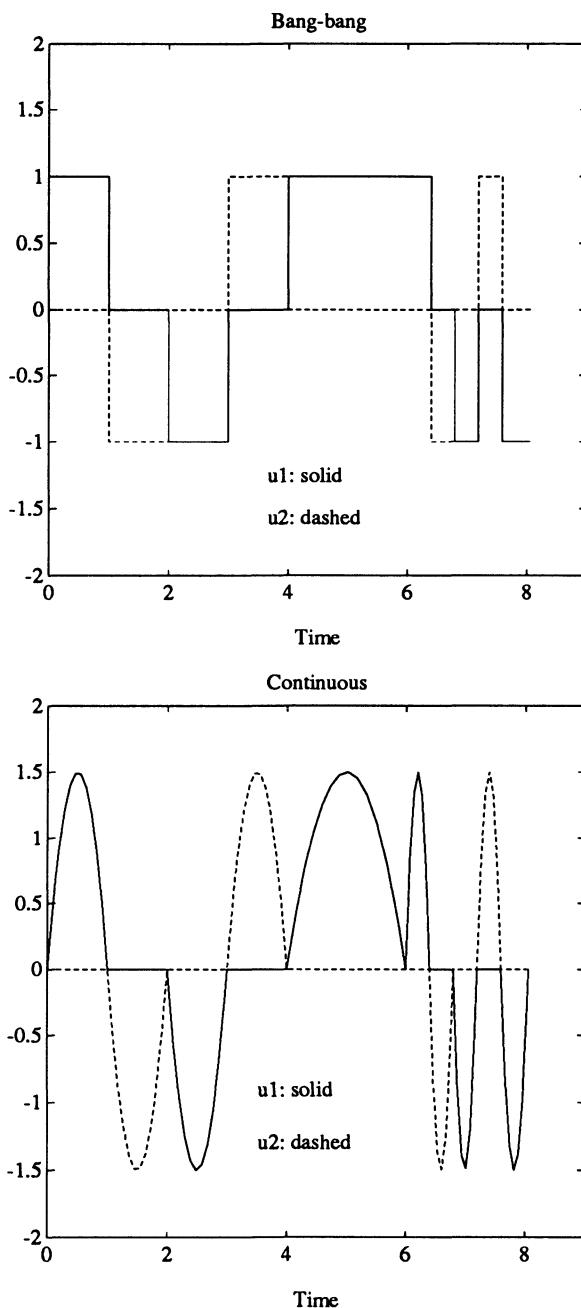


Figure 2: Controls for unicycle.

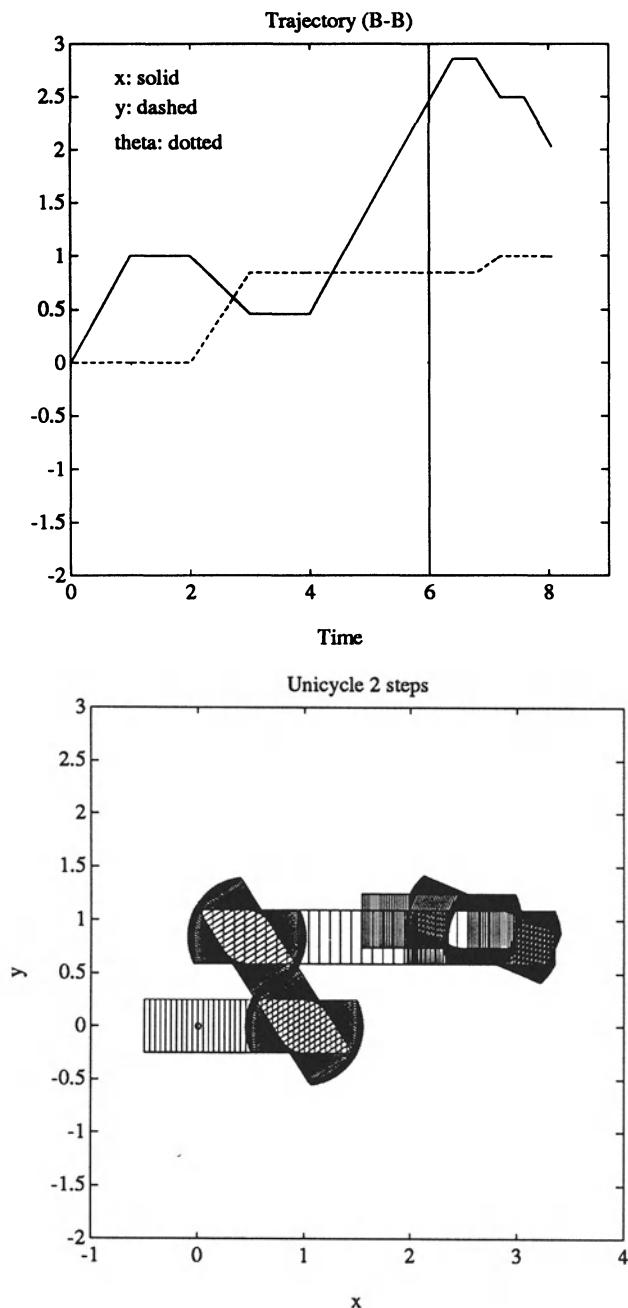


Figure 3: Trajectory generated after 2 iterations with piecewise constant controls.

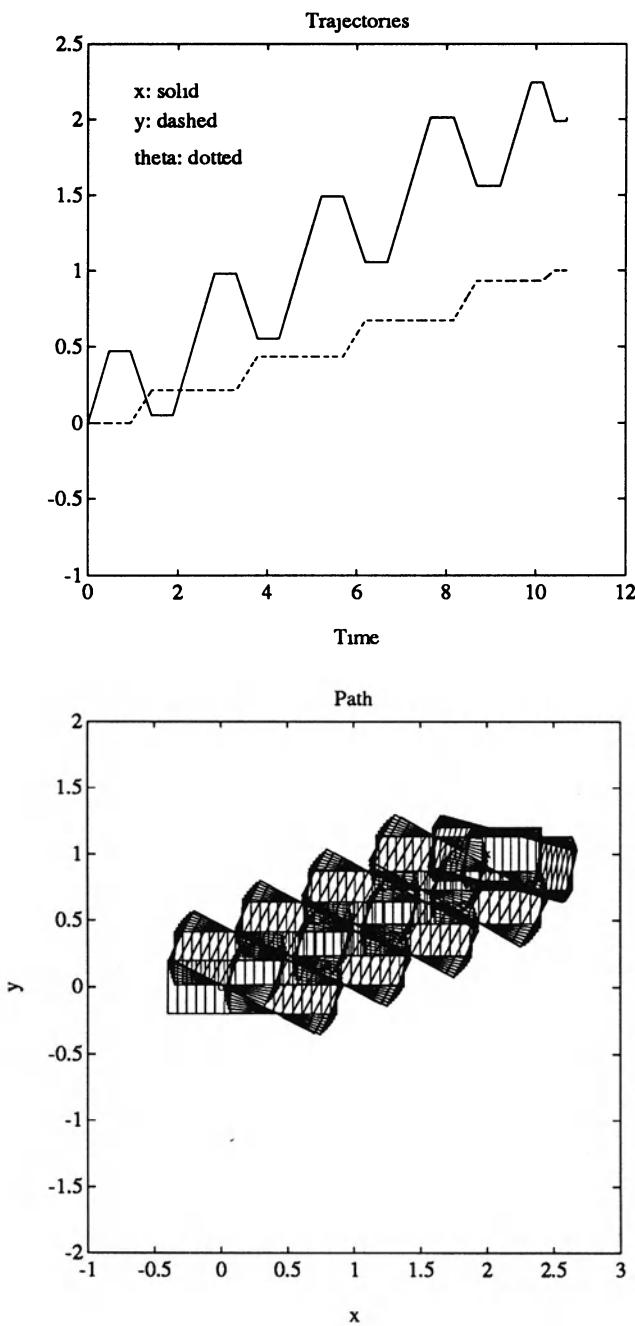


Figure 4: Iterated algorithm with a small critical distance.

to obtain the controls w_i . Then using the feedback (14) compute the controls for the original system. The results of such a simulation (with the same x_0, x_f as before) are presented in Figure 5.

A second example is given by a front wheel drive cart (Figure 6). The controls are the driving speed of the front wheels (u_1) and the turning speed of the front wheels (u_2). The system equations are

$$\begin{aligned}\dot{x}_1 &= \cos(x_3) \cos(x_4) u_1, \\ \dot{x}_2 &= \cos(x_3) \sin(x_4) u_1, \\ \dot{x}_3 &= u_2, \\ \dot{x}_4 &= \frac{1}{l} \sin(x_3) u_1.\end{aligned}\tag{16}$$

where x_1, x_2 are the Cartesian coordinates of the cart, x_3 is the steering angle and x_4 is the angle the main axis of the cart makes with the x_1 -axis. Let's rewrite the system as $\dot{x} = u_1 f_1(x) + u_2 f_2(x)$. Then the vectors $f_1(x), f_2(x), [f_1, f_2](x), [f_1, [f_1, f_2]](x)$ span \mathbb{R}^4 in a neighborhood of $x = 0$. Again we can see that the system is not nilpotent. In fact, $\text{ad}_{f_2}^{2n} f_1 = (-1)^n f_1$ for $n \geq 0$. After three iterations an acceptable error (≤ 0.01) is achieved (see Figure 7). (In this plot we rename the variables as $x = x_1$, $y = x_2$, $\psi = x_3$ and $\theta = x_4$.)

The cart is driven from $x_0 = (0, 0, 0, 0)$ to $x_f = (0, -1, 0, 0)$. The computation of the controls uses the formulas explained in Examples 1 and 2. After some cancellations Examples 2 and 3. After some cancellations and regroupings the resulting controls are piecewise constant ($|u_i| \leq 1$) with 19 switches for each step.

This system is also nilpotentizable and therefore a one step procedure can be used to reach the target exactly. The following feedback makes the system nilpotent.

$$\begin{aligned}u_1 &= \frac{1}{\cos(x_3) \cos(x_4)} w_1, \\ u_2 &= \cos^3(x_4) \cos^2(x_3) w_2 - \frac{3 \sin(x_4) \sin^2(x_3)}{l \cos^2(x_4)} w_1.\end{aligned}\tag{17}$$

In fact, if we make the change of coordinates

$$z_1 = x_1 \quad z_2 = \frac{\tan(x_3)}{\cos^3(x_4)} \quad z_3 = \tan(x_4) \quad z_4 = x_2,\tag{18}$$

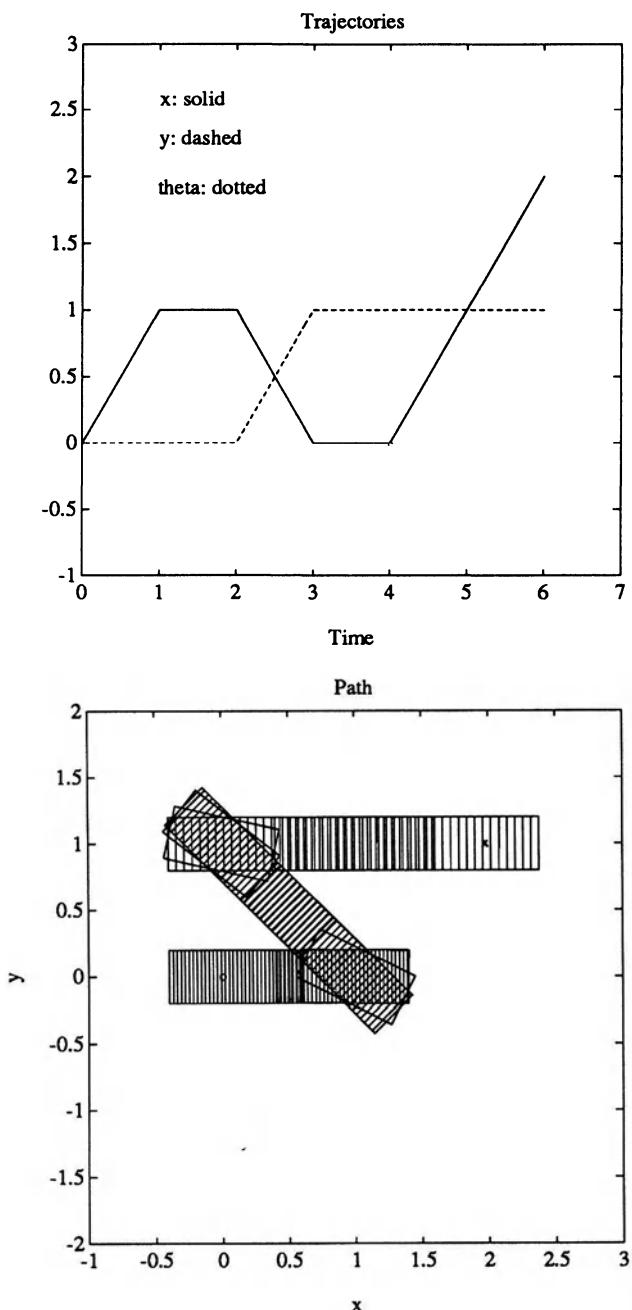


Figure 5: Unicycle with precompensator.

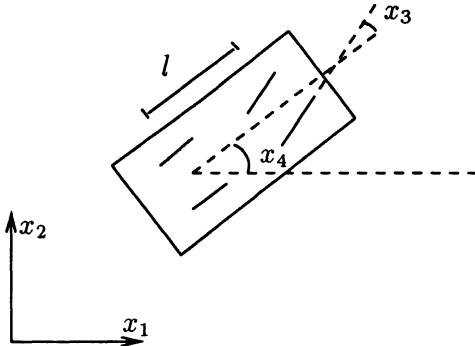


Figure 6: Front wheel drive cart.

the system becomes,

$$\begin{aligned}\dot{z}_1 &= w_1 \\ \dot{z}_2 &= w_2 , \\ \dot{z}_3 &= \frac{1}{l} z_2 w_1 , \\ \dot{z}_4 &= z_3 w_1 ,\end{aligned}\tag{19}$$

which is clearly nilpotent of order 3. Results of the simulation using a precompensator are given in Figure 8.

After the given change of coordinates the end points become $x_0 = (0, 0, 0, 0)$ and $x_f = (0, 0, 0, -1)$. If f_1, f_2 are the vector fields for the nilpotent system, then the desired motion is exactly in the direction of $[f_1, [f_1, f_2]]$. The backward P. Hall coordinates are in fact $(0, -1, 0, 0, 0)$. In Figure 9 we show plots of each variable against x . Notice that the controls generate periodic trajectories for $\theta = x_4$ and $\psi = x_3$. While θ goes around once, ψ traverses two loops. This is a direct consequence of the identity $e^{[X, [X, Y]]} = e^X e^{[X, Y]} e^{-X} e^{-[X, Y]}$, which holds in $L_3(X, Y)$, and the fact that the vectors $f_1, f_2, [f_1, f_2], [f_1, [f_1, f_2]]$ correspond to the coordinate axes x_1, x_3, x_4, x_2 respectively. In general, the trajectories prescribed by the forward P. Hall coordinates (which in this example are $(0, 0, 0, -1, 0)$) are made up of moves in the directions of each

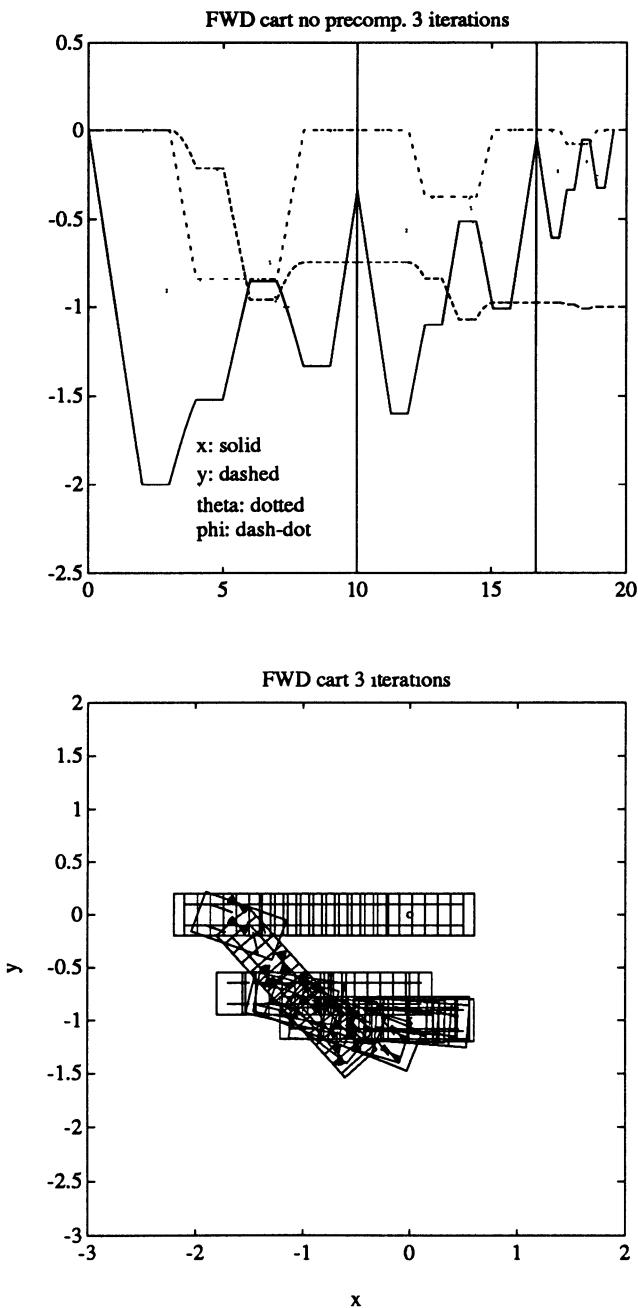


Figure 7: Iterated algorithm applied to front wheel drive cart

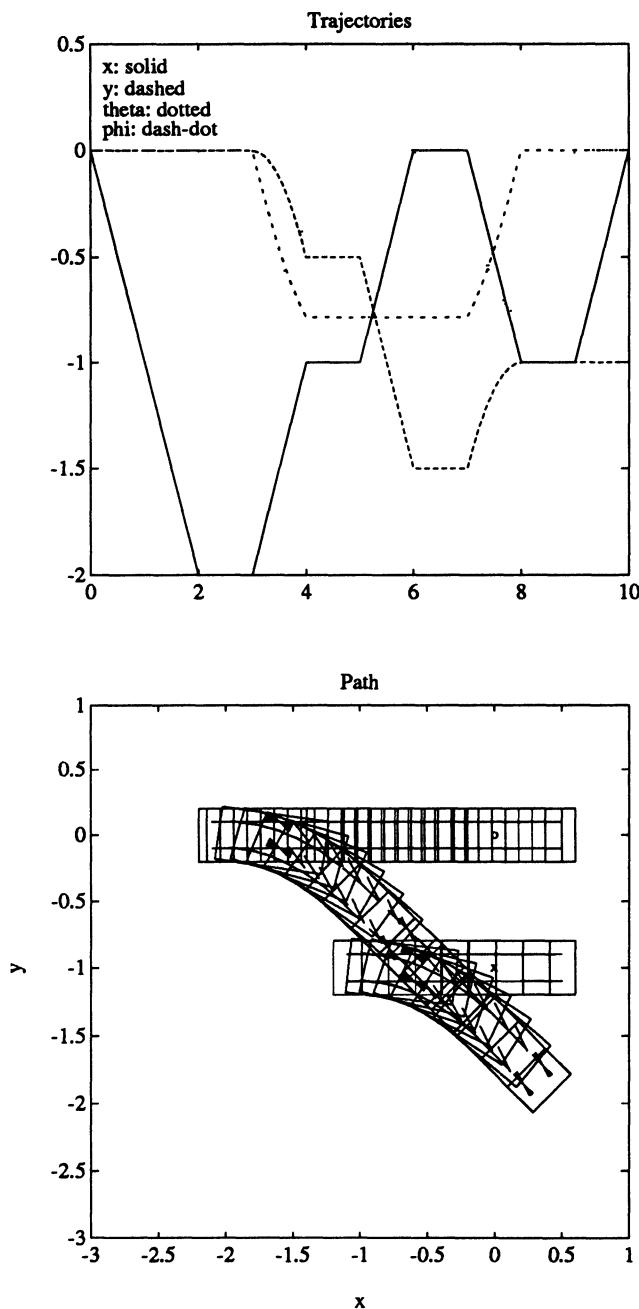


Figure 8: Front wheel drive cart with precompensator.

bracket in the P. Hall basis. The move corresponding to a given bracket results in periodic motions of the variables corresponding to lower order brackets.

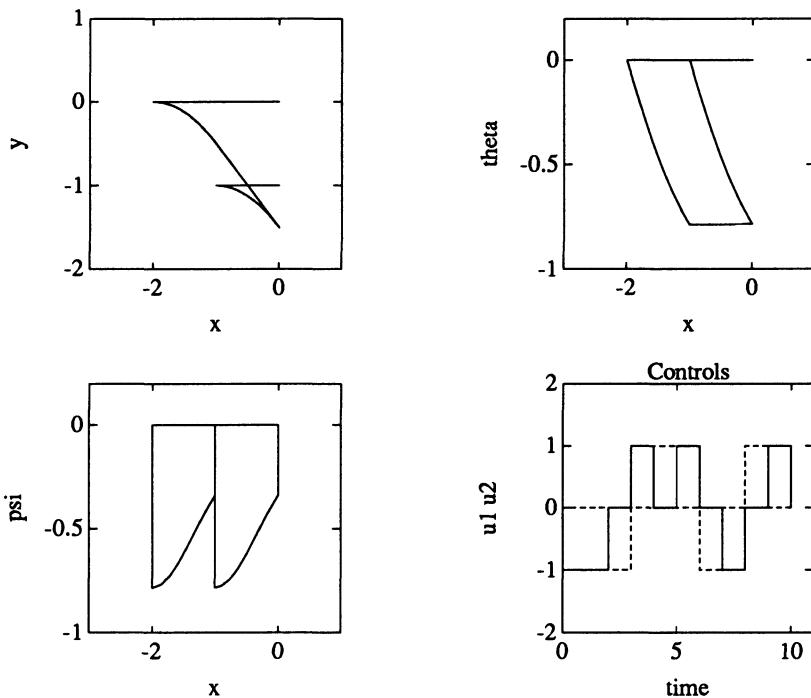


Figure 9: Plots of the different variables against x for the front wheel drive cart with a precompensator

Finally, we present an example of a nilpotentizable system of higher order. Consider a front wheel drive cart as before which in addition is pulling a trailer (Figure 10). The system has one more equation than before to account for the variable representing the angle the trailer makes with the x_1 -axis.

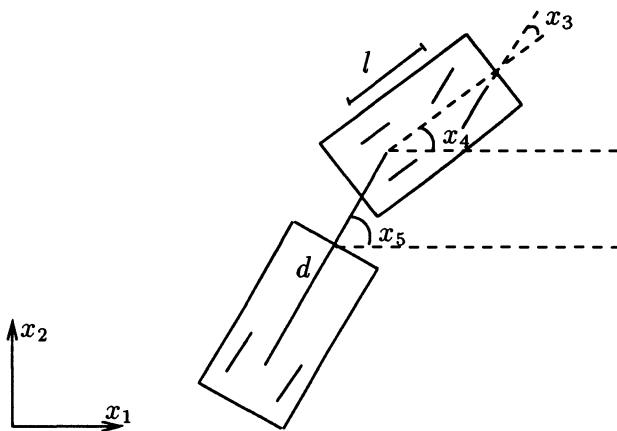


Figure 10: Front wheel drive cart with trailer.

The equations are,

$$\begin{aligned}\dot{x}_1 &= \cos(x_3) \cos(x_4) u_1 \\ \dot{x}_2 &= \cos(x_3) \sin(x_4) u_1 \\ \dot{x}_3 &= u_2, \\ \dot{x}_4 &= \frac{1}{l} \sin(x_3) u_1, \\ \dot{x}_5 &= \frac{1}{d} \sin(x_4 - x_5) \cos(x_3) u_1.\end{aligned}\tag{20}$$

The span of the five vectors $f_1(x)$, $f_2(x)$, $[f_1, f_2](x)$, $[f_1, [f_1, f_2]](x)$ and $[f_1, [f_1, [f_1, f_2]]](x)$ is \mathbb{R}^5 for x near zero. The system is nilpotentizable. In fact, it is feedback equivalent to a nilpotent system of order 4. The calculations are similar to the front wheel drive cart example but have to be iterated twice (complete details are available in [LS90]). After feedback transformations and change of

coordinates the system becomes

$$\begin{aligned}\dot{x}_1 &= u_1, \\ \dot{x}_2 &= u_2, \\ \dot{x}_3 &= x_2 u_1, \\ \dot{x}_4 &= x_3 u_1, \\ \dot{x}_5 &= \left(x_3 \left(\sqrt{1 + x_4^2} \right)^{-1} + x_4 \right) u_1.\end{aligned}\tag{21}$$

Outline of the proofs

The quickest way to prove Theorem 1 is to use standard facts about Lie groups, together with a general result proved by R. Palais (cf. Palais [Pal57]):

Theorem 4 *Let L be a finite-dimensional Lie algebra of vector fields on a smooth manifold M , and assume that f_1, \dots, f_m are generators of L . Assume that the f_i are complete. Let G be the connected simply connected Lie group with Lie algebra L . Then there exists a unique right action of G on M whose differential is the identity map.*

We recall that a *right action* of a Lie group G on a smooth manifold M is a smooth map $M \times G \ni (x, g) \rightarrow xg \in M$ such that $x(gh) = (xg)h$ and $xe = x$ for all $x \in M$, $g, h \in G$, where e is the identity element of G . The *differential* of the action is the map that assigns to each member λ of the Lie algebra L of G the vector field v_λ defined by

$$v_\lambda(x) = \frac{d}{dt} \Big|_{t=0} x \exp(t\lambda),\tag{22}$$

where “exp” is the exponential map from L to G . In the special case when L is already a Lie algebra of vector fields, it makes sense to talk about the differential being the identity map.

We also recall that it is part of the content of Palais’ theorem that, under the stated assumptions, it follows that all the members of L are complete, even though it is not true in general that the Lie

algebra generated by a set of complete vector fields consists entirely of complete vector fields.

We can define a map $\nu : L(X_1, \dots, X_m) \rightarrow L(\mathbf{f})$ by just plugging in the f_i for the X_i . Because $L(\mathbf{f})$ is nilpotent of order k , this map is a Lie algebra homomorphism. So the map extends to a Lie group homomorphism (also denoted by ν) from $G_k(X_1, \dots, X_m)$ to G , where G is the connected simply connected Lie group with Lie algebra $L(\mathbf{f})$. Using Theorem 4, we get an action of G on M . If we are given a control $t \rightarrow u(t) = (u_1(t), \dots, u_m(t))$, $a \leq t \leq b$, and want to find the corresponding trajectory $t \rightarrow x(t)$ starting at $x(a) = \bar{x}$, we can proceed formally by solving the equation $\dot{S}(t) = S(t)(u_1(t)X_1 + \dots + u_m(t)X_m)$ on $G_k(X_1, \dots, X_m)$ with initial condition $S(a) = 1$ and then “projecting down” to M . Precisely, we can define $x(t) = \bar{x} \nu(S(t))$. If $Y \in L(X_1, \dots, X_m)$ is arbitrary, then $\nu(S(t)e^{sY}) = \nu(S(t))\nu(e^{sY}) = \nu(S(t))e^{s\nu(Y)}$, because ν is a homomorphism. Then

$$\frac{d}{ds}\Big|_{s=0} \nu(S(t)e^{sY}) = \frac{d}{ds}\Big|_{s=0} \nu(\Gamma(s))$$

if Γ is any curve in $G_k(X_1, \dots, X_m)$ such that $\Gamma(0) = S(t)$ and $\dot{\Gamma}(0) = S(t)Y$. In particular, we can take $\Gamma(s) = S(t+s)$, in which case $Y = \sum_i u_i(t)X_i$, and we get

$$\frac{d}{d\tau}\Big|_{\tau=t} \left(\bar{x} \nu(S(\tau)) \right) = \nu \left(\sum_i u_i(t)X_i \right) \left(\bar{x} \nu(S(t)) \right),$$

i.e. $\dot{x}(t) = \sum_i u_i(t)f_i(x(t))$. So $x(\cdot)$ is a trajectory, as stated.

It is clear that a similar conclusion holds for trajectories of the extended system. Using this, the assertion of Theorem 1 follows immediately: if u and v give rise to the same $S(T)$, then for any given initial condition \bar{x} the trajectories corresponding to u and v will go through $\bar{x} \nu(S(T))$ at time T , i.e. both u and v steer \bar{x} to the same point. ■

In order to prove Theorem 2, it is important to use a convenient formalism. It turns out that it is best to use the formalism in which vector fields and flows act on the right, as suggested by Agrachev,

Gamkrelidze and Sarychev in their papers on the “chronological calculus” (cf., e.g. [AG78]).

Also, since the possibility of explosions introduces extra complications, we prefer to do most of the work with vector fields in $\mathcal{D}^n(\mathbb{R}^n)$, the space of infinitely differentiable vector fields on \mathbb{R}^n with compact support, and then reduce the general case to this one by means of a cutoff argument.

The formalism is as follows. Points of \mathbb{R}^n , as well as many other objects such as tangent vectors at a point, can be regarded as Schwartz distributions, i.e. as linear functionals on the space $\mathcal{D}(\mathbb{R}^n)$ of compactly supported real-valued functions of class C^∞ on \mathbb{R}^n . Let $\text{Diff}(\mathbb{R}^n)$ be the set of all globally defined surjective diffeomorphisms $\Phi : \mathbb{R}^n \rightarrow \mathbb{R}^n$.

For $\varphi \in \mathcal{D}(\mathbb{R}^n)$, $f \in \mathcal{D}^n(\mathbb{R}^n)$, or $\Phi \in \text{Diff}(\mathbb{R}^n)$, let us use $x\varphi$, xf , $x\Phi$ as alternative notations for $\varphi(x)$, $f(x)$, $\Phi(x)$. If v is a tangent vector at x , and $\varphi \in \mathcal{D}(\mathbb{R}^n)$, then $v\varphi$ denotes what is usually referred to as the “directional derivative of φ at x in the direction of v ,” i.e. $\langle v, \nabla \varphi(x) \rangle$. If $v = \lim_{h \rightarrow 0} \frac{x(h)-x}{h}$, where $x(\cdot)$ is a C^1 curve such that $x(0) = x$, then $v\varphi$ is just equal to $\lim_{h \rightarrow 0} \frac{x(h)\varphi - x\varphi}{h}$. Every $\Phi \in \text{Diff}(\mathbb{R}^n)$ acts on $\mathcal{D}(\mathbb{R}^n)$ by an obvious duality: we let $\Phi\varphi(x) = \varphi(\Phi(x))$ i.e., in our notation, $x(\Phi\varphi) = (x\Phi)\varphi$. (So from now on we can just write $x\Phi\varphi$.) On the other hand, this same duality now enables us to have Φ act on the right on tangent vectors: if v is a tangent vector at x , then $v\Phi$ is the tangent vector at $x\Phi$ defined as the map $\varphi \rightarrow v(\Phi\varphi)$, so that $(v\Phi)\varphi = v(\Phi\varphi)$ and, once again, we can drop the parenthesis and write $v\Phi\varphi$. Notice that $v\Phi$ is the vector that in Differential Geometry is usually denoted Φ_*v , or $d\Phi(v)$, i.e. the image of v under the differential of Φ . If we write maps $\mu : \mathbb{R}^n \rightarrow \mathbb{R}^n$ (such as members of $\text{Diff}(\mathbb{R}^n)$ or of $\mathcal{D}^n(\mathbb{R}^n)$) as columns of real-valued functions, and use $D\mu$ to denote the Jacobian matrix of μ (i.e. the square matrix of functions whose rows are the gradients of the components of f), then $v\Phi$ is just $D\Phi(x) \cdot v$.

The space $\mathcal{D}^n(\mathbb{R}^n)$ is a Lie algebra. Moreover, for any $f \in \mathcal{D}^n(\mathbb{R}^n)$ we can define the norms

$$\|f\|_0 = \sup\{\|f(x)\| : x \in \mathbb{R}^n\}, \quad (23)$$

$$\|f\|_1 = \sup\{\|Df(x)\| : x \in \mathbb{R}^n\}. \quad (24)$$

For $f \in \mathcal{D}^n(\mathbb{R}^n)$, integral trajectories are defined for all times. We let $t \rightarrow xe^{tf}$ denote the integral curve of f that goes through x at time $t = 0$. It is clear that the map $(t, x) \rightarrow xe^{tf}$ is of class C^∞ , and satisfies $xe^{(t+s)f} = xe^{tf}e^{sf}$ for all t, s , and

$$\cdot \frac{d}{dt}xe^{tf} = xe^{tf}f. \quad (25)$$

The maps e^{tf} belong to $\text{Diff}(\mathbb{R}^n)$. In particular, $xe^{tf}ge^{-tf}$ is a well defined tangent vector at x whenever $g \in \mathcal{D}^n(\mathbb{R}^n)$, so $e^{tf}ge^{-tf}$ is a well defined vector field, which is easily seen to be in $\mathcal{D}^n(\mathbb{R}^n)$. This vector field will be denoted $e^{t\text{adj}_f}(g)$. If we fix x, f, g, t , and let $v(t) = xe^{t\text{adj}_f}(g)$, then $v(t) = w(0)$, where $s \rightarrow w(s)$ is the solution of the variational equation

$$\dot{w}(s) = Df(xe^{sf}) \cdot w(s), \quad (26)$$

with terminal condition $w(t) = g(xe^{tf})$. This implies in particular that the bound $\|w(t)\| \leq e^{|t| \|f\|_1} \|g(xe^{tf})\|$ holds. So

$$\|e^{t\text{adj}_f}(g)\|_0 \leq e^{|t| \|f\|_1} \|g\|_0. \quad (27)$$

We also need an estimate on the Lipschitz norm of e^{tf} . Define

$$\|\Phi\|_{Lip} = \sup \left\{ \frac{\|x\Phi - y\Phi\|}{\|x - y\|} : x, y \in \mathbb{R}^n, x \neq y \right\}. \quad (28)$$

Then Gronwall's inequality implies that

$$\|e^{tf}\|_{Lip} \leq e^{|t| \|f\|_1}. \quad (29)$$

Vector fields act as differential operators on functions: if $f \in \mathcal{D}^n(\mathbb{R}^n)$ and $\varphi \in \mathcal{D}(\mathbb{R}^n)$, then $f\varphi$ is the function $x \rightarrow (xf)\varphi$. (Recall that xf is a tangent vector, and tangent vectors act on functions.) Once again, the definitions are set up so that $x(f\varphi) = (xf)\varphi$,

so we can simply write $xf\varphi$. The function $f\varphi$ is often denoted $L_f\varphi$ in the control literature, and referred to as the “Lie derivative of φ in the direction of f ,” but the notation used here is both more convenient and consistent with the one commonly used in Differential Geometry, where vector fields are *defined* as differential operators on functions.

If we let $\text{DO}(\mathbb{R}^n)$ be the algebra of smooth linear differential operators on \mathbb{R}^n (so that every member of $\text{DO}(\mathbb{R}^n)$ acts as a linear operator from $\mathcal{D}(\mathbb{R}^n)$ to $\mathcal{D}(\mathbb{R}^n)$), then products of vector fields are well defined members of $\text{DO}(\mathbb{R}^n)$.

It is then clear that

$$\begin{aligned}\frac{d}{dt}e^{t\text{ad}_f}(g) &= e^{tf}(fg - gf)e^{-tf} \\ &= e^{t\text{ad}_f}([f, g]).\end{aligned}\quad (30)$$

From this one gets by successive integrations by parts

$$e^{t\text{ad}_f}(g) = \sum_{j=0}^k \frac{t^j}{j!} \text{ad}_f^j(g) + \int_0^t \frac{(t-s)^k}{k!} e^{s\text{ad}_f} \left(\text{ad}_f^{k+1}(g) \right) ds. \quad (31)$$

We now assume that we are given vector fields $g_1, \dots, g_m \in \mathcal{D}^n(\mathbb{R}^n)$, and an integer $k > 0$. We let $B_1 = X_1, \dots, B_m = X_m$, and then choose B_{m+1}, \dots, B_s so that B_1, \dots, B_s is a P. Hall basis of $L_k(X_1, \dots, X_m)$. We let $g_j = \mathbf{E}_g(B_j)$ for $j = m+1, \dots, s$, where \mathbf{E}_g is the “plugging in the g ’s” evaluation map, i.e. the map $L_k(X_1, \dots, X_m) \rightarrow \mathcal{D}^n(\mathbb{R}^n)$ defined by plugging in g_1, \dots, g_m for X_1, \dots, X_m . (Notice that this map is *not* in general a Lie algebra homomorphism. For instance, if $k = 1$, then $[X_1, X_2] = 0$ in $L_k(X_1, \dots, X_m)$, but $[g_1, g_2]$ need not vanish.)

We now consider a curve $t \rightarrow x(t)$, $0 \leq t \leq 1$, which is a solution of a differential equation

$$\dot{x}(t) = w_\nu(t)g_\nu(x(t)) + \dots + w_s(t)g_s(x(t)) + h_t(x(t)), \quad (32)$$

where we assume that $1 \leq \nu \leq s$, each control $t \rightarrow w_i(t)$ is bounded and measurable, and $t \rightarrow h_t$ is a measurable bounded $\mathcal{D}^n(\mathbb{R}^n)$ -valued function. (I.e. $h_t(x)$ is jointly measurable and bounded as a

function of t and x , and $h_t \in \mathcal{D}^n(\mathbb{R}^n)$ for each t .) We rewrite this in our notation as

$$\dot{x}(t) = x(t)(w_\nu(t)g_\nu + \dots + w_s(t)g_s + h_t) . \quad (33)$$

We let $W_\nu(t) = \int_0^t w_\nu(\tau)d\tau$, and write

$$y(t) = x(t)e^{-W_\nu(t)g_\nu} . \quad (34)$$

An easy calculation then shows that $y(t)$ satisfies

$$\dot{y}(t) = y(t)e^{W_\nu(t)\text{ad}_{g_\nu}} \left(w_{\nu+1}(t)g_{\nu+1} + \dots + w_s(t)g_s + h_t \right) (x(t)) . \quad (35)$$

We then write each vector field $e^{W_\nu(t)\text{ad}_{g_\nu}}(g_\mu)$, $\nu + 1 \leq \mu \leq s$, as

$$\begin{aligned} e^{W_\nu(t)\text{ad}_{g_\nu}}(g_\mu) &= \sum_{j=0}^{\kappa_\mu} \frac{W_\nu(t)^j}{j!} \text{ad}_{g_\nu}^j(g_\mu) \\ &\quad + \int_0^{W_\nu(t)} \frac{(W_\nu(t)-\omega)^{\kappa_\mu}}{\kappa_\mu!} e^{\omega \text{ad}_{g_\nu}} \left(\text{ad}_{g_\nu}^{\kappa_\mu+1}(g_\mu) \right) d\omega , \end{aligned} \quad (36)$$

using (31). The κ_μ are chosen so that $\text{ad}_{g_\nu}^{\kappa_\mu}(g_\mu)$ has degree $\leq k$ but $\text{ad}_{g_\nu}^{\kappa_\mu+1}(g_\mu)$ has degree $> k$. The brackets $\text{ad}_{g_\nu}^j(g_\mu)$, $j \leq \kappa_\mu$, can be written as linear combinations $\sum_\alpha c_{j\mu\alpha} g_\alpha$ of g_1, \dots, g_s , with constant coefficients. (This follows because it is true on the purely formal level, since the B_i form a basis of $L_k(X_1, \dots, X_m)$, and therefore the $\text{ad}_{X_\nu}^j(X_\mu)$ are linear combinations of them.)

At this point it is crucial to observe that in these expressions all the $c_{j\mu\alpha}$ vanish for $\alpha \leq \nu$. To see this observe that the elements of the P. Hall basis are *homogeneous*. Therefore in the expression of a homogeneous bracket B as a linear combination of the B_i only B_i 's of the same degree as B will occur. The brackets $\text{ad}_{g_\nu}^j(g_\mu)$ with $j > 0$ all have degree strictly larger than the degree of g_ν , and therefore also strictly larger than the degrees of the g_η with $\eta < \nu$. (Because in the P. Hall basis the ordering is compatible with degree, i.e. $\text{degree}(B_i) \leq \text{degree}(B_{i'})$ if $i < i'$.) So in those brackets g_1, \dots, g_ν do not occur. As for the $\text{ad}_{g_\nu}^j(g_\mu)$ with $j = 0$, these are,

of course, the g_μ themselves, for $\mu > \nu$, so again their expressions do not involve g_1, \dots, g_ν . Therefore y satisfies

$$\dot{y}(t) = y(t) \left(\tilde{w}_{\nu+1}(t) g_{\nu+1} + \dots + \tilde{w}_s(t) g_s + \tilde{h}_t \right), \quad (37)$$

with obvious formulae for the \tilde{w}_j and \tilde{h} :

$$\tilde{w}_\mu = \sum_{\rho, j} \frac{W_\nu^j}{j!} w_\rho c_{j\rho\mu}, \quad (38)$$

$$\begin{aligned} \tilde{h}_t = & e^{W_\nu(t) \operatorname{ad}_{g_\nu}}(h_t) \\ & + \sum_\rho w_\rho(t) \int_0^{W_\nu(t)} \frac{(W_\nu(t) - \omega)^{\kappa_\rho}}{\kappa_\rho!} e^{\omega \operatorname{ad}_{g_\nu}} \left(\operatorname{ad}_{g_\nu}^{\kappa_\rho+1}(g_\rho) \right) d\omega. \end{aligned} \quad (39)$$

We want to use these formulas to estimate \tilde{w} . It turns out that the best form of the estimates is in terms of the *dilation norm*. Let δ_i be the degree of B_i . Define $| \dots | : \mathbb{R}^s \rightarrow \mathbb{R}^+$ by

$$|(w_1, \dots, w_s)| = \sum_{i=1}^s |w_i|^{\frac{1}{\delta_i}}. \quad (40)$$

Then $| \dots |$ is called the “dilation norm” (even though it is not a norm in the usual sense). For a control $t \rightarrow w(t) \in \mathbb{R}^s$ we define the dilation norm $|w(\cdot)| = \sup\{|w(t)| : 0 \leq t \leq 1\}$. It then follows from (38) that $|\tilde{w}(\cdot)| \leq K|w(\cdot)|$, where K is a purely combinatorial constant, depending only on m, k , and the choice of a P. Hall basis.

We also estimate \tilde{h} . Using (39) and (27) we conclude that

$$\|\tilde{h}_t\|_0 \leq e^{|w(\cdot)|\delta_\nu} \|g_\nu\| \left(\|h_t\|_0 + \hat{K}|w|^{(\kappa_\rho+1)\delta_\nu + \delta_\rho} \right), \quad (41)$$

where \hat{K} is a constant that depends only on the $\| \dots \|_0$ norms of the “bad brackets” (i.e. the brackets $\operatorname{ad}_{g_\nu}^{j+1}(g_\rho)$ with $j\delta_\nu + \delta_\rho \leq k < (j+1)\delta_\nu + \delta_\rho$).

Now let $g_1, \dots, g_m \in \mathcal{D}^n(\mathbb{R}^n)$, k , and the brackets g_{m+1}, \dots, g_s be as above, and such that in addition $g_1(x), \dots, g_s(x)$ span \mathbb{R}^n for all x such that $\|x\| \leq R + 1$. It then follows in particular

that, if $v \in \mathbb{R}^n$ is any vector, and $\|x\| \leq R + 1$, then we can write $v = \sum_i v_i g_i(x)$ with the v_i such that $|v_i| \leq C\|v\|$ for some fixed constant C . If we now pick two points p, q in \mathbb{R}^n , and take $\gamma : [0, 1] \rightarrow \mathbb{R}^n$ to be defined by $\gamma(t) = p + t(q - p)$, we can then define the functions $t \rightarrow v_i(t)$ as before, and get the bound $|v_i(t)| \leq C\|\dot{\gamma}(t)\|$, i.e. $|v_i(t)| \leq C\Delta$, where $\Delta = \|q - p\|$. In particular, this implies that the dilation norm of $v(\cdot)$ is bounded by a fixed constant times $\Delta^{\frac{1}{k}}$. (From now on we only consider points p, q such that $\Delta \leq 1$, so that $\Delta \leq \Delta^\sigma$ if $0 < \sigma < 1$.)

We then compute u from v as explained in §9. We now have $T = 1$. We observe that the P. Hall coordinates of $S(T)$ are bounded by a fixed constant times Δ , and therefore the u computed by our method is bounded by a constant times $\Delta^{\frac{1}{k}}$. Since u only has components associated to the B_i of degree 1, this implies that the dilation norm of u is also bounded by a constant times $\Delta^{\frac{1}{k}}$.

We now consider the trajectories $t \rightarrow \hat{x}(t)$, $t \rightarrow \check{x}(t)$ of $\dot{x} = x(w_1 g_1 + \dots + w_s g_s)$ with initial condition p and for the two choices $w = v$ and $w = u$. In both cases, we perform repeatedly the transformation described above, obtaining trajectories $t \rightarrow \hat{x}_j(t)$, $t \rightarrow \check{x}_j(t)$ of $\dot{x} = x(v_j^i g_j + \dots + v_s^i g_s + \hat{h}_t^i)$ and $\dot{x} = x(u_j^i g_j + \dots + u_s^i g_s + \check{h}_t^i)$, respectively. (Naturally, $\hat{x}_1 \equiv \hat{x}$, $\check{x}_1 \equiv \check{x}$, $\hat{h}_t = 0$, $\check{h}_t = 0$, $v^1 = v$, $u^1 = u$.) After s applications of these transformations, we end up with trajectories \hat{x}_{s+1} , \check{x}_{s+1} of the systems $\dot{x} = x\hat{h}_t^{s+1}$, $\dot{x} = x\check{h}_t^{s+1}$. By repeated applications of our estimates we get $\|\hat{h}_t\|_0 \leq F(\Delta)\Delta^{1+\frac{1}{k}}$, where F is a function that remains bounded as $\Delta \rightarrow 0$, and a similar bound for $\|\check{h}_t\|_0$. From this it follows that $\|\hat{x}_{s+1} - p\|$ and $\|\check{x}_{s+1} - p\|$ are bounded by $F(\Delta)\Delta^{1+\frac{1}{k}}$, and so $\|\hat{x}_{s+1} - \check{x}_{s+1}\| \leq 2F(\Delta)\Delta^{1+\frac{1}{k}}$.

Now, while we apply the transformations to the \hat{x}_j , \check{x}_j , we can simultaneously apply them as well to the solutions \hat{S}_j , \check{S}_j of the corresponding formal equations, beginning with $\hat{S}(t)$ and $\check{S}(t)$, the solutions of the formal extended equation $\dot{S} = S(v_1 X_1 + \dots + v_s X_s)$, and of the formal equation $\dot{S} = S(u_1 X_1 + \dots + u_m X_m)$, respectively. It is easily seen that the formulas that give the new v 's and u 's in terms of the old ones are exactly the same for the formal systems as for the original differential equation. However, in the formal case,

after we get through all the transformations, the right-hand sides of both equations are just zero, because the brackets of degree $> k$ vanish.

So we end up with the following equations:

$$\begin{aligned}\hat{x}(1) &= \hat{x}_{s+1}(1)e^{\alpha_s g_s} \dots e^{\alpha_1 g_1}, \\ \check{x}(1) &= \check{x}_{s+1}(1)e^{\beta_s g_s} \dots e^{\beta_1 g_1}, \\ \hat{S}(1) &= e^{\bar{\alpha}_s X_s} \dots e^{\bar{\alpha}_1 X_1}, \\ \check{S}(1) &= e^{\bar{\beta}_s X_s} \dots e^{\bar{\beta}_1 X_1}.\end{aligned}\tag{42}$$

Because of the parallelism pointed out above between the transformations for the formal and non-formal systems, we have $\alpha_i = \bar{\alpha}_i$ and $\beta_i = \bar{\beta}_i$ for all i . On the other hand, u is constructed so that $\hat{S}(1) = \check{S}(1)$, so that (by the uniqueness of the P. Hall coordinates) $\alpha_i = \beta_i$ as well. We then have $\hat{x}(1) - \check{x}(1) = \hat{x}_{s+1}(1)\Phi - \check{x}_{s+1}(1)\Phi$, where Φ is the map

$$z \rightarrow z e^{\alpha_s g_s} \dots e^{\alpha_1 g_1}.\tag{43}$$

In view of (29), the Lipschitz norm of Φ is bounded by e^{CK} , where K is a bound for the norms $\|g_j\|_1$ and C is a bound for the α_i . But then C can be taken to be bounded by a constant times Δ . So we have the bound

$$\begin{aligned}\|\hat{x}(1) - \check{x}(1)\| &\leq e^{K'\Delta} \|\hat{x}_{s+1}(1) - \check{x}_{s+1}(1)\| \\ &\leq F(\Delta) \Delta^{1+\frac{1}{k}}\end{aligned}\tag{44}$$

for some choice of F .

This completes the proof for the case of vector fields in $g_i \in \mathcal{D}^n(\mathbb{R}^n)$. If we now have vector fields f_i that are not necessarily of compact support, we proceed as follows. Pick a ball $\mathcal{B}_R = \{x : \|x\| \leq R\}$. Let $g_i = \psi f_i$, where ψ is a function in $\mathcal{D}(\mathbb{R}^n)$ which is equal to 1 for $\|x\| \leq R + 1$. Pick p and q in the ball \mathcal{B}_R . Then the f_i and the g_i agree on the segment from p to q , so the v_i are the same whether we compute them using the f_i or the g_i . The u_i are therefore the same as well, since the calculation of u from v is purely formal. In particular, this implies that the u_i are bounded

by a fixed constant times $\Delta^{\frac{1}{k}}$. Therefore, if Δ is sufficiently small, the trajectory of the original system that corresponds to u cannot leave the ball $\{x : \|x\| \leq R + 1\}$. But then this trajectory is also a trajectory of the g -system, and the bound proved above holds. ■

Conclusion

The method presented above for solving the MPP rests on a sound theoretical basis, and our preliminary simulation results appear to show that it performs rather well in practice. Naturally, the procedure we have outlined can still be improved in a number of ways, and this requires continued research on several important issues.

A crucial problem is that of estimating the critical distance. The proof of Theorem 2 only provides a very rough estimate of $\Delta_c(\mathcal{R})$. In the simulations, $\Delta_c(\mathcal{R})$ appears to be much larger.

Another important question is that of the class of controls u being used. The method described here involves the concatenation of controls that realize motion along the trajectories of the vector fields f_i . One can use bang-bang controls, but this is not necessary. For instance, the role of A in Example .3 can be played by a constant control $u_1 \equiv 1$, $u_2 \equiv 0$, during time 1, but we can also use instead a nonconstant control —e.g. $u_1(t) = \frac{1}{2}\sin t$, $0 \leq t \leq \pi$ — as long as its integral is equal to 1. In this way we end up with continuous controls. Alternatively, we can use more complicated pieces to build up our controls—e.g. controls that directly generate exponentials $e^{\alpha X + \beta Y}$. In this way, the total number of pieces can be cut down considerably.

A third important issue is that of fully exploiting the possibilities of feedback nilpotentization. This requires that one look for new classes of nilpotentizable systems, and also that one improve the existing nilpotentization results by making them as explicit as possible.

References

- [AG78] A.A. Agrachev and R.V. Gamkrelidze. The exponential representation of flows and chronological calculus. *Math. Sbornik*, 107(149):467–532, 1978.
- [Bat78] J.A. Bathurin. *Lectures on Lie algebras*. Akademie Verlag, Berlin, 1978.
- [Bro81] R.W. Brockett. *Control theory and singular Riemannian geometry*, pages 11–27. New Directions in Applied Mathematics. Springer-Verlag, New York, 1981.
- [Fli83] M. Fliess. Réalisation locale des systèmes non linéaires, algèbres de lie filtrées transitives et séries génératrices non commutatives. *Invent. Math.*, 71:521–537, 1983.
- [Her89] H. Hermes. Distributions and the lie algebras their bases can generate. In *Proc. Amer. Math. Soc. 106, No. 2*, pages 555–565, 1989.
- [HLS84] H. Hermes, A. Lundell, and D. Sullivan. Nilpotent bases for distributions and control systems. *J. Diff. Eqs.*, 55(3):385–400, 1984.
- [HSK89] J. Hauser, S. Sastry, and P. Kokotovic. Nonlinear control via approximate input-output linearization: the ball and beam example. In *Proc. 28th IEEE CDC*, pages 1987–1993, Tampa, Florida, Dec. 1989.
- [LS90] G. Lafferriere and H.J. Sussmann. Motion planning for controllable systems without drift: a preliminary report. SYCON report 90-04, Rutgers Center for Systems and Control, Rutgers University, New Brunswick, New Jersey, July 1990.
- [MS90a] R.M. Murray and Sastry S.S. Grasping and manipulation using multifingered robot hands. Memorandum

UCB/ERL M90/24, Electronics Research Laboratory, Univ. of California, Berkeley, Berkeley, California, March 1990.

- [MS90b] R.W. Murray and S.S. Sastry. Steering nonholonomic systems using sinusoids. In *Proc. 29th IEEE CDC*, pages 2097–2101, Honolulu, Hawaii, Dec. 1990.
- [Pal57] R. Palais. *Global formulation of the Lie theory of transformation groups*, volume 22 of *Mem. Amer. Math. Soc.* AMS, 1957.
- [SL] H.J. Sussmann and G. Lafferriere. Motion planning for controllable systems without drift. In preparation.
- [SL89] S. Sastry and Zexiang Li. Robot motion planning with nonholonomic constraints. In *Proc. 28th IEEE CDC*, pages 211–216, Tampa, Florida, Dec. 1989.
- [Sus83] H.J. Sussmann. Lie brackets and local controllability: a sufficient condition for scalar input systems. *S.I.A.M. J. Control and Optimization*, 21(5):686–713, 1983.
- [Sus86] H.J. Sussmann. *A product expansion for the Chen series*, pages 323–335. *Theory and Applications of Nonlinear Control Systems*, C. Byrnes and A. Lindquist Eds. North-Holland, 1986.
- [Sus87] H.J. Sussmann. A general theorem on local controllability. *S.I.A.M. J. Control and Optimization*, 25(1):158–194, 1987.

8

Planning Smooth Paths for Mobile Robots

Paul Jacobs* and John Canny†

University of California
Berkeley, Ca 94720

Phone: (510) 642-9955 FAX: (510) 642-5775
email: jfc@cs.berkeley.edu, pjacobs@qualcomm.com

Abstract

A mobile robot with a wheeled base is subject to non-holonomic constraints on its motion. We present a planner for finding obstacle-avoiding paths subject to the constraints. We define a set of canonical trajectories which satisfy the constraints. A configuration space can be constructed for these trajectories in which there is a simple characterization of the boundaries of the obstacles generated by the workspace obstacles.

We describe a graph search algorithm which divides the configuration space into sample trajectories. The characterization of the boundaries enables us to calculate an approximate path in time $O(\frac{n^3}{\delta} \log n + A \log(\frac{n}{\delta}))$, where n is the number of obstacle vertices, A is the number of free trajectories, and δ is a robustness measure. Using a plane sweep technique and quadtrees we can generate robust paths in time $O(n^4 \log n + \frac{n^2}{\delta^2})$.

*Research supported by the Defense Advanced Research Projects Agency (DoD), monitored by Space and Naval Warfare Systems Command under Contract N00039-88-C-0292.

†Supported by a David and Lucile Packard Foundation Fellowship and by NSF Presidential Young Investigator Grant IRI-8958577

1 Planning Smooth Paths for Mobile Robots

Research in path planning for robots has traditionally been focused on finding ways to ensure that a fixed arm will not collide with objects in its workspace as it performs its tasks. However, to be truly flexible, manipulators will someday be mounted upon mobile platforms, giving them not just the ability to manipulate objects in a fixed region, but to interact with their environment in more complex ways, much as humans do.

In this paper, we consider the problem of planning paths for a robot which has a minimum turning radius. This is a first step towards accurately modeling a robot with the kinematics of a car. The approach we have taken is to define a set of trajectories which satisfy the constraints. We show that it is sufficient to consider only these trajectories to find a path if one exists. We then exploit the geometry of this restricted class of trajectories to produce an efficient search algorithm.

1.1 Statement of the Problem

In this section, we consider a constraint on the average curvature of the unit speed path $X(t)$ ³ given by

$$\|\dot{X}(t_1) - \dot{X}(t_2)\| \leq R^{-1}|t_1 - t_2|, \quad (1.1)$$

where the condition holds for all t_1, t_2 for which the trajectory is defined. From the viewpoint taken in this paper, we have already translated the problem from a dynamic constraint to a geometric one. However, the motivation behind constraining the average curvature comes from the inability of a mobile robot to make sharp turns.

³The average curvature is used instead of the curvature $\|\ddot{X}(t)\|$ because, as we later show, the minimal length path with bounded average curvature is only C^1 . It is crucial for these definitions that the curve has unit speed, or $\|\dot{X}(t)\| = 1, \forall t$.

Here we give a precise formulation of the general problem of planning minimal-length (equivalently, minimal-time) unit speed paths with bounded average curvature.

Notation. We describe the path $X(t) \in \mathbb{R}^2$ in coordinates by

$$X(t) = \begin{pmatrix} x(t) \\ y(t) \end{pmatrix}.$$

We can compactly represent the velocity vector in this system by the angle, $\theta(t)$, it makes with the x-axis.

$$\theta(t) = \text{atan2}(\dot{y}(t), \dot{x}(t)),$$

where the function $\text{atan2}(y, x)$ gives the arctangent respecting the sign of the components.

Problem Statement

Let $\Omega \subset \mathbb{R}^2$ be a (closed) set of polygons in the plane. This set represents the obstacles in the environment. $\partial\Omega$ represents the boundary of the set of obstacles and $\text{int}(\Omega)$ denotes its interior. The variable R is the minimum turning radius of the mobile robot.⁴ We consider the class of paths $X(t) \in \mathbb{R}^2$ which satisfy the following constraints.⁵

1. $X(0) = \text{IP}$, $\theta(0) = \theta_0$.
2. $\exists t_f > 0$, such that $X(t_f) = FP$, $\theta(t_f) = \theta_f$.
3. $X(t) \in \mathbb{R}^2 \setminus \text{int}(\Omega)$, $\forall t \in [0, t_f]$.
4. $\|\dot{X}(t)\| = (x^2(t) + y^2(t))^{\frac{1}{2}} = 1$, $\forall t \in [0, t_f]$.
5. Given $R > 0$, $\|\dot{X}(t_1) - \dot{X}(t_2)\| \leq R^{-1}|t_1 - t_2|$, $\forall t_1, t_2 \in [0, t_f]$.

The problem is to find a path $X^*(t)$ defined for $t \in [0, t_f^*]$ such that t_f^* is minimal over all paths in this class. We should note that Condition 4 constrains the possible paths to have unit speed. Because of this, the parameter t denotes both elapsed time and distance traveled. Such paths are called **arc-length parametrized**, and the minimal-time and minimal-length paths coincide.

⁴The maximum average curvature of the path is therefore R^{-1} .

⁵This general formulation of the problem is due to Dubins [Dub57].

1.2 Statement of the Results

In this paper, we present an algorithm for planning paths for a mobile robot subject to the nonholonomic kinematic constraint imposed by the Pfaffian

$$dx \tan \theta - dy = 0 \quad (1.2)$$

and the dynamic constraint imposed by a bound on the change in θ .

Our technique is to define a set of canonical trajectories which satisfy the constraints. These trajectories are formed by concatenating subpaths that pass between points on the boundaries of the obstacles. The subpaths are useful because they have a very simple geometric structure.

We conceptually split the problem of planning a complete trajectory into two steps. The first part is to find all of the collision-free subpaths which pass between points on the obstacle boundaries. In order to solve this problem, we construct a configuration space representation *for the subpaths*. Thus, for each pairing of obstacle vertices and edges there is a configuration space defined. Then we map the workspace obstacles into configuration space obstacles. We show that there is a simple characterization of the boundaries of the obstacles in configuration space. In fact, there are only four basic types of constraints which can form these boundaries.

Then, to form the complete trajectory, we connect the subpaths together so that the orientations match smoothly at the points of the obstacle boundaries. Based on the simple characterization of the configuration space, we construct a graph search algorithm which divides the configuration space into sample trajectories. In this case, only slices of the configuration space obstacle are actually computed. The characterization of the boundaries enables us to calculate an approximate path in time $O(\frac{n^3}{\delta} \log n + A \log(\frac{n}{\delta}))$, where n is the number of obstacle vertices in the environment, δ describes both the robustness of the generated path and the closeness of the approximation and A is the number of free trajectories (at most $(\frac{n}{\delta})^2$). By robustness, we mean that the directions of travel may be

changed in a defined interval at certain points along the trajectory and a neighboring collision-free path will still exist.

Another method involves analyzing the form of the configuration space constraints and using this information in a plane sweep algorithm for computing the configuration space obstacles. The obstacles are represented by a quadtree which is used to generate robust paths in time $O(n^4 \log n + \frac{n^2}{\delta^2})$.

2 Preliminary Discussion

In this section, we develop the restricted class of trajectories over which we perform the search.

2.1 Well-Posedness

A theorem of Dubins asserts that there exists a minimal length path which satisfies the constraints of the problem statement when there are no workspace obstacles. We present a corollary which extends this result to allow obstacles. A consequence of this corollary is that the problem we have posed will have a solution. That is, if a path exists, then a minimal length path exists.

Theorem 2.3 ([Dub57]) *Consider the collection of curves which satisfy the conditions of the problem statement. If $\Omega = \emptyset$, then there exists an X^* of minimal length in this collection.*

Corollary 2.4 *If $\Omega \neq \emptyset$, and $\exists X$ satisfying the conditions of the problem statement with finite length, then $\exists X^*$ of minimal length which also satisfies the conditions.*

Proof. It is shown in the proof of Theorem 2.3 [Dub57] that there exists a sequence of curves satisfying the conditions which

converges uniformly to a curve of minimum length. The points of the curves in this sequence must all lie outside the open set, $\text{int}(\Omega)$, therefore the limit curve must also lie outside $\text{int}(\Omega)$. ■

Remark. The corollary asserts that the problem is well-posed. If a path exists which satisfies all the constraints, then there is such a path of minimal length.

2.2 The Canonical Trajectories

In this section we describe the form of the minimal length paths. First, we consider the case in which there are no obstacles ($\Omega = \emptyset$). This work was done by Dubins [Dub57]. Then we extend this result to allow for obstacles in the environment.

Theorem 2.5 ([Dub57]) *Every minimum length planar curve satisfying the conditions of the problem statement with $\Omega = \emptyset$ is necessarily a C^1 curve which is either*

1. *an arc of a circle of radius R , followed by a line segment, followed by an arc of a circle of radius R*
2. *a sequence of three arcs of circles of radius R*
3. *a subpath of a path of either of these two types*

Two examples of minimal length paths are depicted in Figure 1.

Definition 2.6 *A simple path is a path of the form given in Theorem 2.5 which has as its initial endpoint either IP or a point of $\partial\Omega$ and as its final endpoint FP or a point of $\partial\Omega$.*

Corollary 2.7 *Every minimum length planar curve satisfying the conditions of the problem statement is composed of path segments*

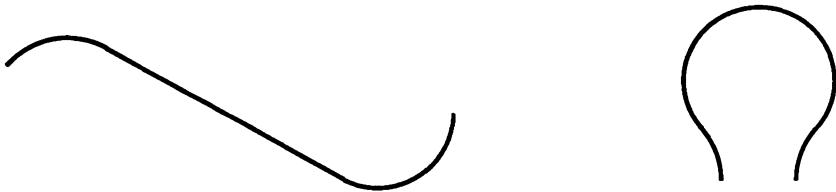


Figure 1: Minimal-Length Paths without Obstacles

which are simple paths. It must pass through $\partial\Omega$ finitely many times.⁶

Proof. If the minimal-length path from (IP, θ_i) to (FP, θ_f) does not touch $\partial\Omega$, we can ignore the obstacles and therefore by Theorem 2.5 it is a simple path. Otherwise, we claim that it touches $\partial\Omega$ finitely many times. In between the contacts with $\partial\Omega$, the curve must be a simple path, by reduction to the previous case. It can pass through $\partial\Omega$ only finitely many times because a path of minimum length is necessarily of finite length. For the two arc type simple paths, it is clear that for the path to leave an edge, return and satisfy the curvature constraints requires a finite distance to be traveled. It is shown in [Dub57] that for it to be of minimal length, the three arc type simple paths must have length greater than πR . X is necessarily a straight line for the intervals $[s_i, s_{i+1}]$ with $s_i \neq s_{i+1}$ and $X(s) \in \partial\Omega$, $\forall s \in [s_i, s_{i+1}]$. ■

Remark. This corollary establishes the form of the minimal length trajectories. They are obtained by concatenating paths of the type given in Theorem 2.5 which pass between the edges and vertices of obstacles in the environment, or connect to IP or FP, in such a way that the resulting path is C^1 .

Definition 2.8 *The class of trajectories formed by concatenating simple paths that satisfy the necessary orientation constraints at the points of $\partial\Omega$, are called the canonical trajectories.*

⁶By "pass through finitely many times" we mean the curve is partitioned into finitely many intervals $[s_i, s_{i+1}]$ for which $X(s) \subset \partial\Omega$, $\forall s \in [s_i, s_{i+1}]$ and $X(s) \cap \partial\Omega = \emptyset$, $\forall s \in (s_{i+1}, s_{i+2})$.

Finally, by combining the corollaries to Theorem 2.3 and Theorem 2.5, we arrive at our method for finding a path satisfying the constraints. If a path exists, then there is a minimal-length path. Furthermore, any minimal-length path is contained in the set of canonical trajectories. Since we know that all canonical trajectories can be described by sequences of simple paths, searching for a collision-free minimal length path in the set of canonical trajectories can be reduced to finding the collision-free simple paths which, when joined in the appropriate fashion, form a complete C^1 trajectory from (IP, θ_i) to (FP, θ_f) of minimal length.

2.3 Geometry of the Simple Paths

The solution which we have devised depends on the relative simplicity of the geometry of the components of the simple paths: the circle and the straight line. Given two circles, there are four possible lines tangent to both if the two circles do not touch. If they overlap, then they share only two common tangent lines. However, if it exists, the tangent which forms the t-line is unique. This is because the t-line must satisfy additional constraints. First, the t-circles are oriented circles.⁷ There are at most two lines which are tangent to both circles and satisfy the orientation constraints. Furthermore, we have an initial t-circle and a final t-circle. Thus, of the two tangents which satisfy the constraints, only one is oriented to pass from the initial t-circle to the final t-circle. It is this tangent which forms the t-line. If we determine the t-line by first finding all of the tangents between two circles and throwing away the extraneous solutions, we will have increased the order of the equations which must be solved, and hence the time required to solve them. We will show that in order to determine the proper tangents and directions of travel along the trajectories the only geometrical operations needed are to solve for the intersections between circles, lines and line segments. These can all be done in closed form. Furthermore, if we use the standard substitution of

$$x = \tan \frac{\theta}{2} \quad (2.9)$$

⁷An **oriented circle** is a circle along with a direction clockwise or counterclockwise.

we only need to solve *quadratic* equations.

We will start off by giving some terminology which will allow us to refer unambiguously to the geometric objects making up the simple paths. The circles of radius R which contribute the arc portions of the trajectory are called **t-circles**. The straight-line portion of the path is called the **t-line**. In some cases, we will use this term to refer to the infinite line which supports the t-line.

In order to remove the symmetry of the describing equations, we often solve for the centers of the t-circles which are tangent to a given t-line. This technique brings with it another set of lines and circles, which are related to the t-circles and t-lines. For a given fixed point $p \in \mathbb{R}^2$, the locus of points defined by the centers of all the t-circles passing through p is also a circle, which we will call the **CCC** or circle of curvature centers. Associated with the t-line are two lines parallel to it at a distance equal to the fixed radius of curvature. Each line is called an **LCC** or line of curvature center. There are two LCC's to account for the direction of the turn on each end of the t-line, whether right hand or left hand.

The utility of this description of the geometry is illustrated in Figure 2. In this drawing, we show two of the four trajectories connecting IP and FP that pass along the t-line (shown as a thick line). The CCC's are the dashed circles centered at IP and FP. The LCC's associated with the t-line are indicated by the parallel dashed lines. Only one of the LCC's intersects with the CCC's. The intersection points determine the centers of the t-circles, which are marked **tci** and **tcf**. The t-circles themselves are divided into two arcs, the dark one represents the portion actually traversed by the corresponding trajectory. This figure shows that it is possible to represent a trajectory by choosing a point on each of the CCC's, and deciding which of the LCC's should pass through these points.⁸

We are often interested in knowing how many times a t-circle intersects a given obstacle edge. For a given R , we define the **tube of radius R about an edge** to be the locus of all centers of circles

⁸It may not always be possible to choose the LCC's arbitrarily. The centers of the t-circles must be separated by at least $2R$ for the two CCC's to be intersected by the two different LCC's.

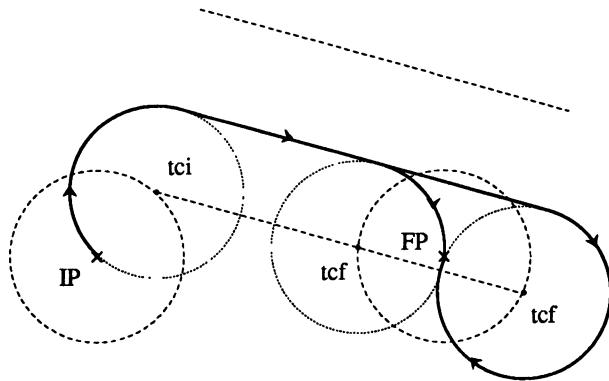


Figure 2: Encoding Trajectories by the Intersections of the LCC and CCCs

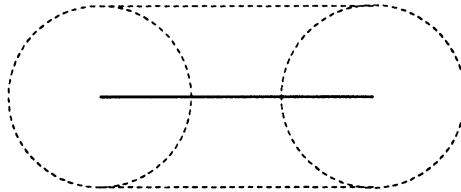


Figure 3: Tube About an Edge

of radius R which either lie tangent to an obstacle edge or pass through an obstacle vertex. See Figure 3 for a picture of the tube. The regions cut out by the dashed lines in the drawing correspond to locations of centers of a circle of radius R which intersects the edge in 0, 1 or 2 places. By intersecting the CCC of a given point with the tube about an edge, we can easily determine the number of times a t-circle passing through the point intersects the edge.

3 Configuration Spaces for Simple Paths

Once we have conceptually divided a complete trajectory connecting IP to FP into a linked sequence of simple paths, we confront the problem of determining which of the sub-trajectories are collision-free. There are an infinite number of simple paths for each pairing of obstacle vertices or edges. In order to determine which ones

are collision-free, we construct a configuration space representation. Note that the configuration space we are discussing is not for the robot, but rather for the entire simple path.⁹ In this section, we specify how the configuration spaces are constructed, and then discuss mapping the workspace obstacles into this configuration space. We will see that there is a particularly simple characterization of the configuration space obstacle.

The construction of the configuration space which is presented here is exact. No approximations have been made. Because it is based only on geometry, there are no restrictions on the direction of travel (ie. the robot could travel in either direction along these simple paths).¹⁰ We will describe two approximate search algorithms which incorporate the restriction on the direction of travel in Section 4.

3.1 Parametrizing the Trajectories

In order to construct a configuration space, we must first have a unique way of specifying the simple paths using few variables. We show how this is possible by examining the geometry of the simple paths.

For a given location and orientation, there are two oriented circles of maximum curvature along which the robot can effect a change in orientation. One corresponds to a left hand turn, the other a right hand turn. For a path which consists of an arc followed by a straight line motion followed by an arc, there are then four possible combinations of turns at the two endpoints. We denote these paths as LL,LR,RL,RR, where the L and R denote left and right hand turns, and the ordering denotes at which endpoint of the path the turn is made. In the case that the trajectory is of the three

⁹The analogy to the standard usage of configuration space is more clear if, for example, we consider the simple paths connecting two vertices as defining a mechanical linkage whose "joint angles" are the oriented tangent directions at the endpoints of the path.

¹⁰Of course, we no longer have the theorem which gives the canonical form of the trajectories if we allow backing up, because this violates the smoothness assumption on which Theorem 2.5 and its corollary were based.

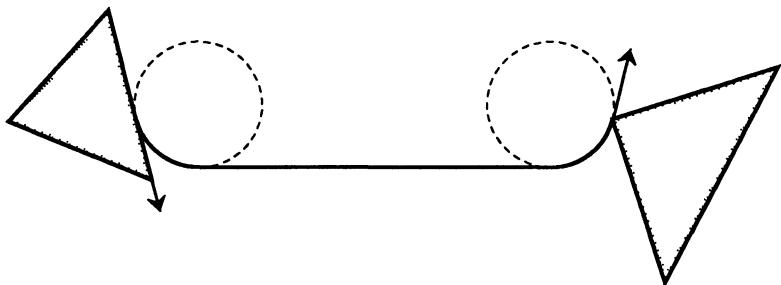


Figure 4: Subpath Connecting an Edge and Vertex

arc type, there are only two possibilities, LRL or RLR. Furthermore, Dubins [Dub57] has shown that the middle arc of the three arc type path have length $\geq \pi R$. This fact resolves the ambiguity about which of the two possible circles is the one used in the three arc path; we only consider the one satisfying this condition. Thus there are six types of paths. These we will call the **turn types**, and denote them $\mathcal{T} = \{LL, LR, RL, RR, LRL, RLR\}$. A simple path is completely determined by the initial and final orientations, turn type, initial and final locations, and maximum curvature of the path.

The simple paths begin and end at points in the boundary of the set of obstacles.¹¹ At each endpoint of a simple path, the trajectory may lie along an obstacle edge or pass through an obstacle vertex. We can parameterize all the simple paths which pass between two obstacle contacts using two parameters for each turn type. If a trajectory touches an obstacle edge, then the curve must be tangent to the edge at the point of contact. This essentially fixes the robot's orientation, but allows the initial (or final) position to be anywhere along that edge. If a path passes through a vertex, then its location is known, but its orientation is free. Therefore, for every simple path of a given turn type there is one parameter each which specifies the initial and final circular arcs traversed. This is illustrated in Figure 4. The dark line represents an LL type path which starts from a point on the left-hand obstacle edge and ends at the obstacle vertex on the right heading in the direction shown. The initial

¹¹We can consider simple paths starting at IP or ending at FP in the same manner as those which start or end at obstacle vertices

direction¹² and the final position constraints which must be satisfied are evident.

3.2 Mapping the Obstacles to C-Space

Because we can parametrize the simple paths in a unique fashion, we can construct a configuration space for each turn type and each pair of obstacle vertices and edges. For example, for a vertex to vertex path, the configuration space consists of the product of the sets of initial and final orientations with the set of turn types, $S^1 \times S^1 \times \mathcal{T}$. The other cases are analogous, where the point along an edge is parameterized by a normalized distance along that edge.¹³ The obstacles in the work space can then be mapped to the configuration space. If the trajectory represented by a point in C-space enters the interior of an obstacle, then the point is in the C-space obstacle. The complete configuration space for the environment consists of the product of the configuration spaces for all pairings of obstacle vertices, edges, IP and FP.

An example of a configuration space is shown in Figure 5. The environment is depicted in two drawings in the upper half of the picture. In the lower portion, we have two pictures of the configuration space for every RL simple path that connects the two endpoints shown as small circles.¹⁴ For ease of discussion, we will often consider the torus of configuration space as it is shown here, cut apart and laid out flat so that the x-axis corresponds to the initial directions of travel and the y-axis to the final directions.¹⁵ The axes run from 0 to 2π . We show two sample simple paths which are represented in configuration space by the small x's in the square below the corresponding picture. The figure on the left shows the constraint curves that are described later in this section. The figure on the right shows the configuration space obstacle.

¹²Note that the combination of the turn type and the fact that the initial point lay on an edge combine to uniquely determine the initial direction.

¹³In this case, the corresponding parameter space S^1 is replaced by $I = [0, 1]$.

¹⁴The circle in the lower left represents the starting endpoint.

¹⁵The opposite edges of the squares are identified to make a torus.

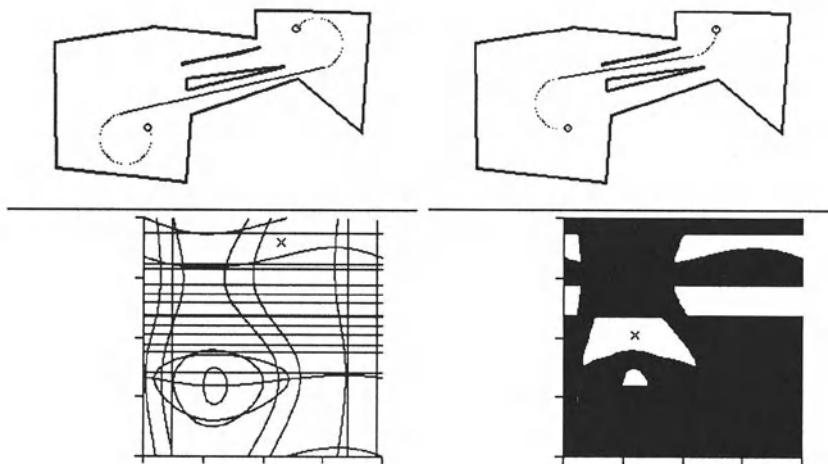


Figure 5: Configuration Space for Simple Paths

In this section, we show how the workspace obstacles are mapped into configuration space. In deriving the techniques for computing the configuration space obstacle for any one of the turn types, we show that the curves which can form its boundary can be enumerated and that they are particularly simple to compute.

We define curves which cut out regions in the configuration space with the following properties.

1. Any trajectory can be continuously deformed into any other trajectory in the same region by a change of the initial and/or final parameters.
2. The number of intersections of every trajectory in a region with any obstacle edge is constant.

We define the number of intersections for a region to be the number of intersections along any trajectory in the region. This number is well-defined for each region where the trajectories exist¹⁶. A region is part of the configuration space obstacle if the associated number of intersections is non-zero.

¹⁶We discuss below the regions in which the trajectory does not exist.

Theorem 3.10 (Four Type Theorem) *There are exactly four types of constraints which can form the boundary of a configuration space obstacle for a curve of type LL, RR, LR, or RL:*

Type A *These constraints are associated with the angles for which a t-circle either passes through an obstacle vertex or lies tangent to an obstacle edge.*

Type B *This curve is generated by the angles for which the point of tangency of the t-line with a t-circle is at either the initial or final endpoint of the simple path.*

Type C *This constraint defines the angles for which the t-line intersects a particular vertex of an obstacle polygon between the points where it is tangent to the t-circles.*

Type D *For RL and LR paths, these are the angles for which the t-circles are osculating.¹⁷ It is not defined for the LL and RR turn types.*

A superset of curves with these properties is also discussed by Fortune and Wilfong[FW88].

The constraint curves and their method of generation are similar for an edge or vertex contact. In the remainder of this section, we discuss the case of a simple path connecting two vertices. The constraint curves are illustrated in the left-hand portion of Figure 5.

The Type A constraints involve the t-circle at only one end of the simple path, and thus they hold for a fixed angle on one axis and all angles on the other.¹⁸ The angles can be found by intersecting the appropriate CCC with the tube of radius R around the obstacle edge. The Type A constraints indicate when it is possible for the simple path to intersect an obstacle edge along a t-circle segment and when it is possible for the number of intersections with an edge to change as the initial and final directions are varied. The

¹⁷Two t-circles are osculating when their centers are separated by exactly $2R$.

¹⁸The Type A constraints are the straight horizontal and vertical lines in the lefthand portion of Figure 5.

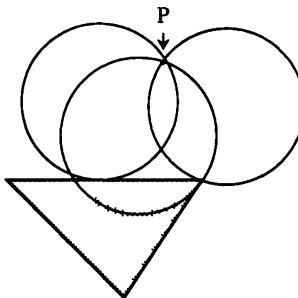


Figure 6: Type A Constraints

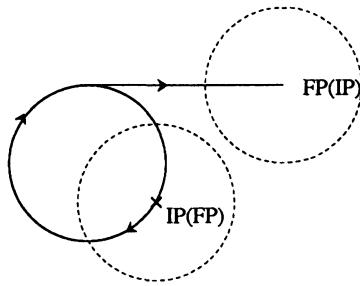


Figure 7: Type B Constraints

geometry of this constraint is shown in Figure 6, where the solid circles represent the three possible t-circles passing through P that are either tangent to the edge or pass through one of its vertices. In this illustration, we can see that the circles with centers between the leftmost circle and the middle circle will intersect the top edge twice. Those between the middle circle and the rightmost will intersect the top edge of the obstacle once. All other circles of radius R which pass through point P will not intersect the top edge at all.

The Type B constraints account for the fact that the deformation of the simple paths is not continuous as the initial and final angles are varied. The discontinuity occurs as the oriented common tangent point to a t-circle crosses one of the endpoints of the simple path (which indicates that the robot could proceed straight ahead in that direction to or from the endpoint generating the constraint). When a Type B curve is crossed, the arc traveled along the corresponding t-circle goes from being the entire perimeter of the circle to being a very small portion of it (or vice-versa depending upon the direction that the tangent moved across the endpoint). Thus,

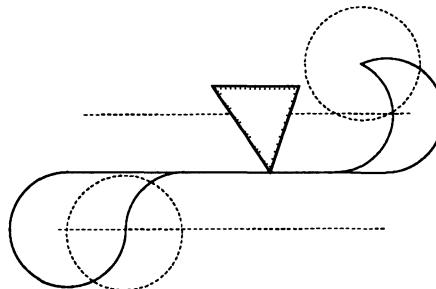


Figure 8: Type C Constraints

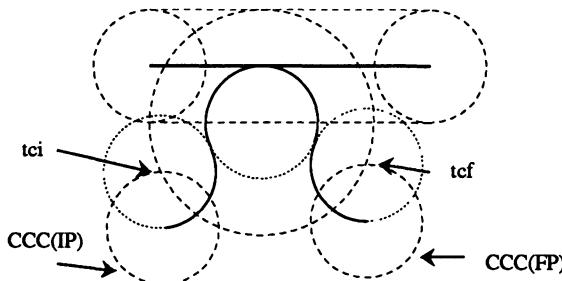


Figure 9: Type C' Constraints

if a t-circle intersects an obstacle, on one side of the constraint the simple path will include the portion of the t-circle which intersects the obstacle edge; on the other side it may not.

A Type C constraint corresponds to the t-line just touching a vertex of an obstacle. Crossing this curve by changing either the initial angle or the final angle determines whether the t-line passes through the polygon or passes completely clear of it. For a curve of type LRL or RLR, there is no straight-line path, and hence no Type C constraint. There is an additional constraint which we will call **Type C'**. It corresponds to the curve in configuration space generated by the angles for which the middle arc passes through an edge or vertex of an obstacle.

For all of the turn types except LL and RR, there are conditions under which no path of the type will exist. For LR and RL paths, this happens if the t-circles overlap. The boundary condition occurs when the t-circles are just touching. The Type D boundary cuts C-space into two regions, one in which the trajectories exist, and one

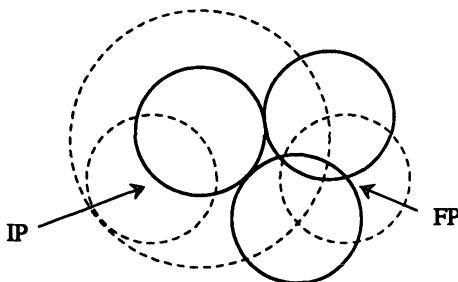


Figure 10: Type D Constraints

in which they do not. Type D constraints occur only for trajectories which change the sign of their curvature along the path segment. If the two t-circles overlap, there is no LR or RL path which can connect them. Similarly, the existence of an LRL or RLR trajectory requires that the centers of the t-circles be separated by less than four times the minimum radius of curvature. If the separation is larger, there will be no three arc path possible. The constraint is calculated in a different manner than Type D, so to distinguish the two, we will call the constraint **Type D'**.

3.3 Proof of the Four Type Theorem

3.3.1 Overview

The proof of Theorem 3.10 is made by examining the changes in the number of intersections of a trajectory with a line segment, specifically an edge of a polygonal obstacle. We will examine the constraints generated when we wish to define regions for which the deformation of a trajectory is continuous and the trajectories exist. In this discussion we will consider the paths which connect one vertex to another, and just the arc-line-arc paths. The discussion follows along similar lines for the other types of simple paths.

The essence of the proof is to show that the regions which we have described in Section 3.2 are disjoint open sets separated by the constraint curves. Thus, we can not move by a continuous change of initial or final directions from one region to another without cross-

ing a constraint. Furthermore, the union of the regions and the constraint curves make up the entire torus of initial and final directions.

Fix IP, FP, and the minimum radius of curvature $R > 0$. We have discussed in Section 3.1 that the simple paths of a given turn type which pass from one vertex in the environment to another can be parametrized by the initial and final directions of travel. The product of the initial and final directions forms a torus ($S^1 \times S^1 = T^2$). A simple path is formed from two types of geometric objects: circles and line segments. We want to consider the changes in a simple path as the initial and final directions are changed.¹⁹ If we continuously change θ_i and θ_f the trajectory deforms continuously, as long as no Type B constraint curves are crossed.

The direction to the center of the t-circle always differs from the direction of travel by $\pm\frac{\pi}{2}$ in angle. However, the sign depends upon the turn type of the simple path and whether the t-circle in question is the initial or final t-circle. Once the turn type is chosen, however, the signs are fixed. We can therefore more easily parametrize the simple paths without regard to turn type by using the direction to the center of the t-circle. This makes the proofs simpler. Henceforth in this section, *all angles will refer to the direction to the center of the appropriate t-circle unless otherwise noted.*

For a given turn type, define $l_i(\theta_i, \theta_f), l_f(\theta_i, \theta_f) \in \mathbf{R}^2$ to be the points of tangency of the common tangent to the oriented circles of minimum radius passing through IP and FP. In some cases these functions will not be defined, as when the turn type is LR or RL and the t-circles overlap. In the first half of the discussion, we ignore the obstacles and just examine the regions for which the trajectories exist. For the turn types LL and RR, the functions $l_i(\theta_i, \theta_f)$ and $l_f(\theta_i, \theta_f)$ are defined for all pairs (θ_i, θ_f) . In the case of LR and RL simple paths, we remove the pairs of directions for which $l_i(\theta_i, \theta_f) = l_f(\theta_i, \theta_f)$ from the torus. In general, this cuts the torus into two connected regions. In one of these regions the t-circles do not overlap, and hence $l_i(\theta_i, \theta_f)$ and $l_f(\theta_i, \theta_f)$ are defined. These regions are separated by the Type D constraints. See Figure 22 for

¹⁹Changing θ_i and θ_f corresponds to choosing a different point on the torus of initial and final directions.

an illustration.

Then we include the effects of the obstacles. We reduce the analysis by examining the constraints imposed by a single line segment, representing one edge of an obstacle. As the simple path deforms from changes in θ_i and θ_f , the number of times it passes through this obstacle line segment may change. We break this down further, separately considering intersections which occur along the t-circle and t-line portions of the simple path. The intersections due to the t-circles are the simplest to analyze. These are associated with the Type A constraints. We simply find the angles for which the initial and final t-circles pass through the endpoints of a line segment or lie tangent to this segment. Because these constraints hold for one fixed angle on one end and all angles on the other, these constraints divide the torus into a connected region. We will see that these constraints correspond to boundaries between regions for which the t-circle intersects the obstacle line segment and where it does not. Suppose, for a given θ_i and θ_f , a simple path intersects the obstacle line segment on a t-circle portion. We consider how we can deform the path by changing the pair (θ_i, θ_f) so that the line segment and t-circle no longer intersect. There are only a few possible ways that this can happen if the simple path deforms continuously. One is that the point of intersection passes through the endpoint of the obstacle line segment. Another is that the point of intersection is a point of tangency of the circular arc to the line segment. These two are taken care of by the Type A constraints discussed earlier in this paragraph. The only other possibility is that the intersection moves from the arc portion of the path to the straight line part. Note that in this situation, the simple path still intersects the line segment, so we need not produce another constraint curve to account for this.²⁰

We also need to consider the intersections between the t-line part of the trajectory and the obstacle line segment. A change in (θ_i, θ_f) changes the endpoints of the t-line segment. Now suppose that the t-line segment and the obstacle line segment intersect. This intersection can be removed only if the endpoint of one line segment goes through the other line segment. In the case that the t-line goes

²⁰This is fortunate since this constraint is no longer quadratic, as are all the others.

through the obstacle line endpoint, we have a Type C constraint. When the t-line endpoint goes through the obstacle line segment, there is really no constraint, because the simple path still intersects the line segment, only now the point of intersection is on the t-circle portion.

In the actual proof, we parametrize the geometric objects which are the t-circle and t-line segments. We find open sets in the space of these parametrizations which represent various ways the geometric objects can intersect. Then we show that the functions which map from the torus of initial and final directions to the space of parametrizations are continuous over an open domain. Thus, by taking inverse images using these continuous functions we get disjoint open sets on the torus which are separated by sets corresponding to the constraint curves.

3.3.2 Type D Constraints

To start off the proof, we need to show that the set over which the simple paths are defined is an open set on the torus of initial and final directions that has as its boundary a set defined by the Type D constraints. Since the simple paths always exist for turn types LL or RR, the analysis in this section only applies to the other cases. In particular, we examine the LR and RL turn types.

Fact 3.11 *Two circles in \mathbb{R}^2 can intersect in 0, 1, 2, or ∞ points.*

Lemma 3.12 *Assume that $IP \neq FP$. We can divide the torus into three sets.*

1. *An open set consisting of*

- *the angles for which the t-circles intersect transversally, where the set of intersection points is non-empty.*
- *the angles for which the t-circles coincide.*

2. An open set consisting of angles for which the two t-circles do not intersect.
3. A set consisting of angles for which the t-circles are osculating which corresponds to the Type D constraints.

Proof. Without loss of generality, we can set the radius of the t-circles $R = 1$, IP = $(0,0)$, and FP = $(\Delta x, 0)$ where $\Delta x \neq 0$.

The center of the initial t-circle is then

$$\mathbf{tci} = (\cos \theta_i, \sin \theta_i) \quad (3.13)$$

The center of the final t-circle is then

$$\mathbf{tcf} = (\Delta x + \cos \theta_f, \sin \theta_f) \quad (3.14)$$

We wish to consider the function

$$f(\theta_i, \theta_f) = \|\mathbf{tci} - \mathbf{tcf}\|^2 \quad (3.15)$$

We note that, in this case

$$f : S^1 \times S^1 \rightarrow \mathbb{R} \quad (3.16)$$

Clearly, f is a continuous function of (θ_i, θ_f) . The sets enumerated above can be described using f as follows.

1. This set is open by continuity of f , because it is $f^{-1}((-\infty, 4))$.
2. This set is open by continuity of f , because it is $f^{-1}((4, \infty))$.
3. This set is closed, being the inverse image of a closed set $f^{-1}(4)$. Furthermore, by definition, these are the Type D constraints.



Remark. This partitioning of the torus is important only in the case of LR and RL simple paths. The second set along with the

preimage $f^{-1}(4)$ is the region of the torus over which such simple paths are defined, since the common tangent to the two t-circles that forms the t-line exists in this region. Clearly, if $\Delta X \neq 0$, the second set is non-empty.

Remark. In the case that $IP = FP$ ($\Delta x = 0$), the only trajectories for which the t-circle centers are separated by $2R$, return the robot to the same location and direction as it started in. Thus, in the case of LR or RL turn types, which are the only cases in which Type D constraints matter, we can consider the configuration space obstacle to be all of T^2 when $\Delta x = 0$. In the planning algorithm, we therefore never compute a configuration space for LR or RL simple paths where the two endpoints of the path are at the same location.

Remark. In the rest of this proof, if the turn type is LR or RL, we assume that the domain of the functions $l_i(\theta_i, \theta_f)$ and $l_f(\theta_i, \theta_f)$ is the open set $f^{-1}((4, \infty))$, on which they are defined.

3.3.3 Trajectory Constraints

In this section we consider when the trajectory will change the number of intersections which it makes with objects in the environment. We can analyze this situation by considering a single line segment fixed in the plane as the obstacle. In order to simplify the analysis, we consider the trajectory in pieces. First, we will determine the angles for which the straight line portion of the trajectory may pass through the obstacle line segment. Then we consider the arc sections. These pieces of the simple path are separated at the points of tangency of the t-line with the two t-circles.

Notation Consider two points $a, b \in \mathbb{R}^2$. We denote the line which passes through these points as $l(a,b)$ for $a \neq b$. We denote the open line segment $s(a,b) = \{x \in \mathbb{R}^2 \mid x = \lambda a + (1 - \lambda)b \text{ where } \lambda \in (0, 1)\}$. We denote the closed line segment $s[a,b] = \{x \in \mathbb{R}^2 \mid x = \lambda a + (1 - \lambda)b \text{ where } \lambda \in [0, 1]\}$.

Fix two distinct points $c, d \in \mathbb{R}^2$. The closed line segment $s[c,d]$

in the plane will be the obstacle which we consider in this section.

3.3.4 T-line Constraints

Here we consider the intersections of the straight line portion of the set of simple paths passing between two vertices. Now we can partition \mathbb{R}^4 into four sets according to the number of intersections of $l(a,b)$ and $l(c,d)$.

$$\begin{aligned} Z &= \{(a, b) \in \mathbb{R}^4 \mid a = b\} \\ I_0 &= \{(a, b) \in \mathbb{R}^4 \setminus Z \mid l(a, b) \cap l(c, d) = \emptyset\} \\ I_\infty &= \{(a, b) \in \mathbb{R}^4 \setminus Z \mid l(a, b) = l(c, d)\} \\ I_1 &= \mathbb{R}^4 \setminus (Z \cup I_0 \cup I_\infty) \end{aligned}$$

We can also partition \mathbb{R}^4 into five sets according to the number of intersections of the line segments $s[a,b]$ and $s[c,d]$.

$$\begin{aligned} Z &= \{(a, b) \in \mathbb{R}^4 \mid a = b\} \\ N &= \{(a, b) \in \mathbb{R}^4 \setminus Z \mid (s(a, b) \cap s(c, d) = \emptyset) \wedge (s[a, b] \cap s[c, d] \neq \emptyset)\} \\ S_0 &= \{(a, b) \in \mathbb{R}^4 \setminus Z \mid s[a, b] \cap s[c, d] = \emptyset\} \\ S_\infty &= I_\infty \setminus (S_0 \cup N) \\ S_1 &= \mathbb{R}^4 \setminus (Z \cup N \cup S_0 \cup S_\infty) \end{aligned}$$

We will show that the regions S_0 and S_1 are open disjoint subsets of \mathbb{R}^4 . We will later note that the sets Z, N , and S_∞ are all taken care of by the constraints we have described. The consequence is that if we pass from S_0 to S_1 by any continuous change of (a, b) , then we must pass through one of the sets defined by the constraints.

Lemma 3.17 S_1 is open.

Proof. Clearly $S_1 \subset I_1$. Fix $c, d \in \mathbb{R}^2$, with $c \neq d$. Take any point $(a, b) \in S_1$, then by definition $\exists! p \in \mathbb{R}^2$ such that $s(a, b) \cap s(c, d) = \{p\}$. We can express the point $p \in \mathbb{R}^2$ using the vector equation

$$p = \lambda_p a + (1 - \lambda_p)b \tag{3.18}$$

$$= \gamma_p c + (1 - \gamma_p)d \tag{3.19}$$

By combining these equations and eliminating either λ or γ we can see that λ_p and γ_p are the zeroes of two polynomials in λ and γ , respectively, whose coefficients are continuous in a and b . Since the number of roots of both these polynomials is constantly equal to 1 in S_1 , we know that the roots λ_p and γ_p are continuous functions of a and b . Call them $\lambda_p(a, b)$ and $\gamma_p(a, b)$. Because $s(a, b)$ and $s(c, d)$ are open segments, we know $\exists \epsilon > 0$ such that $(\lambda_p - \epsilon, \lambda_p + \epsilon) \subset (0, 1)$ and $(\gamma_p - \epsilon, \gamma_p + \epsilon) \subset (0, 1)$. The intersection of the inverse images of these sets under $\lambda_p(a, b)$ and $\gamma_p(a, b)$ is relatively open in I_1 . Since I_1 is clearly open in \mathbb{R}^4 , we are done. ■

Remark. The intersection of the inverse image of the sets $(\lambda_p - \epsilon, \lambda_p + \epsilon)$ and $(\gamma_p - \epsilon, \gamma_p + \epsilon)$ is the set of (a, b) such that the intersection of the two lines $l(a, b)$ and $l(c, d)$ is within the open segments $s(a, b)$ and $s(c, d)$. We have shown that this set of (a, b) is an open subset of \mathbb{R}^4 .

Lemma 3.20 S_0 is open.

Proof. Choose any $(a, b) \in S_0$. We use the standard notion of a set distance function d , to define a mapping $\mathbb{R}^4 \setminus Z \rightarrow \mathbb{R}$ by $(a, b) \mapsto d(s[a, b], s[c, d])$. This is a continuous function of a and b since the points of $s[a, b]$ are. Since the two sets are compact, there is a minimum distance ϵ attained by some point of $s[a, b]$. By definition of S_0 , $s[a, b] \cap s[c, d] = \emptyset$, and therefore $\epsilon > 0$. Since the distance function is continuous, the inverse image of the open set $(\frac{\epsilon}{2}, \infty)$ is open. It contains (a, b) and is clearly in the set S_0 . Thus, S_0 is relatively open in $\mathbb{R}^4 \setminus Z$ which is itself an open set in \mathbb{R}^4 . ■

Remark. Because these two sets are open and disjoint, if we pass by a continuous change of the endpoints of the line segment from S_0 to S_1 , we must pass through one of the sets N, Z , or S_∞ . However, we can see that, with one exception, the points in these sets correspond to one of the constraints specified in the Theorem 3.10.

Lemma 3.21 The map from (θ_i, θ_f) to the endpoints of the t-line is continuous over its domain.

Proof. We prove this by simply writing out the formula. We will denote the point of tangency of the t-line with the initial(final) t-circle as $l_i(\theta_i, \theta_f)(l_f(\theta_i, \theta_f))$. As we have previously, the centers of the t-circles are denoted tci and tcf , respectively.

$$tci = IP + R \begin{pmatrix} \cos \theta_i \\ \sin \theta_i \end{pmatrix}$$

$$tcf = FP + R \begin{pmatrix} \cos \theta_f \\ \sin \theta_f \end{pmatrix}$$

$$\theta_{if} = \arctan 2(tcf.y - tci.y, tcf.x - tci.x)$$

$$\theta_{inter} = \arccos \left(\frac{2R}{\|tci - tcf\|} \right)$$

The first of these two angles measures the angle between the centers of the two t-circles. The second one is the angle formed by the common tangent to the centers of the t-circles with respect to the line connecting the t-circle centers. Now we have a different function depending upon the turn type of the simple path in question.

LL

$$l_i(\theta_i, \theta_f) = tci + R \begin{pmatrix} \cos(\theta_{if} - \frac{\pi}{2}) \\ \sin(\theta_{if} - \frac{\pi}{2}) \end{pmatrix}$$

$$l_f(\theta_i, \theta_f) = tcf + R \begin{pmatrix} \cos(\theta_{if} - \frac{\pi}{2}) \\ \sin(\theta_{if} - \frac{\pi}{2}) \end{pmatrix}$$

RR

$$l_i(\theta_i, \theta_f) = tci + R \begin{pmatrix} \cos(\theta_{if} + \frac{\pi}{2}) \\ \sin(\theta_{if} + \frac{\pi}{2}) \end{pmatrix}$$

$$l_f(\theta_i, \theta_f) = tcf + R \begin{pmatrix} \cos(\theta_{if} + \frac{\pi}{2}) \\ \sin(\theta_{if} + \frac{\pi}{2}) \end{pmatrix}$$

RL

$$l_i(\theta_i, \theta_f) = tci + R \begin{pmatrix} \cos(\theta_{if} + \theta_{inter}) \\ \sin(\theta_{if} + \theta_{inter}) \end{pmatrix}$$

$$l_f(\theta_i, \theta_f) = tcf - R \begin{pmatrix} \cos(\theta_{if} + \theta_{inter}) \\ \sin(\theta_{if} + \theta_{inter}) \end{pmatrix}$$

RL

$$\begin{aligned} l_i(\theta_i, \theta_f) &= tci + R \begin{pmatrix} \cos(\theta_{if} - \theta_{inter}) \\ \sin(\theta_{if} - \theta_{inter}) \end{pmatrix} \\ l_f(\theta_i, \theta_f) &= tcf - R \begin{pmatrix} \cos(\theta_{if} - \theta_{inter}) \\ \sin(\theta_{if} - \theta_{inter}) \end{pmatrix} \end{aligned}$$

We can see that these are continuous functions except in the case that $\|tci - tcf\| \leq 2R$ for the turn types LR and RL. ■

In the following lemmas, we consider the function mapping $T^2 \rightarrow \mathbb{R}^4$ by $(\theta_i, \theta_f) \mapsto (l_i(\theta_i, \theta_f), l_f(\theta_i, \theta_f))$.

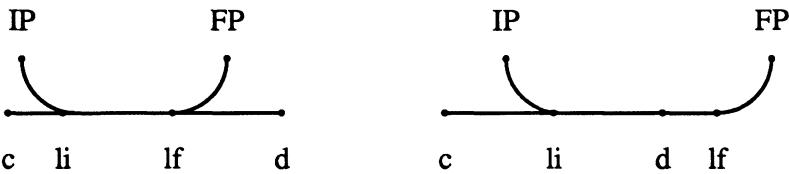
Lemma 3.22 *The inverse image of S_1 and S_2 are open disjoint subsets of the torus of initial and final directions.*

Proof. By continuity, the inverse images of these sets are relatively open in the domain of the function $l_i(\theta_i, \theta_f)$ and $l_f(\theta_i, \theta_f)$. By Lemma 3.12, the domain is open, thus the inverse images are. ■

Lemma 3.23 *The inverse images of the sets N , Z , and S_∞ correspond to constraints.*

Proof. By inspection:

Set N These are the Type C constraints when the endpoint of the obstacle line segment (c or d) lies along the interior of the t-line(s(a,b)). Otherwise, if the endpoint of the t-line passes through the interior of the line segment, it is not a constraint since a small change in θ_i or θ_f will cause the point of intersection to lie either on the relative interior of the t-line or on the t-circle. In both cases, there will remain an intersection of the entire trajectory with the obstacle edge.

Figure 11: Two simple paths from the set S_∞

Set Z For the turn types LR and RL, we can examine the function $l_i(\theta_i, \theta_f)$ and $l_f(\theta_i, \theta_f)$ to see that they are equal when the distance between the t-circle centers is exactly $2R$. Such points (θ_i, θ_f) are specified by the Type D constraint. In the case of LL and RR turn types, it must be that tci and tcf are equal. This means that one circle passes through both IP and FP. In this case the points (θ_i, θ_f) are specified by the Type B constraint.

Set S_∞ These are either Type A constraints or Type C constraints or both, depending upon which segment's endpoints lie within which segment. This is illustrated in Figure 11. On the left, the two constraints are both Type A, since the t-circles are tangent to $s[c,d]$. On the right, one constraint is Type A and the other is Type C, since the t-line goes through the endpoint d.

■

3.3.5 T-Circle Constraints

In this case we define the regions in which an arc of a circle of fixed radius may have a given number of intersections with a fixed line segment in the plane.

Fact 3.24 *The only non-transversal intersection of a circle with a line is when the circle lies tangent to the line.*

In the present case, we are interested in the intersection of a circular arc with the line segment $s[c,d]$. Because the arc portions of the

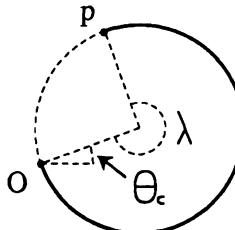


Figure 12: Parametrizing a t-circle arc

simple path are constrained to have one endpoint at a fixed point, we will assume, without loss of generality, that the arc must start at the origin. We also assume that its radius is 1 and that the arc extends in the counterclockwise direction from the origin. We describe the t-circle by the angle to its center which is at a position $\text{tci}(\theta_c)$.

$$\text{tci}(\theta_c) = \begin{pmatrix} \cos \theta_c \\ \sin \theta_c \end{pmatrix} \quad (3.25)$$

We can parametrize a point $p(\theta_c, \lambda)$ on this circle a distance λ from the origin by the equation

$$p(\theta_c, \lambda) = \begin{pmatrix} \cos \theta_c - \cos(\theta_c + \lambda) \\ \sin \theta_c - \sin(\theta_c + \lambda) \end{pmatrix} \quad (3.26)$$

See Figure 12 for an illustration of the parametrization. Because we don't want the simple path to follow a t-circle farther than $2\pi R$ in distance, we restrict $\lambda \in [0, 2\pi]$. Given this restriction, we can unambiguously determine λ given θ_c and p , a point on the circle. Furthermore, λ is a continuous function of θ_c and p except where $p=0$.²¹ We write it $\lambda(\theta_c, p)$.

Notation Denote the open and closed arcs as follows.

$$\begin{aligned} r(\theta_c, \theta_l) &= \{p(\theta_c, \lambda) \mid \lambda \in (0, \theta_l)\} \\ r[\theta_c, \theta_l] &= \{p(\theta_c, \lambda) \mid \lambda \in [0, \theta_l]\} \end{aligned}$$

Now consider the t-circle portions of a simple path. It follows the initial t-circle from IP to the point of tangency $l_i(\theta_i, \theta_f)$ and from the point of tangency $l_f(\theta_i, \theta_f)$ to FP. This implies that if we fix an arc length along the simple path, the point in space which is determined is a continuous function of (θ_i, θ_f) , unless $\lambda(\theta_i, l_i(\theta_i, \theta_f)) = 2n\pi$ or

²¹In this case, λ may change by 2π .

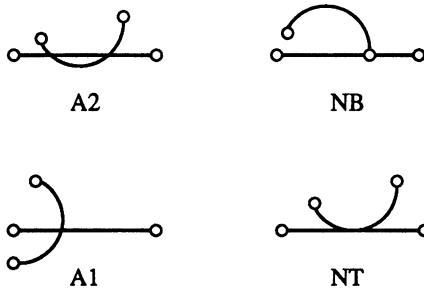


Figure 13: Possible Intersections of an Arc and a Segment

$\lambda(\theta_f, l_f(\theta_i, \theta_f)) = 2n\pi$.²² If we remove these sets of angles from the torus of initial and final angles, we can construct a continuous mapping $(\theta_i, \theta_f) \mapsto (\theta_c, \theta_l)$. The space of parametrizations is a torus once again. We can define regions of the torus corresponding to various types of intersections of an arc $r[\theta_c, \theta_l]$ and the line segment $s[c, d]$.

$$\begin{aligned}
 A_1 &= \{(\theta_c, \theta_l) \mid \exists p \text{ s.t. } (r(\theta_c, \theta_l) \cap s(c, d) = \{p\}) \wedge (r(\theta_c, \theta_l) \overline{\cap} s(c, d))\} \\
 A_2 &= \{(\theta_c, \theta_l) \mid \exists p_1, p_2 \text{ s.t. } p_1 \neq p_2, r(\theta_c, \theta_l) \cap s(c, d) = \{p_1, p_2\}\} \\
 A_0 &= \{(\theta_c, \theta_l) \mid r[\theta_c, \theta_l] \cap s[c, d] = \emptyset\} \\
 NT &= \{(\theta_c, \theta_l) \mid r(\theta_c, \theta_l) \not\cap s(c, d)\} \\
 NB &= \{(\theta_c, \theta_l) \mid (r(\theta_c, \theta_l) \cap s(c, d) = \emptyset) \wedge (r[\theta_c, \theta_l] \cap s[c, d] \neq \emptyset)\} \\
 ZA &= \{(\theta_c, \theta_l) \mid \theta_l = 2n\pi\}
 \end{aligned}$$

The geometry corresponding to these sets is depicted in Figure 13.

Lemma 3.27 A_0, A_1 and A_2 are open and disjoint.

The proof is similar to that of Lemmas 3.17 and 3.20 and will not be presented. This lemma and the previous discussion of the continuity of the mapping $(\theta_i, \theta_f) \mapsto (\theta_c, \theta_l)$ give the following lemma.

Corollary 3.28 *The inverse images of A_0, A_1 and A_2 are open and disjoint on the torus (θ_i, θ_f) under the continuous mapping which takes $(\theta_i, \theta_f) \mapsto (\theta_c, \theta_l)$.*

²²In this case, either $l_i(\theta_i, \theta_f)=IP$ or $l_f(\theta_i, \theta_f)=FP$. These angles are defined by the Type B constraints.

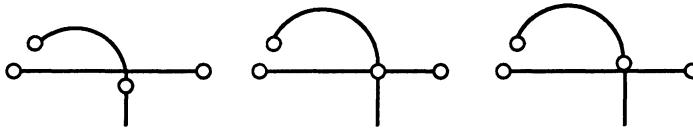


Figure 14: Intersection Moves From Arc to Line

Lemma 3.29 *The inverse images of NT, NB, and ZA correspond to constraints.*

Proof. We perform this proof by inspection.

ZA We have already discussed the set ZA. Its inverse image is the set of angles described by the Type B constraints.

NT Clearly, the inverse image of this set is a subset of the Type A constraints.

NB These points correspond to the Type A constraints for the (θ_c, θ_l) for which $r[\theta_c, \theta_l] \cap s[c, d] \subset \{c, d\}$. In the case that the intersection occurs at the endpoint of the segment, we really do not have a constraint which must be evaluated. This is because a small change in (θ_i, θ_f) will cause the intersection point to be on the relative interior of the t-line or arc, thus preserving the intersection between the simple path and the obstacle line segment. This is illustrated in Figure 14.

■

3.3.6 Combining the Results

In the preceding sections we have divided the trajectory up into pieces. We have shown that the constraints form boundaries between regions of the configuration space for which the number and type of intersections of the pieces of the simple path with an obstacle segment remain constant. It is clear that by taking the intersections of the regions found and the union of the constraints, we will have all of the regions and boundaries for the simple path as a whole. Because we have taken inverse images of the entire range of functions

which are defined on either the entire torus, or the torus minus a subset which corresponds to a constraint curve, we know that every point of the torus (θ_i, θ_f) has been determined to be either in a region or on a constraint line. Thus we have proven the assertion that the constraints specified are sufficient to partition the torus into regions with the desired properties.

3.4 Modifications When No Trajectory Exists

It is useful for the plane sweep algorithm described later in this paper to define a number of intersections even for the portions of the configuration space corresponding to unrealisable trajectories. Such regions will exist in the configuration space representation if the turn type is either LR or RL and the initial and final points of the path segment are closer than four times the minimum radius of curvature. This region is cut out by the Type D curve. It is important to define a number of intersections for the region which is consistent along its boundary. We later describe a plane sweep algorithm for computing the number of intersections for each region which makes use of the method we describe below.

The technique we use is to define a **modified trajectory** in the case that there is no trajectory in the class which is realizable. There are many possible choices for such a trajectory which have the property that they go to the correct real trajectory at the region boundary. We have chosen a particular form for the modified trajectory which is shown in Figure 15, and is described below. Strictly speaking, it is not a trajectory since it does not connect the two endpoints of the path. The two directions define t-circles through the two endpoints of the path segment. These t-circles are oriented by the turn type chosen. No LR or RL trajectory exists when the two t-circles intersect in more than one point. The modified trajectory is defined as two parts: the path traversing the initial t-circle from the starting point until it enters the final t-circle, and the path traversing the final t-circle from the point it exits the initial t-circle until it reaches the final point of the trajectory. We define **virtual endpoints** to be the endpoints created by entering and leaving the t-circle of the opposite path endpoints.

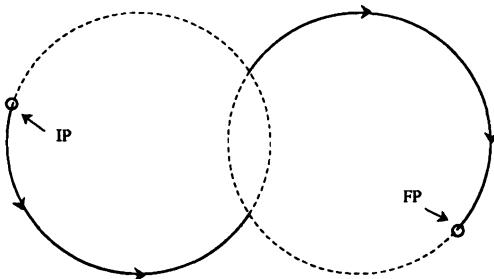


Figure 15: The Modified Trajectory

The modified trajectory goes to the true trajectory at the two extreme cases where the two t-circles have rotated such that they just touch. Furthermore, in a small neighborhood of the Type D boundary, the number of intersections along the modified trajectory is equal to the number of intersections of the region across the boundary. In order to distinguish the two regions, and to ensure that the region for which no trajectories exist is counted as an obstacle region, we define the number of intersections for the modified trajectory to be one more than the actual number of times the trajectory intersects an obstacle edge. In order to cut out regions for which the number of intersections is invariant, we must define another constraint.

Type E This constraint accounts for the virtual endpoints of the modified trajectory passing through an obstacle edge.

We also must include some additional Type B constraints in the region where the trajectories do not exist. The trajectories can not be continuously deformed into each other throughout this entire region. The discontinuities occur at the four angles, two each for IP and FP, for which a t-circle passes through both IP and FP. We can then determine when a discontinuity occurs in the modified trajectory by examining the angle of the other t-circle as the constraint is crossed.

We can define a modified trajectory for the LRL and RLR simple paths in a similar fashion. We leave the details to the reader.

4 A Grid Based Algorithm

In our presentation so far, we have discussed breaking down the problem of finding a complete path through the environment into one of finding all of the free simple paths which pass between edges or vertices of the obstacles. We have shown how we can compute the exact configuration space for these simple paths. In order to construct a near-minimal-length canonical trajectory connecting (IP, θ_i) to (FP, θ_f) , we must determine exactly which of these many free simple paths are traversed. This involves determining not just the sequence of vertices and edges touched by the complete trajectory, but also the exact position and heading at each touch and the turn type of the free simple path connecting them.

In this section, we present an algorithm which is based upon placing a grid in each configuration space. Because a configuration space represents all the simple paths of a given turn type passing between two points (or edges), each such grid point represents a particular simple path. This path connects the two points with a certain initial and final direction. Thus we can think of each grid point as representing a link between two position and orientation pairs. We can then construct a graph which encodes the connectivity of position and orientation pairs. Each node of the graph is a pairing of a specific point on the boundary of an obstacle and a fixed orientation. In addition, (IP, θ_i) and (FP, θ_f) are the start and goal nodes, respectively.²³ One graph node is adjacent to another if there exists a collision-free trajectory which starts at the latter position and orientation and connects to the former. Each link has a cost associated to it, which is the length of the corresponding path.

In summary, a simple path used by our grid-based algorithm is represented in two ways, by a grid point in the appropriate configuration space and, if it is collision-free, by a link in the graph. Once we have constructed this graph, the problem of finding an approximation to the shortest canonical trajectory reduces to finding the shortest path through this graph.

²³Henceforth, we will use the term **node** to refer to both a position and orientation pair and the abstract node representing it in the graph. The correct interpretation should be clear from the context.

4.1 Robustness of Paths

By placing a grid on each configuration space, we restrict the possible simple paths which are under consideration. This introduces the possibility that a collision-free path exists which the algorithm will not find. In this situation the algorithm will fail, without informing us that, in fact, a path exists. In this section we introduce the notion of robustness of paths, which requires that there exist trajectories in some neighborhood of the optimal. Under the assumption of robustness, the problem is again well-posed even though the algorithm searches among a restricted subset of the canonical trajectories. That is, we can give conditions under which the algorithm will find a solution if one exists.

A path is **robust** provided that

1. it is collision-free
2. if a small perturbation in orientation (position) is made at each point where the path passes through an obstacle vertex (edge), the path will remain collision-free, and will continue to pass in order through the same set of obstacle vertices and edges.

Figure 16 depicts the envelope of a trajectory segment passing between two points which is robust. The initial direction and final direction can be independently changed within the range of size δ without causing a collision. We call a path δ -robust if a change of up to $\pm \frac{\delta}{2}$ can be made in the orientation.

Remark. The assumption of δ -robustness is equivalent to requiring that a square with sides of length δ be able to fit in the free part of each configuration space through which the path passes. This represents that there is a collision-free trajectory that starts at the initial vertex with an orientation in the range of size δ and can reach the final vertex with *any* of the orientations in the range of size δ .

Robustness of a path is crucial because practical mobile robot

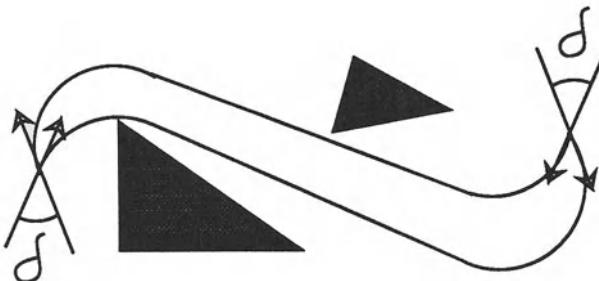


Figure 16: Envelope of a Robust Trajectory

systems are not able to follow a planned trajectory perfectly. Thus, some tolerance for errors should be built into any path-planning system. The grid trajectory which we present does not guarantee that the paths which are generated are robust, however the quadtree-based algorithm which is discussed in Section 5 does.

Figure 17 shows a configuration space with a grid laid upon it, where the grid points are separated by δ . Suppose this configuration space represents all of the LR simple paths starting at vertex v_1 and ending at vertex v_2 . As we noted above, each δ -robust path segment which connects v_1 to v_2 by an LR simple path is represented by a square in this configuration space. Due to the spacing of the grid, any δ -robust path segment has a representative square which must overlap some grid point. Suppose this grid point represented the LR simple path starting at v_1 with a heading θ_1 and ending at v_2 with a heading θ_2 . Thus, if there is a δ -robust canonical trajectory with LR simple path connecting v_1 and v_2 , then there is an approximating canonical trajectory which passes through v_1 with heading θ_1 and v_2 with heading θ_2 . This implies that we can construct approximating canonical trajectories by considering only the simple paths which correspond to grid points.

In Figure 17, we show three squares which all overlap the same grid point. This suggests that the canonical trajectory passes between v_1 and v_2 three times, in a loop.²⁴ However, the approximating trajectory does not need to have loops in it. This is because there is a collision-free trajectory which starts at v_1 with heading θ_1 that reaches v_2 with any heading which falls within the union

²⁴This may be necessary to preserve the robustness of the path.

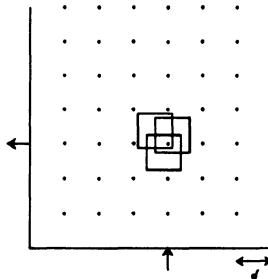


Figure 17: A Configuration Space with an Overlaid Grid

of all of these ranges. Thus, if we go on to the next vertex v_3 in the canonical trajectory, any heading which can be reached by the δ -robust canonical trajectory can be reached by the approximating trajectory which passes through grid points. Therefore, we eliminate the part of the canonical trajectory between the first touch at v_1 with heading in the range overlapping θ_1 and the last touch at v_2 with heading in the range θ_2 . In this manner, the necessity of having a loop is obviated.

The implication of the robustness requirement is that we can only find the shortest path for which there exist other collision-free paths that are close. It may be that a minimal-length path exists, but it is less robust than we have demanded. In this case, the algorithm will not find an approximation to this path, but rather of a longer one which is robust, if such a path exists. We note that some requirement of robustness is necessary for any approximation scheme to work.

Finally, there is an explicit tradeoff between the length of the approximate path found and that of a minimal length robust path in terms of the required robustness. This tradeoff will be discussed in Section 4.8.

4.2 The Search Algorithm

In this section we see how the grid technique is used to generate an approximation to the shortest canonical trajectory. The orienta-

tions are chosen to be uniformly spaced by δ for those nodes which represent the robot lying at an obstacle vertex. The positions are uniformly spaced by $\epsilon(\delta)$ for those nodes along obstacle edges. In a later section we discuss the effect of the choice of the spacing on the closeness of the approximation.

There are many well-known techniques for finding the shortest path through a weighted digraph. The search algorithm we present is based upon a graph search technique known as Dijkstra's algorithm [AHU83, pp. 203 – 208]. One could just as easily use a heuristic algorithm such as A^* [Win84, p. 113].

Nodes in the graph are in one of three states:

OPEN: A node which is in OPEN is a neighbor of some node which has been previously visited, but has not yet had its neighbors calculated.

CLOSED: A CLOSED node has been visited and its neighbors have been calculated.

UNSEEN: These comprise the remaining nodes.

Grid-Based Search Algorithm

1. Find neighbors of (IP, θ_i) and compute the distance to each along the appropriate trajectory.
2. The current node is the OPEN node with the minimum path length from IP. If the current node is (FP, θ_f) , report path. If OPEN is empty, report failure.
3. Mark the current node as CLOSED.
4. Find all of the neighbors of the current node.
5. For each neighbor which is:

CLOSED: Do nothing.

UNSEEN: Mark as OPEN. Compute the length of the trajectory from IP as the sum of the length of the path from IP to the current node plus the length from the current node to this neighbor.

OPEN: Check if the path through the current node is shorter than the one which is previously computed. If so, change the path length.

6. Goto 2.

4.3 Algorithm to Generate Neighboring Nodes

The part which remains is to generate the neighbors of a given node. For ease of presentation, we assume that the both the initial and final locations of the robot are at obstacle vertices. We will denote the current node by (v_c, θ_c) . To find its neighbors, we must look at all of the possible other nodes which can be reached from it by following a collision free trajectory. Here we use the characterization of the configuration space which we have developed in the preceding sections. For each turn type, we consider the configuration spaces for each pairing of v_c with one of the vertices in the environment. We can determine the neighboring nodes faster than by just looking at the corresponding trajectories one at a time using this approach. A link exists between two graph nodes if, for some turn type, the corresponding grid point in the appropriate configuration space does not lie within the configuration space obstacle. For a given node and a given destination vertex, the set of all possible final orientations starting from the node is represented by a line in the corresponding configuration space. This line crosses the constraint curves described in Section 3.2. The intersections with these constraint curves partition the line into segments. By the definition of these curves, either all the trajectories represented by a single segment are collision-free or all are not. In order to generate the neighbors, we utilize a sweep algorithm to determine which portions of the line represent collision-free trajectories. We then find all of the grid points which lay in those segments. These are the grid points associated with the desired links in the graph.

Remark. We should note that, in this algorithm, the entire configuration space obstacle is not generated. Only slices representing the orientations reachable from (v_c, θ_c) are calculated.

Neighbor Generation Algorithm

1. Set current final orientation to some fixed initial value.
2. Determine the number of intersections of the trajectory connecting the initial node with the current final node. Calculate the number of intersections of the initial and final t-circles with the obstacles as well.
3. Find the intersection of all of the constraint curves with the straight line representing all possible final orientations of the robot for the given initial orientation. Sort them in increasing order starting from the current final orientation.
4. If the number of intersections of the trajectory with the obstacles is zero, report the current final node as a neighbor.
5. Increment the current final orientation by δ . If we have examined all of the final orientations, stop.
6. If we have crossed any constraints, update the count of intersections for each one.
7. Go to 4.

4.4 Complexity of Neighbor Generation

In order to determine the time required for the search algorithm we must first examine the algorithm which generates the neighbors of a given node. The algorithm is run once for each possible final vertex or edge. The time required for each such step of the algorithm is given below.

1. $O(1)$.
2. Check for intersection of a trajectory with each obstacle edge. $O(n)$.
3. There are $O(n)$ intersections of the constraint curves with the vertical sweep line. To sort them takes time $O(n \log n)$.
4. $O(1)$.
5. $O(1)$. We perform this step $O(\frac{1}{\delta})$ times.
6. $O(1)$ per constraint. We cross at most $O(n)$ constraints during the entire algorithm.

To generate all the neighbors for a given node requires performing this algorithm $O(n)$ times, once for each possible final position. Hence the complexity of finding the neighbors of a node is $O(n^2 \log n + \frac{n}{\delta})$. The brute force method of checking every trajectory individually would take $O(\frac{n^2}{\delta})$.

4.4.1 Cardinality of the Constraint Curves

In Step 6 we update the number of intersections for a region. In the complexity analysis, we assert that for a given node and a single configuration space, there are at most $O(n)$ constraint curves which are crossed and that it takes constant time to perform each update. In this section, we discuss the first of these two claims.

A t-circle fixed in orientation and position at one end of the path segment and allowed to vary in either orientation or position at the other defines a straight line in configuration space orthogonal to one of the axes.²⁵ We call this line a **sweepline**. For complexity analysis, we need to know the number of constraint curves which the sweepline can intersect. To do so, we examine each fixed t-circle to determine the number of possible ways the constraint types can

²⁵Once again, we use the convention that the torus is cut open and laid out in the plane such that the horizontal axis corresponds to θ_i and the vertical to θ_f .

be satisfied. Here we show that there are at most $O(n)$ crossings, where n is the number of obstacle edges in the environment.

Type A From the drawing in Figure 6, we can see that each obstacle vertex can generate at most 2 Type A curves for each endpoint. There are 0,1,2, or an infinite number of intersections between the CCC and the circle around the endpoint. We ignore the intersections when there are an infinite number, because the constraint has no significance. Each obstacle edge may also generate at most 2 for each path endpoint. Once again, if the edge generating the constraint is the one for which we are determining the Type A constraints, these can be ignored. For a given sweepline, the constraints which are orthogonal to the sweepline will all be crossed.

Type B There is one curve each generated for the t-line going straight into the IP and FP. For any fixed initial or final angle, there are three possible intersections of the straight line with Type B curves. These correspond to the straight line going into the fixed endpoint at the angle of the sweepline and then to possibly two t-circles at the other endpoint, and the straight line from the non-fixed endpoint going tangent to the fixed t-circle.

For the modified trajectory, there are two t-circles at the other path endpoint which pass through the path endpoint for which the angle is fixed.

Type C Fixing the t-circle at one end of the path segment and forcing the t-line to pass through an obstacle vertex will result in at most two possible t-circles at the other endpoint which can complete the trajectory.

Type C' Intersecting a circle of radius $2R$ centered at the center of the fixed t-circle with the radius R tube around an obstacle edge yields two intersections per obstacle vertex and two intersections per obstacle edge. Intersecting a radius $2R$ circle centered at each one of these intersection points with the CCC of the moving t-circle gives the angles at that side which satisfy the constraint. There are at most 2 intersections per circle.

Type D There is one curve which is generated as shown in Figure 10. There are at most two t-circles at the other path endpoint which are tangent to the fixed t-circle.

Type D' This constraint crosses the sweep line at most twice. The crossings correspond to intersecting a $4R$ circle centered around the center of the t-circle associated with the sweep line with the CCC of the other endpoint of the path.

Type E For a fixed t-circle, there are at most two intersections with the endpoint of a line segment. For each of these two intersections there are two possible t-circles at the other endpoint which pass through this point. Therefore the sweepline may pass through the Type E curve for a given obstacle edge at most four times.

4.5 Transitions between Regions

In this section we discuss a method for determining in constant time the changes in the number of intersections for a region when a constraint curve is crossed. We can perform these updates in constant time because each constraint curve is associated with a small number of edges in the environment. Thus, two regions which share a constraint curve as a boundary contain trajectories which have at most a small difference in the number and type of intersections. The basic idea is to keep a tally for each region of the number of intersections its trajectories make with the obstacles. When this number is non-zero, the region is part of the configuration space obstacle.

The number of intersections for a new region can be determined in constant time from an adjacent region for which the number is known. We must spend $O(n)$ time computing this number only for the first region. The following rules can then be used to compute the desired result for any other region in constant time.

Type A A Type A constraint curve is due to either a single obstacle edge or a single obstacle vertex. In the latter case, all edges

which share this vertex must be checked. First compute the number of intersections of a trajectory in the known region with the obstacle edge(s). Then perform the same calculation for a trajectory in the new region. To get the total number for the unknown region, subtract the number of intersections with the segment(s) in the known region from the total number for that region and add the number for the unknown region.

There is one additional operation which must be performed, only in the case of a Type A constraint. As discussed below, the number of intersections of the t-circles with the obstacle edges must be maintained. This can be done using the fact that this number only changes when a Type A constraint is crossed. We can make the computation of the change in the number at the same time that the entire trajectory is checked. Updating the number is done in exactly the same fashion as above, by adding and subtracting the numbers on either side of the constraint.

Type B In this case, all of the intersections remain the same except for one t-circle which goes from contributing almost no part of the trajectory to contributing almost its entire perimeter. The t-line and the intersections along the other t-circle remain unchanged. The only change here is to add or subtract all of the intersections of the t-circle in question to or from the the total number of intersections for the adjacent region. However, to do so requires keeping track of the total number of intersections of the whole t-circle with all of the obstacle edges. This is updated only when crossing Type A constraints. The method of updating is explained above.

Type C Type C curves are due to a single obstacle vertex. Therefore, as in the Type A case we must check all of the obstacle edges which touch this vertex.

Type C' These curves are due to either an obstacle vertex or an obstacle edge. To calculate the total intersections for an unknown region from that of a known region, add and subtract the appropriate values as for the Type C constraints.

Type D Type D curves are the boundary between the region for which the trajectories exist and those for which they don't.

For the sake of book-keeping, increment the number of intersections when passing into the region for which trajectories do not exist and decrement the number when passing out of this region.

Type E Type E curves occur only within the region for which the trajectories do not exist. They are associated with a specific obstacle edge. The number of intersections of the modified trajectory with this edge can be calculated in the known region and subtracted from the total, and then calculated in the unknown region and added to the total.

4.6 Speed-up to the Algorithm

We need not perform the loop starting at Step 4 $O(\frac{1}{\delta})$ times if there are final orientations which can not be reached and which cause some of the possible links to be missing from the graph. We know that the number of intersections can not change between intersections of the constraint curve with the sweep line. Thus the final orientation counter can be incremented to jump above the next intersection when the current trajectory has intersections with obstacles. Thus, the algorithm is forced to generate only the links which are actually free. If a is the number of actual neighbors of the node, the neighbor generating algorithm takes time $O(n^2 \log n + a)$. Note that a can be $O(\frac{n}{\delta})$ for each of the $O(\frac{n}{\delta})$ nodes, but in general it will not be that large because many of the nodes will not be adjacent.

4.7 Complexity of Path Generation

The complexity of the full search algorithm can be determined using the neighbor generation algorithm discussed in Section 4.6. We assume that the set OPEN is stored in a heap, so that the member with the shortest current path can be found in constant time. The complexity for each step of the algorithm is given below.

1. $O(n^2 \log n + a_{ip})$

2. $O(1)$.
3. $O(1)$.
4. $O(n^2 \log n + a_{cn})$
5. CLOSED: $O(1)$.
UNSEEN: $O(\log \frac{n}{\delta})$.
OPEN: $O(\log \frac{n}{\delta})$.
6. This loop gets performed $O(\frac{n}{\delta})$ times (once per node).

The notation a_{ip} and a_{cn} refers to the number of links for the initial node and the current node, respectively. The total number of links for every node in the graph is A , and is upper bounded by $\frac{n^2}{\delta^2}$. By this analysis, we see that the path generating algorithm will run in time $O(\frac{n^3}{\delta} \log n + A \log(\frac{n}{\delta}))$.

4.8 Closeness of the Approximation

At any point of the algorithm, we will have a free grid point with initial and final orientation within δ of the shortest robust path. In order to determine the distance which the approximate path travels with respect to the shortest path, we need only to consider the difference in path length for a path which differs from another by δ in one orientation. We assume that there is a continuous homotopy between the two paths parameterized by θ_i , *wlog*. Therefore we need not consider multiples of 2π in the orientations.

We assume that $\delta \in [0, \pi]$. The minimum radius of curvature is given by R . The initial position and final position of the path segment are given by IP and FP, respectively. The center of the initial t-circle is denoted tci and the final tcf. Hence we have:

$$\text{tci}(\theta_i) = IP + R \begin{pmatrix} \cos \theta_i \\ \sin \theta_i \end{pmatrix} \quad (4.30)$$

$$\text{tcf}(\theta_f) = FP + R \begin{pmatrix} \cos \theta_f \\ \sin \theta_f \end{pmatrix} \quad (4.31)$$

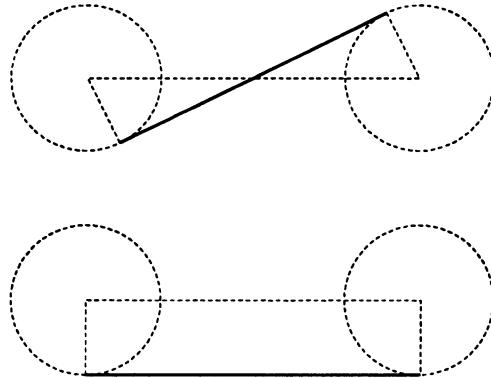


Figure 18: The straight line segment of the two types of paths are shown.

First we bound the difference

$$\|\mathbf{tci}(\theta_i + \delta) - \mathbf{tci}(\theta_i)\| = R\sqrt{2(1 - \cos\delta)} \quad (4.32)$$

using the law of cosines. Furthermore, $1 - \cos\theta$ is monotone increasing over $[0, \pi]$, so Equation 4.32 gives the maximum over the range $[\theta_i, \theta_i + \delta]$.

Using this estimate, we can bound the change in the distance traveled. Figure 18 shows the t-line for the two possible cases of the trajectories. By symmetry, we only need to consider these.

4.8.1 LL and RR Turns between Vertices

In the case that the turn type is LL or RR the length of the straight line path is equal to the distance between the centers of the t-circles. Furthermore, modulo multiples of 2π , the distance traveled along the t-circles is given by $R(\theta_i - \theta_f)$. So for a change of θ_i to $\theta_i + \delta$, the change in path distance, Δp , is given by:

$$\Delta p = R\delta + \|\mathbf{tci}(\theta_i + \delta) - \mathbf{tcf}(\theta_f)\| - \|\mathbf{tci}(\theta_i) - \mathbf{tcf}(\theta_f)\| \quad (4.33)$$

$$\Delta p \leq R\delta + \|\mathbf{tci}(\theta_i + \delta) - \mathbf{tci}(\theta_i)\| \quad (4.34)$$

Substituting Equation 4.32 yields

$$\Delta p \leq R\delta + R\sqrt{2(1 - \cos\delta)} \quad (4.35)$$

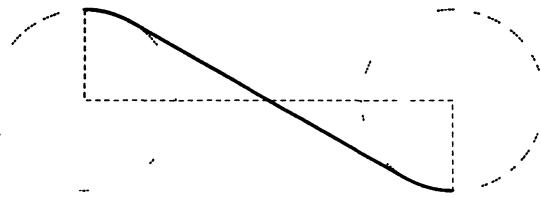


Figure 19: The path used to bound the length of an RL or LR path is the dark line. The two dashed circles are the t-circles. The dashed line connects their centers.

4.8.2 LR and RL Turns between Vertices

It is not as simple to bound the path length of the LR and RL turns, because the expression for the length of the path has a more complex form. Instead of splitting the path length between distance along the straight line and distance along the t-circles, as we did above, we include a portion of the path which is along the t-circles. The path is shown in Figure 19. The arclength, p , of the path is related to the distance $\|\text{tci}(\theta_i) - \text{tcf}(\theta_f)\|$, which we denote d , by

$$\frac{p}{d} = \frac{2R}{d} \sin^{-1}\left(\frac{2R}{d}\right) + \sqrt{1 - \frac{4R^2}{d^2}} \quad (4.36)$$

Since $d \geq 2R$, we have that this function is monotonically decreasing for $\frac{d}{2R} \geq 1$. This can be seen by taking the derivative of this function (where we change variables by using $x = \frac{d}{2R}$).

$$\frac{d}{dx} \left(\frac{1}{x} \sin^{-1}\left(\frac{1}{x}\right) + \sqrt{1 - \frac{1}{x^2}} \right) = -x^2 \sin^{-1}\left(\frac{1}{x}\right) \quad (4.37)$$

Since $x \geq 1$, $\pi \geq \sin^{-1}\left(\frac{1}{x}\right) > 0$, since we are taking the principal value of the arcsin. Hence the derivative is strictly negative, implying that the function takes its maximum at $d = 2R$ where it equals $\frac{\pi}{2}$. Now as the t-circle rotates the path goes from different points along the t-circle. However, we can bound the change of the locations of the endpoints of this path with respect to any fixed point on the non-rotating t-circle and the IP for the rotating t-circle. The angle at which the line perpendicular to the line joining the two t-centers can change by is at most δ . This is because the $\text{tci}(\theta_i)$ and $\text{tci}(\theta_i + \delta)$ must be at least $2R$ from $\text{tcf}(\theta_f)$. Along the rotating

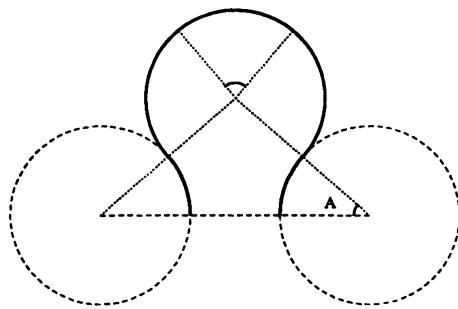


Figure 20: This shows the path used to compute the change in path length of the LRL and RLR paths. The darkened line represents the path.

t-circle the angle to IP also changes by an additional δ . Together, these yield:

$$\Delta p \leq 3R\delta + \frac{\pi}{2}\sqrt{2(1 - \cos\delta)} \quad (4.38)$$

4.8.3 LRL and RLR Turns between Vertices

For either an LRL or RLR trajectory to exist, the centers of the two t-circles must be separated by less than $4R$. When deriving the bounds for these turn types, we assume that the perturbation of the orientation at the endpoints of the path does not cause this condition to be violated. Figure 20 shows the path which we consider to derive the bound. The length of the path shown is $R(4A + \pi)$.

We wish to compute the maximum change in the length of a trajectory path as one t-circle changes by δ . The rolling of the moving t-circle induces a change of at most $R\delta$ in the path. The change in angle of the line connecting the two t-circle centers may change by a large amount, but the net change in path due to this is zero, because the distance which is added along one t-circle is subtracted from the other. So the maximum change, Δp , is given by

$$\Delta p = R\delta + 4R\max\Delta A \quad (4.39)$$

We can express the angle A as

$$A = \cos^{-1} \left(\frac{d}{4R} \right) \quad (4.40)$$

where d is the distance between the two centers of the t-circles. The maximum change in d can be written as $2R\sin(\frac{\delta}{2})$, which is the same as Equation 4.32. We have assumed that

$$\|d - 4R\| \leq 2R\sin\left(\frac{\delta}{2}\right) \quad (4.41)$$

We then have that

$$\max_{x \in [-2R\sin(\frac{\delta}{2}), 2R\sin(\frac{\delta}{2})]} \left\| \cos^{-1} \left(\frac{d}{4R} \right) - \cos^{-1} \left(\frac{d+x}{4R} \right) \right\| \quad (4.42)$$

The maximum occurs when $d = 4R$ and $x = -2R\sin(\frac{\delta}{2})$.

$$\max_{x \in [-2R\sin(\frac{\delta}{2}), 2R\sin(\frac{\delta}{2})]} \left\| \cos^{-1} \left(\frac{d}{4R} \right) - \cos^{-1} \left(\frac{d+x}{4R} \right) \right\| \leq \cos^{-1} \left(1 - \frac{\sin(\frac{\delta}{2})}{2} \right) \quad (4.43)$$

We are finally left with

$$\max_{x \in [-2R\sin(\frac{\delta}{2}), 2R\sin(\frac{\delta}{2})]} \left\| \cos^{-1} \left(\frac{d}{4R} \right) - \cos^{-1} \left(\frac{d+x}{4R} \right) \right\| \leq R \left(\delta + 4\cos^{-1} \left(1 - \frac{\sin(\frac{\delta}{2})}{2} \right) \right) \quad (4.44)$$

If δ is sufficiently small, this bound will be less than the bound given in Equation 4.38.

4.8.4 Paths along walls

Paths are also allowed for which the IP and FP translate along walls, as opposed to rotating about a vertex. It can easily be seen that in the case of LL or RR turns the change in path length is bounded above by ε , where ε is the separation of the IPs for two trajectories which end along a wall. For the case of LR and RL turns, we have an upper bound for the change in path length, Δp . We know that the distance between the center of the t-circles is at least $2R$. Thus, we can determine the change in angle of the perpendicular by looking at the triangle with sides $2R$, $2R$ and ε .

$$\Delta p \leq \frac{\pi}{2}\varepsilon + 2R\cos^{-1} \left(1 - \left(\frac{\varepsilon}{2R} \right)^2 \right) \quad (4.45)$$

The change in path length due to either an LRL or RLR path whose initial position slides along the wall is

$$\Delta p \leq 4R\cos^{-1}\left(1 - \frac{\varepsilon}{2R}\right) \quad (4.46)$$

It is evident from Equations 4.45 and 4.46 that we can choose a uniform spacing, ε , along the walls which will generate $O(\frac{n}{\delta})$ nodes and not increase the bounds derived for trajectories which pass through the vertices.

4.8.5 Deriving a Multiplicative Bound

The bound on the change in path length derived for LR and RL is larger and therefore must be used. Because we want a bound which is independent of the length of the minimum path, it is necessary to express the change in path length as a stretching factor. To do so, requires that we know the shortest path segment which can be part of a globally minimum length path. A path of type LRL or RLR is at least πR in length, if it is a minimal path [Dub57, p. 513]. For paths which travel between two separate vertices, the minimum length path is a straight line connecting the two. For a path which goes to the same vertex, the minimum length path is at least $2\pi R$ long. For a path between two non-intersecting edges, the path length must be larger than the minimum distance separating them. This leaves only the length of a path which passes between two edges which are adjacent. In the case that the corner is not convex, robustness of the path indicates that the path can be pulled free of the wall. This will minimize its length. Thus this case will not occur as part of a minimizing robust trajectory. In the case that the corner is convex, the path must travel at least πR in distance. The minimum length of a path segment, p_{min} , can be given.

$$p_{min} = \min\left(\pi R, \min_{e_i \cap e_j = \emptyset} \text{dist}(e_i, e_j)\right) \quad (4.47)$$

where in Equation 4.47 the second term bounds the distance between any two object edges, e_i, e_j which don't share a vertex. Then we can finally bound the stretch which occurs by the generated path

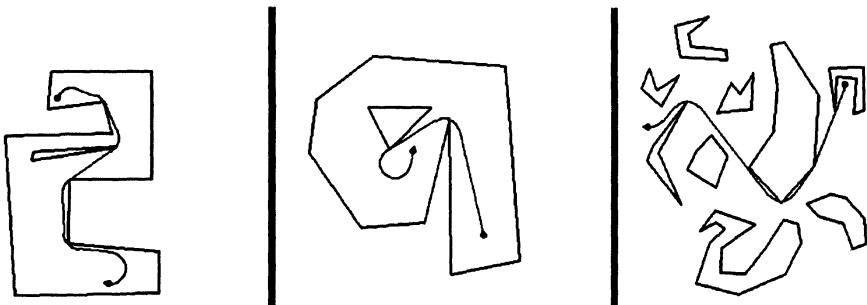


Figure 21: Paths generated by the algorithm.

having orientation off by at most δ from the global minimum path (with length \hat{p}).

$$p_{generated} \leq \left(1 + 2 \frac{3R\delta + \frac{\pi}{2}\sqrt{2(1 - \cos\delta)}}{p_{min}} \right) \hat{p} \quad (4.48)$$

We are guaranteed that this path segment, or one closer to the global minimum will be chosen by the action of Dijkstra's algorithm. Thus the algorithm will generate a path whose length differs from an optimal path by this same factor.

4.9 Implementation

The algorithm described above has been implemented and tested. The implementation consists of roughly 5000 lines of C code running on Sun 3 workstations. Users are able to input the workspace obstacles interactively using a graphical interface under the sun-tools windowing system. A facility exists for entering and saving previously defined environments and calculations in a batch mode.

Paths which were generated using this algorithm are shown in Figure 4.9.

5 A Quadtree Based Algorithm

In the previous section, we never explicitly computed the regions generated by mapping the workspace obstacles into each configuration space. The grid algorithm only required that we find slices of the configuration space obstacle. Here we determine the entire obstacle and use this as a basis for a motion planning algorithm which generates trajectories that are guaranteed to be robust.

In Theorem 3.10, we have specified all of the possible constraint curves which may form the boundary of the obstacle in configuration space. In order to construct the obstacle explicitly, we must determine which parts of these curves actually are boundaries. The pieces which do not contribute to the boundary of the C-space obstacle can be eliminated using a plane sweep algorithm, which we describe in this section. Once we know where the obstacle lies in configuration space, we partition the free regions into quadtree cells. The quadtree data structure [Sam84] is a natural representation of regions in configuration space, because a quadtree cell represents the cartesian product of a range of orientations at each endpoint of the simple path. If the cell is contained in a free region, then all of the final orientations can be reached from any of the initial orientations. If we construct a graph whose nodes represent ranges of orientations at the points on the obstacle boundaries, we can find a collision-free trajectory using a graph search similar to that described in the previous section. Under this construction, the graph search will yield a path which is robust to orientation perturbations. This is in contrast to the solution trajectories generated by the previous grid-based algorithm which were not guaranteed to have this property.

The majority of the section is spent on the details which are needed to perform the plane sweep, since this is the main difference from the algorithm of the previous section. We first consider the form of the constraint curves in the configuration space. As an example, we examine the Type D constraints carefully and show that these generally form smooth curves. We then show that the extrema of the constraint curves with respect to θ_i or θ_f can be represented geometrically, as well as analytically. As we will see, these

extrema are needed for the plane sweep. We give the geometric interpretation of the points in configuration space at which the regions intersected by the sweeline will change. Finally, we discuss the method for updating the number of intersections in a region based upon the regions which are adjacent to it.

5.1 Plane Sweep in Configuration Space

The constraint curves cut out regions in the configuration space for which the number of intersections is constant for all of the trajectories represented by the interior of the region. The parts of the constraint curves which form the obstacle boundary are all those pieces which separate a region with zero intersections from one with non-zero intersections.

It is a standard technique in computational geometry to determine the regions formed by an arrangement of lines. See [Meh84, pp. 147 – 160] for details of a plane sweep algorithm which works on straight lines. It can be easily extended to the case of arbitrary curves provided that certain critical features of the curves can be determined. In order to use the plane sweep algorithm, we must find solutions for the following two tasks:

1. To determine the angles at which the sweeline will intersect a new region bounded by the constraint curves.
2. To determine the number of intersections for a region as it is found during the plane sweep.

The remainder of this section addresses these questions.

One final note is that we are performing this plane sweep on the torus. The sweeping of the sweeline is associated with the sweeping in orientation of one of the t-circles of the simple path. In our discussion, we once again assume that the torus of configuration space is cut open and laid flat on the plane.²⁶ In this case, the sweeline is represented by a straight horizontal or vertical line.

²⁶Clearly, regions may extend across the edges formed when configuration

5.2 Tracking the Regions in C-space

As the sweep line progresses, it will overlap different regions of the configuration space. A data structure will maintain a tree holding the constraint curves which are currently crossed by the sweep line. We will show that we can define critical angles of the sweep at which the ordering of curves along the sweep line may change. In addition, we will see that there is some geometry which is associated with these critical angles. The intervals of the sweep line between intersections with the curves correspond to connected regions of C-space cut out by the constraints. The regions may change their size and shape, but they appear and disappear only at the critical angles of the constraint curves which bound them. At each critical angle the sweep line structure is updated to reflect the changes in the affected regions.

The type of critical angle which is crossed determines the number and type of regions which result. In order to perform the sweep in time proportional to the number of critical angles, there must be a method for determining in constant time which critical angles cause qualitative changes in the boundary of the C-space obstacle. This can be done because each critical angle is associated with a small number of constraint curves, and hence with a small number of edges in the environment. The basic idea of the plane sweep is to keep a tally for each region of the number of intersections its trajectories make with the obstacles. When this number is non-zero, the region is part of the configuration space obstacle.

The number of intersections for a new region can be determined in constant time from an adjacent region for which the number is known. This has been discussed in Section 4.5 on the grid-based algorithm.

space is sliced in this manner. When this happens, the constraint curve that forms its boundary will move from the top of the sweepline to the bottom (or vice versa). This situation can easily be accounted for in the plane sweep.

5.3 The Form of the Constraint Curves

In order to perform the plane sweep, we must know when the sweepline intersects each constraint curve. Although we envision the sweepline moving continuously across the configuration space, in reality it moves by jumping between consecutive angles at which the arrangement of the curves along the sweepline can change. To determine these **critical angles**, we must have a description of the form of the constraint curves.

In this section, we show that it is possible to rigorously analyze the form of the curves in the configuration space which are generated by the constraints of Theorem 3.10. However, we then use the topological description of the curves we derive to then show that there is a simpler geometric representation. This allows us to find the extrema of the curves with respect to θ_i or θ_f without being forced to solve transcendental equations. Instead, we reduce the problem to one of finding the intersections of lines and circles.

We will discuss only the Type D curves in this section. The other constraints can be analyzed in a similar fashion, but are not included. From the calculations which follow, it becomes clear that there are certain situations for which the form of the curves fundamentally changes. By identifying when these changes occur, we can determine the form of the constraint curves in configuration space by plotting a representative curve for each situation. Figure 22 depicts the Type D curves in configuration spaces corresponding to different separations between endpoints of the simple paths (which are vertices in this picture). The squares correspond to the torus (θ_i, θ_f) cut open so that the initial directions are along the horizontal axis and the final directions are along the vertical axis. Opposite sides of the squares are identified.

We begin analyzing the form of the constraint curves using the techniques of differential topology. We will only use some simple results about the level sets of smooth functions. The necessary theorems are stated in the body of the text with references given for the more interested reader.

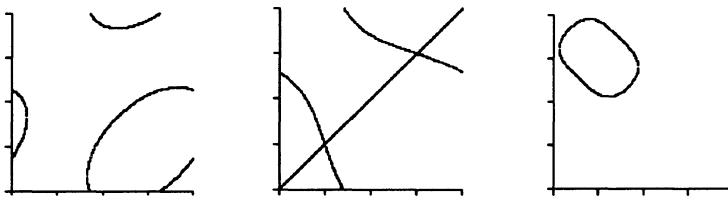


Figure 22: Form of the Type D constraint curves for different values of Δx . From left to right: $0 < |\Delta x| < 2$; $|\Delta x| = 2$; $2 < |\Delta x| < 4$.

Theorem 5.49 *The set of angles for which the two trajectory circles are osculating is a 1-dimensional submanifold of the torus, except in the case that the distance between IP and FP is exactly $2R$ or $\geq 4R$.*

Proof. Without loss of generality, we can set $R=1$, $IP = (0,0)$ and $FP = (\Delta x, 0)$.

The center of the initial t-circle is then

$$\mathbf{tci}(\theta_i) = (\cos \theta_i, \sin \theta_i) \quad (5.50)$$

The center of the final t-circle is then

$$\mathbf{tcf}(\theta_f) = (\Delta x + \cos \theta_f, \sin \theta_f) \quad (5.51)$$

We wish to consider the distance function

$$f(\theta_i, \theta_f) = \|\mathbf{tci}(\theta_i) - \mathbf{tcf}(\theta_f)\|^2 \quad (5.52)$$

We note that, in this case

$$f : S^1 \times S^1 \rightarrow \mathbb{R} \quad (5.53)$$

By the Preimage theorem [GP74, p.21], if y is a regular value of the map $f : S^1 \times S^1 \rightarrow \mathbb{R}$ then the preimage $f^{-1}(y)$ is a sub-manifold of T^2 , with $\dim f^{-1}(y) = \dim T^2 - \dim \mathbb{R} = 1$.

So we need to show that 4 is a regular value of the map f . This is because the set $f^{-1}(4)$ is the set of angles for which the t-circles

are osculating. To check that 4 is a regular value, we check that $\forall(\theta_i, \theta_f)$ such that $f(\theta_i, \theta_f) = 4$, we never have simultaneously

$$\frac{\partial f}{\partial \theta_i} = 0 \quad (5.54)$$

$$\frac{\partial f}{\partial \theta_f} = 0 \quad (5.55)$$

where

$$\frac{\partial f}{\partial \theta_i} = 2\Delta x \sin \theta_i + 2\sin(\theta_i - \theta_f) \quad (5.56)$$

$$\frac{\partial f}{\partial \theta_f} = -2\Delta x \sin \theta_f - 2\sin(\theta_i - \theta_f) \quad (5.57)$$

We now examine the situations in which both partials may equal zero.

$$\Delta x \sin \theta_i + \sin(\theta_i - \theta_f) = 0 \quad (5.58)$$

$$\Delta x \sin \theta_f + \sin(\theta_i - \theta_f) = 0 \quad (5.59)$$

We being by assuming $\Delta x \neq 0$, then Equations 5.58 and 5.59 imply that

$$\sin \theta_i = \sin \theta_f \quad (5.60)$$

and therefore that either

$$1. \theta_i = \theta_f$$

$$2. \theta_i = \pi - \theta_f$$

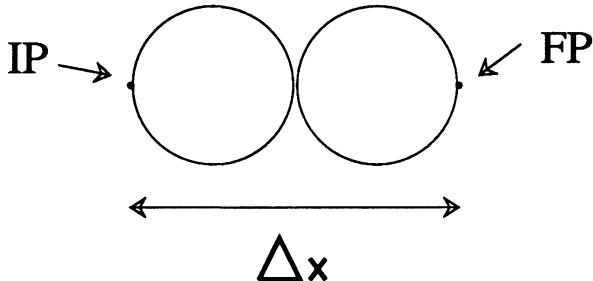
If $\theta_i = \theta_f$, then Equation 5.58 implies that $\theta_i = \theta_f = n\pi$. The constraint that $f = 4$, yields that $\Delta x = \pm 2$. In this case, 4 is not a regular value, and hence the preimage is not a manifold.

Now assume that $\Delta x \neq \pm 2$. If $\theta_f = \pi - \theta_i$, then the distance function gives us

$$4 = f(\theta_i, \pi - \theta_i) = (\Delta x - 2 \cos \theta_i)^2 \quad (5.61)$$

$$\cos \theta_i = \frac{1}{2} \Delta x \pm 1 \quad (5.62)$$

We can derive two results from this,

Figure 23: Critical case for $\Delta x = 4$

1. if $|\Delta x| > 4$, this equation can't be satisfied.
2. if $\Delta x = 4$ then $\theta_i = 0, \theta_f = \pi$. If $\Delta x = -4$ then $\theta_i = \pi, \theta_f = 0$.

In both of these cases, 4 is not a regular value, and hence the preimage is not a one-dimensional manifold. We can see geometrically, that in the former case, there are no angles for which the two circles may touch, and hence the preimage is empty. In the latter case, the analysis agrees with our intuition that the two circles may touch only at one isolated pair of angles. That is when the centers of the t-circles lie on the line segment joining IP and FP, as shown in Figure 23. Now we must consider the cases that $4 > |\Delta x| > 2$. and $2 > |\Delta x| > 0$. Since $\theta_f = \pi - \theta_i$, we can rewrite Equation 5.58 to get Equation 5.63.

$$\sin \theta_i (\Delta x - 2 \cos \theta_i) = 0 \quad (5.63)$$

Substituting Equation 5.62 shows that the second factor is always ± 2 . Thus, Equation 5.63 can be satisfied only when $\sin \theta_i = 0$. We can solve for $\sin \theta_i$, using the expression for $\cos \theta_i$ in Equation 5.62.

$$\sin \theta_i = \pm \sqrt{\pm \Delta x - \frac{1}{4} \Delta x^2} \quad (5.64)$$

Thus, $\sin \theta_i = 0$ when $\Delta x = 0, \pm 4$. Therefore, for $4 > |\Delta x| > 2$, and $2 > |\Delta x| > 0$, the value 4 is a regular value of f .

Finally, we examine the case that $\Delta x = 0$. In this case, 4 is not a regular value of f , but geometrically we can see that $\theta_f = \theta_i + \pi$ is the only possibility for $f(\theta_i, \theta_f) = 4$. This defines a one-dimensional submanifold of T^2 . ■

Remark. We note that in the case that $\Delta x = \pm 2$, we still have a 1-dimensional submanifold at all of the points of $T^2 \setminus \{(\pi, \pi), (0, 0)\}$. If we then add these critical points back in, we have a constraint curve on the torus which does divide the torus into two regions, it just has two points where the constraints cross.

Remark. In the case that $\Delta x = 0$, we note that the only allowable trajectories return the robot to the same location and direction as it started in. Thus, in the case of LR or RL turn types, which are the only cases in which we have Type D constraints, we can consider the configuration space obstacle to be all of T^2 when $\Delta x = 0$. Thus in the planning algorithm, we never compute a configuration space for LR or RL turns which return the robot to the same location.

5.3.1 Geometry of the Type D Curves

In order to find the critical angle of the first or last intersection of the sweepline with the Type D curve, we must know the location of the extrema with respect to θ_i and θ_f . The extrema of the Type D curves occur when $f(\theta_i, \theta_f) = 4$ and one of the Equations 5.58 or 5.59 holds. However, we would like to examine the Type D curves from a geometric point of view. This perspective results in a much simpler characterization of the extrema of the Type D curves. The theorem of this section states that in order to find the angles which correspond to the extrema of the Type D curves, it suffices to intersect a circle of radius $3R$ with center IP (FP) with the CCC around FP (IP). This gives a simple geometric characterization of these extremal points in terms of a quadratic equation describing the intersection of two circles.

Lemma 5.65 Denote the unit vector in the direction θ as $u(\theta)$. The projection of $u(\theta_f)$ on the perpendiculars to $u(\theta_i)$ has length $\pm \sin(\theta_f - \theta_i)$.

Proof. The perpendiculars to $u(\theta_i)$ are given by $u(\theta_i \pm \frac{\pi}{2})$. Because all vectors are unit length, the length of the projection is

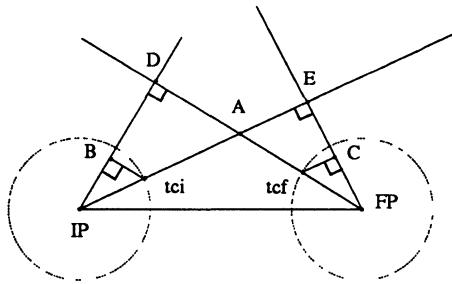


Figure 24: Geometry of Theorem 5.66

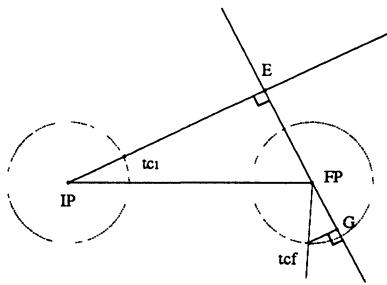


Figure 25: Geometry of Theorem 5.66

given by the dot product of the vectors, which is given by the cosine of the angle between them. Applying the identity $\cos(\theta \pm \frac{\pi}{2}) = \mp \sin(\theta)$ yields the result. ■

Theorem 5.66 Suppose $\|IP - FP\| = \Delta x > 0$ and $\|\text{tci} - \text{tcf}\| = 2$ (ie. $f(\theta_i, \theta_f) = 4$). Then Equation 5.58 is equivalent to $\|IP - \text{tcf}\| = 3R$ and Equation 5.59 is equivalent to $\|FP - \text{tci}\| = 3R$.

Proof. Without loss of generality, we assume $R = 1$.

We make use of Figure 24 to prove the assertion.²⁷ We use the symbol \angle to denote the angle between line segments, which are denoted by their endpoints as labeled in Figure 24. For example $\angle \text{tci} - IP - FP = \theta_i$ and $\angle IP - FP - \text{tcf} = \pi - \theta_f$. The box symbols represent right angles. For example, $\angle C - E - A = \frac{\pi}{2}$.

²⁷Figure 24 is drawn for a general choice of θ_i and θ_f .

Now we have the following relationships.

$$\|IP - FP\| = \Delta x \quad (5.67)$$

$$\|IP - tci\| = 1 \quad (5.68)$$

$$\|FP - tcf\| = 1 \quad (5.69)$$

$$\|IP - D\| = \Delta x \sin \theta_f \quad (5.70)$$

$$\|FP - E\| = \Delta x \sin \theta_i \quad (5.71)$$

The previous lemma implies that

$$\|IP - B\| = \sin(\theta_f - \theta_i) \quad (5.72)$$

$$\|FP - C\| = \sin(\theta_f - \theta_i) \quad (5.73)$$

For the angles, we have these relationships.

$$\angle C - tcf - FP = \angle E - A - FP \quad (5.74)$$

$$\angle IP - tci - B = \angle IP - A - D. \quad (5.75)$$

Equations 5.74 and 5.75 imply that the triangles $A - C - FP$ and $tcf - C - FP$ are similar, as are the triangles $A - D - IP$ and $tci - B - IP$. Now Equation 5.58 demands that $\|FP - C\| = \|FP - E\|$ and Equation 5.59 that $\|IP - B\| = \|IP - D\|$. The similarity of the appropriate triangles yields that $A = tcf$ for Equation 5.58 to hold, and $A = tci$ for Equation 5.59 to hold. Clearly, the result still holds if Figure 24 is flipped across the line connecting IP and FP.

We are left with the case that the ray from IP to tci does not intersect the ray from FP to tcf. In this case the drawing in Figure 24 is not appropriate, because the point labeled A does not exist. This case is shown in Figure 25. Here we have that $\|FP - E\| = \Delta x \sin \theta_i$ and, from the previous lemma, $\|FP - G\| = \sin(\theta_i - \theta_f)$. In this case, we see that tcf must lie on the line passing through IP and tci in order to have $\|FP - E\| = \|FP - G\|$ as required by Equation 5.59. A similar situation, not shown, holds for IP.

Thus, we have that tci, tcf and either IP or FP must be collinear. Combining this with the constraint that $\|tci - tcf\| = 2$ and $\|IP - FP\| \neq 0$ yields the desired result. ■

5.4 Critical Angles of the Plane Sweep

Now that we have examined the Type D constraints closely, we present the geometry of the critical angles associated with the other constraint curves. There are three types of critical angles, the first of which we have mentioned in the discussion above.

1. Extrema of the constraint curves with respect to θ_i or θ_f .
2. Endpoints of the curves.
3. Intersections between constraint curves.

Fortunately, as we have indicated in Theorem 5.66, the extrema of the constraint curves in C-space have geometric significance which can be exploited. In fact, we can find all of the critical angles of the plane sweep using simple geometric operations. In this section we present a number of the situations which generate the critical angles of the plane sweep. The rest are similar, and we leave to the reader to derive the geometry which is associated with these. All can be found in closed form using the intersections of lines and circles.

5.4.1 Critical Angles Due to Extrema

Type AM There is no Type AM corresponding to maxima or minima of the Type A constraints. This is because these constraints are straight lines and hence do not have maxima or minima.

Type BM There are two effects which cause the extrema for the Type B constraints in C-space. These come about in different ways depending upon the separation of the IP and FP. If IP and FP are the same point, then the Type B constraint is a diagonal line with no extrema. When two distinct CCCs intersect, there are up to two t-circles which pass through both the IP and FP. The angles which correspond to these points are critical angles. If the CCCs do not intersect, there exist

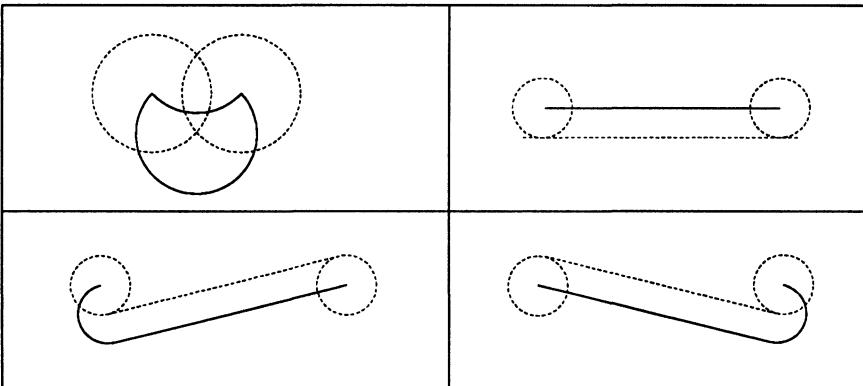


Figure 26: Type BM: The CCCs and LCCs are shown as the dashed circles and lines, respectively. The t-lines and t-circles are solid. The geometry of the maxima and minima, for the case that the CCCs don't intersect, are shown in the lower half of the drawing. In the upper left is the case when the two CCCs intersect. In the upper right of the drawing is the Type BB constraint. Note that the LCC is tangent to the CCCs.

two bitangents to the CCCs which cross. These particular two bitangents correspond to the maxima and minima of the Type B constraints where they don't intersect. The Type BB critical angle also defines points of maxima and minima for the two curves taken separately. It corresponds to one of the bitangents of the CCCS which does not cross. See Figure 26.

Type CM Maxima or minima for the Type C constraint curve correspond to the t-line passing through the obstacle vertex, while the appropriate LCC is tangent to a CCC. The Type CM critical angles contain the Type BC critical angles, which is evident by comparing the descriptions of the BC and CM Types.

5.4.2 Critical Angles due to Curve Endpoints

Type CE The only curves which have endpoints are due to the Type C constraints. This is because the Type C constraints

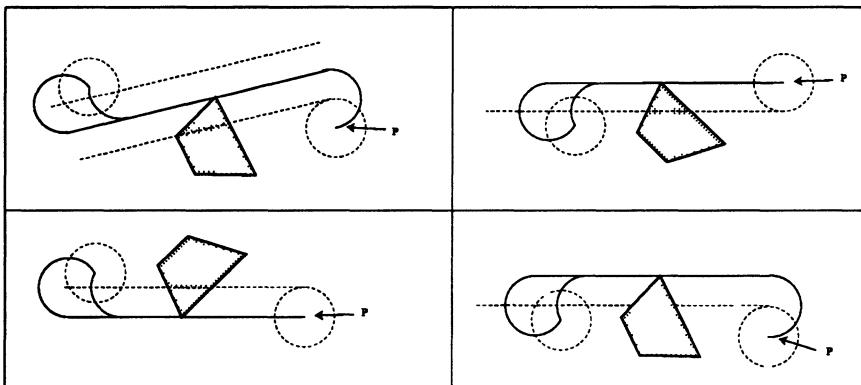


Figure 27: Type CM: The LCCs, CCCs, t-lines and t-circles are shown in the usual fashion. In some cases, both LCCs must be used. In others, only one is important. Contrast these pictures with the one illustrating the Type BC critical angles.

arise from intersections of the t-line with a obstacle vertex. However, it is often the case that the point of intersection moves from the t-line to the t-circle as the initial and final angles are changed. Thus the constraint changes from a Type C constraint to a Type A constraint. The angles for which the obstacle vertex lies at the point of tangency of the t-circle and the t-line corresponds to the endpoint of the Type C constraint curve in C-space. These points are determined by looking at the Type A constraints for each endpoint for which the t-circle intersects an obstacle vertex. The point of intersection of the t-circle and vertex determines the t-line. The appropriate LCC can then be intersected with the CCC of the other endpoint to determine the angles. See Figure 28 for the geometry involved.

5.4.3 Critical Angles Due to Intersections of Constraints

Type AA The intersections of the Type A constraints for each endpoint can be found simply by pairing up the appropriate angles. There are no additional calculations which need to be made.

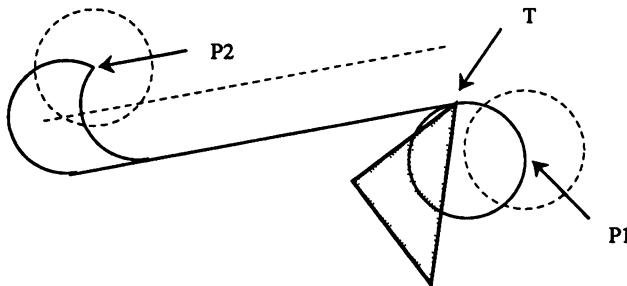


Figure 28: Type CE: The CCCs and LCCs are shown as the dashed circles and lines, respectively. The t-lines and t-circles are solid. The t-circle through P1 intersects the obstacle vertex at point T. This determines the t-line. The LCC is intersected with the CCC of point P2 to determine the appropriate angles.

Type AB Each A constraint essentially fixes a t-circle at an end of the trajectory. Thus an intersection with a Type B constraint can be found for this specific t-circle. There are two possibilities for an intersection here.

1. The endpoint causing the A constraint is the one which has the t-line going directly into it. The A constraint, along with the turn type fixes the LCC which is then intersected with the CCC of the other endpoint to determine the centers of the t-circles on that end.
2. The endpoint causing the A constraint is the one for which the trajectory is curved initially. Given the type of turn involved, one can then determine the angle at the other endpoint.

Type AC Once again the A constraint fixes a t-circle. The vertex in question and the type of turn determines the t-line which then determines the angles at the other end of the trajectory by intersecting the appropriate LCC with the CCC.

Type BB This constraint only occurs when both endpoints have Type A constraints. Otherwise, one of them won't have a Type B constraint. Intersections for these points occur at the angle corresponding to the direction taken when moving along a straight line connecting the IP and FP.

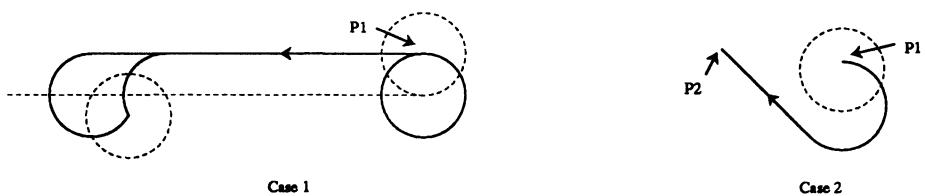


Figure 29: Type AB: The CCC is shown as a dashed circle for the point P1. The solid circle passing through P1 is the t-circle which is associated with the Type A constraint (the obstacle causing this constraint is not shown). The solid line is the t-line which in case 1 goes directly into P1 and in case 2 goes directly into P2.

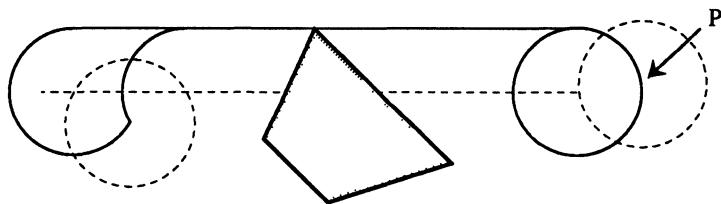


Figure 30: Type AC: The CCC is shown as a dashed circle for the point P. P is the endpoint contributing the Type A constraint and the solid circle passing through P is the t-circle associated with the A constraint (the obstacle edge which is causing the constraint is not shown, however). The other solid circles represent the t-circles which connect to the t-line at the other endpoint, thus determining the final angles of travel at that point.

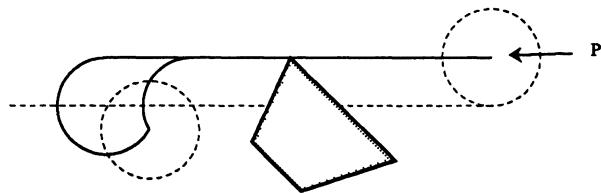


Figure 31: Type BC: The CCC is shown as a dashed circle for the point P. P is the point contributing the Type B constraint.

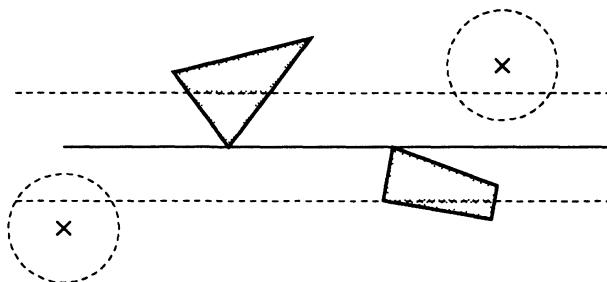


Figure 32: Type CC: The CCC is shown as a dashed circle for the point P. The t-line passing through the two vertices is shown as a solid line. The two LCCs are shown as dashed lines. The t-circles are not shown.

Type BC The t-line can be determined by the endpoint and the vertex through which it travels. From this, the LCC can be found which corresponds to the particular turn type. By intersecting this with the CCC of the other endpoint, the angles can be determined.

Type CC The two vertices of the polygons determine the t-line. From this the appropriate LCC(s) for the turn type can be found which is(are) intersected with the CCC's of both endpoints to find the angles.

In all of the above cases, the particular kind of critical angle may not exist. In all cases, the conditions under which the critical angles exist are clear.

5.5 Complexity of the Plane Sweep Algorithm

By the enumeration of critical angles, we can see that there are $O(n^2)$ of them. Because we must sort them, the plane sweep algorithm takes $O(n^2 \log n)$, where n is the number of corners in the environment, for each such piece of configuration space. All told generation of the $O(n^2)$ pieces of configuration space then takes time $O(n^4 \log n)$.

5.6 Graph Search to Generate Paths

The path represented by the center of each free quadtree block is analogous to a grid point of the previous algorithm. We can thus perform a search using Dijkstra's algorithm in the same manner as before. Since there are again at most $O(\frac{n^2}{\delta^2})$ quadtree blocks, this is the complexity of the search, disregarding the cost of generating the quadtree blocks. This can be done during the plane sweep at a cost proportional to the number of blocks. Therefore, the search requires time $O(n^4 \log n + \frac{n^2}{\delta^2})$. The same approximation bounds derived in Section 4 apply here.

6 Alternate Formulation

We can also give the conditions of the problem statement in a state space form. This clearly points out that the constraints are non-holonomic. The effect of the non-holonomic constraint is to force us to use a configuration space of tuples (x, y, θ) , when it can be seen that θ is actually determined by the ratio of the derivatives of x and y . If we express the non-holonomic joint first in terms of the Pfaffian

$$dx \tan \theta - dy = 0 \quad (6.76)$$

there is no constraint on the minimum radius of curvature of the path. This simply expresses the kinematics of "steering" (see figure 33.).

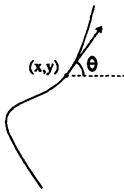


Figure 33: Kinematics of Steering

In state space form we can write out the conditions as follows:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos \theta \\ \sin \theta \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{1}{R} \end{bmatrix} u(t) \quad (6.77)$$

where $u(t) \in \mathbb{R}$ is the steering control. The constraint on the curvature of the path can now be expressed.

$$|u(t)| \leq 1, \quad \forall t \in [t_i, t_f] \quad (6.78)$$

$$(x(t), y(t)) \in \mathbb{R}^2 \setminus \text{int}(\Omega), \quad \forall t \in [t_i, t_f] \quad (6.79)$$

Because the path is arc-length parametrized, we can consider the minimal length path problem to be an optimization problem of determining the control $u(t)$ which minimizes the time required to steer from the initial state (IP, θ_i) to the desired final state (FP, θ_f) subject to the constraint 6.78. Theorem 2.5 states that the control is bang-bang.

$$u(t) \in \{-1, 0, +1\}, \quad \forall t \in [t_i, t_f] \quad (6.80)$$

The corollary adds that the control remains bang-bang even with constraints on the state (the obstacles) which are linear. This suggests that the idea of determining canonical trajectories can be generalized to other systems with non-holonomic joints.

7 Further Areas of Research

Various problems and extensions to this work are under preparation or work is in progress. The immediate focus of our attention is on the following list, which is by no means complete.

Different Robot Models This involves extending the algorithm described in this paper to plan paths for a robot which is larger than a point. We have solved one such case. A paper describing the extension of this algorithm to the case of a robot which is described as a disk is currently under preparation.

Planning Optimal Paths Dividing the configuration space into free quadtree blocks will, in general, cause an optimal path to be overlooked. The quadtree and grid based algorithms can only approximate the free space region. However, eliminating the quadtree partitioning will necessitate a completely different approach to searching free space.

Planning Robust Paths Because of the use of quadtree partitioning of the free space, the orientation of the robot at the via points may be in a neighborhood of the nominal orientation specified for the path generated by the algorithm without causing a collision with an obstacle. This suggests that if robustness with respect to perturbation of the location of the via points can also be achieved, then the path will be very robust to errors in controlling the robot to follow a specified trajectory. There are two possible approaches to take when dealing with robustness issues: how robust is a given path which has been generated and can this measure be used to guide the search; and given a measure of robustness, can a satisfactory path be generated.

Planning C^r Paths, $r > 1$ Practical systems will not be able to track a path which requires instantaneous changes in acceleration. The algorithm presented in this paper relies very heavily on the geometry of the class of trajectories considered. However, it may be possible to generalize to other types of trajectories without sacrificing complexity.

Many open problems remain to be solved which fall under the category of planning motion subject to non-holonomic constraints.

Acknowledgments

The authors would like to thank Professor Shankar Sastry, Greg Heinzinger, Jeff Mason, Niklas Nordstrom, Saman Behtash, and Richard Murray for their helpful discussions and aid in producing this paper.

References

- [AHU83] Alfred Aho, John Hopcroft, and Jeffrey Ullman. *Data Structures and Algorithms*. Addison-Wesley, Reading MA, 1983.
- [Dub57] L. E. Dubins. On curves of minimal length with a constraint on average curvature and with prescribed initial and terminal positions and tangents. *American Journal of Mathematics*, 79:497–516, 1957.
- [FW88] Steven Fortune and Gordon Wilfong. Planning constrained motion. In *STOCS*, pages 445–459, Chicago IL, May 1988. Association for Computing Machinery.
- [GP74] Victor Guillemin and Alan Pollack. *Differential Topology*. Prentice-Hall, 1974.
- [Meh84] Kurt Mehlhorn. *Data Structures and Algorithms 3: Multi-dimensional Searching and Computational Geometry*. Springer-Verlag, Berlin, 1984.
- [Sam84] Hanan Samet. The quadtree and related hierarchical data structures. *ACM Computing Surveys*, 16(2):187–260, June 1984.
- [Win84] Patrick Henry Winston. *Artificial Intelligence*. Addison Wesley, second edition edition, 1984.

9

NONHOLONOMIC CONTROL AND GAUGE THEORY

RICHARD MONTGOMERY
MATHEMATICS DEPARTMENT
UNIVERSITY OF CALIFORNIA SANTA CRUZ
SANTA CRUZ CA, 95064

Phone: (408) 459-4841 FAX: (408) 459-3260
email:rmont@ucscd@ucsc.edu or rmont@cats.ucsc.edu

April 9, 1992

Abstract

We present a dictionary between gauge theory and control theory. This is useful for problems involving the control of the orientations of deformable bodies (robots, gymnasts) by means of shape deformations. In the last section we present some ideas on the stabilization of nonholonomic control systems, where the objective is a given submanifold instead of a single point.

Acknowledgement : I would like to thank R. Murray for a useful discussion regarding feedback stabilization, M. Kawai for telling me about Coron's result, and M. Enos for correcting an error concerning U-joints.

1 The setting

We will be discussing control laws:

$$\dot{q} = h(q)u \quad (1)$$

linear in the control u . Thus $h(q)$ is a linear operator depending smoothly on the state q . We will be especially interested in cases where q splits locally as

$$q = (x, g) \quad (2)$$

so that the control law has the form

$$\dot{x} = u; \dot{g} = -A(x, g)u. \quad (3)$$

Moreover g will take values in a Lie group G , usually the rotation group.

Remark 1 Any control law of the form (1) can be reduced to the form (3) by a smooth feedback transformation $u \mapsto \alpha(x)u$, linear in the controls: provided that

- a) there are fewer controls than states.
- b) $h(q)$ has maximal rank.

The class of examples we have in mind concerns the attitude control of a deformable body in free fall. The problem of a falling cat righting itself or of a satellite reorienting itself by means of rotors are examples. For these examples x coordinatizes the body's shape and g coordinatizes its orientation relative to a fixed inertial frame. Thus g takes values in the rotation group G of (two or three dimensional) space. $A(x, g)$ is then a matrix which transforms according to

$$A(x, gg_1) = gA(x, g_1) \quad (4)$$

Control law (3) can be rewritten in terms of matrix-valued differential forms:

$$dg + A(x, g)dx = 0 \quad (5)$$

In our class of examples eq.(5) is a rewrite of the statement "angular momentum equals zero".

We are deformable bodies! Imagine that we are in freefall with zero total angular momentum. Our problem is to reorient ourselves, say right-side-up, by changing our shape. Such a problem is faced by gymnasts, falling cats, and unfortunate robots. Our control variables are the deformations dx of our shape. These are affected in turn by exerting torques on joints and extending or retracting limbs and we will henceforth ignore the dynamical problem of implementing the dx 's. Our objective is to control g .

A spatial rotation $g \in G$ acts on us by rotation:

$$q \longmapsto gq = q', \quad (6)$$

q being our configuration before rotation and q' our configuration after rotation. The space of all q 's forms our configuration space Q . Two shapes are the same if they differ by a rotation. Thus the *shape space* S is the quotient space:

$$S = Q/G$$

Let

$$\pi : Q \rightarrow S$$

denote the map which assigns to each configuration q its shape $x = \pi(q)$. We say that G acts freely if whenever $gq = q$ we have that $g = e$ where e denotes the identity of the group. In this case S is a smooth manifold and π gives Q the structure of a principal G -bundle.

Definition 1 $\pi : Q \rightarrow S$ is a principal G -bundle if there is a covering of shape space S by open sets $U \subset S$ together with a family of diffeomorphisms (smooth maps with smooth inverses)

$$\phi_U : U \times G \rightarrow \pi^{-1}(U) \subset Q,$$

$$\phi_U : (x, g) \longmapsto q$$

called "local trivializations" with the property that whenever $q = \phi_U(x, g)$ undergoes the rotation $q \rightarrow g_1 q$ then according to the local trivialization ϕ_U we have

$$(x, g) \longmapsto (x, g_1 g)$$

Robot coordinates provide an example of a local trivialization. If there are no constraints on the joints then they actually afford a global trivialization, that is , we can take $U = S$ so that $Q = S \times G$. Consider a robot made of rigid bodies $B_\alpha, \alpha = 1, \dots, N$ attached to each other by joints j_i . Abstractly we can think of the robot as a graph with edges B_α and vertices j_i . The shape $x \in S$ is specified by listing the relative joint angles $X_{\alpha\beta} \in G = SO(d)$, where $d = 2$ for planar robots and 3 for spatial ones. If in addition to these joint angles we specify the center of mass c of the entire configuration, or of one of the bodies, and the orientation $g \in G$ of a single one of the bodies **relative to an inertial frame** then we have completely specified the robot's configuration q . If the body is in freefall then $c = c(t)$ is determined beyond our control. Consequently we ignore the center of mass coordinate c , except for the time constraint that it gives us: whatever we decide to do about our orientation we must do before we hit the ground! The global trivialization is then $q \rightarrow (x, g) = (X_{\alpha\beta}, g)$.

Remark 2 *Why should we worry about bundles when the configuration space Q is isomorphic to $S \times G$? There are two reasons. The first is that this isomorphism is not canonical. It involved singling out one of the bodies in order to compare its orientation with an inertial frame. Thus we have broken some symmetry (gauge invariance) in the problem and ignoring this symmetry can be detrimental. The other reason is that typically there are numerous and various constraints on the $X_{\alpha\beta}$ according to the incidence relations of the graph, the fact that solid objects cannot pass through each other, and the type of joints which join them. These constraints can lead to nontrivial bundles, in spite of the fact that without the constraints the bundle is always trivial.*

The constraint “angular momentum equals zero” can be written

$$I(x, g)dg + M(x, g)dx = 0 \quad (7)$$

$I(x, g)$ is the locked inertia tensor. This is the moment of inertia tensor of the robot when all of its joints are locked in the shape x and orientation g . $M(x, g)dx$ is the total angular momentum which would result from deforming the joints from x to $x + dx$ without changing g . Our control law is thus of the form of eq. 5 with

$$A(x, g) = I(x, g)^{-1}M(x, g) \quad (8)$$

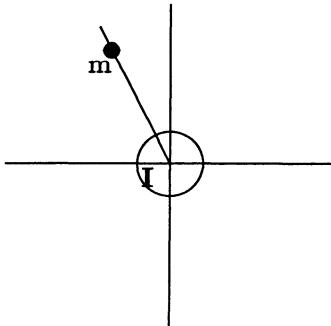
(I is invertible if the action is free.) This last formula is called the ‘master formula’ or ‘master gauge’ by Shapere and Wilczek , [25] , [27]. The reader may check that it satisfies the transformation law of eq. (4).

Remark 3 Historical Remark *Formulas 7 , 8 and the fact that they define a connection can be found in Guichardet’s paper ”On Molecular Dynamics” [14]. They were rediscovered and further exploited by Shapere and Wilczek [25] [26] [27]. See also Montgomery [20], [21].*

Example 1 Heisenberg’s Flywheel Consider a point mass m connected by a massless rod to a flywheel with moment of inertia I . The flywheel is in turn attached to a table by a joint on which the wheel spins freely. The joint is frictionless so that exerts no torque on the assembly. Then the total angular momentum is zero.

$$Id\theta + m(xdy - ydx) = 0 \quad (9)$$

by conservation, assuming it is initially zero. Here θ is the angle of the flywheel relative to the table and (x, y) are the mass’s coordinates, both measured with respect to coordinate axes laid out the table. See figure 1, below.



We can exert a torque on the rod to rotate it relative to the wheel and we can slide the mass back and forth on the rod. Thus we have two controls, the torque τ and sliding speed s , and three states (x, y, θ) . Eq. (

9) encodes the control law. A linear transformation of the form

$$\begin{pmatrix} u \\ v \end{pmatrix} = \mathbf{J}(x, y) \begin{pmatrix} \tau \\ s \end{pmatrix},$$

where J is a 2×2 matrix depending algebraically on x and y , will transform the control laws into the form

$$\begin{aligned} \dot{x} &= u \\ \dot{y} &= v \\ \dot{\theta} &= -\alpha(xv - yu) \end{aligned}$$

where $\alpha = \frac{-1}{m}$. This is the model equation investigated by Brockett in [6], [5].

Suppose we want to optimize the length

$$\int \sqrt{u^2 + v^2} dt$$

of the xy path among all paths with fixed endpoints $q_0 = (x_0, y_0, \theta_0)$ and $q_1 = (x_1, y_1, \theta_1)$. Then the optimal path projects to an arc of a circle in the xy plane. The end point conditions on θ determines the radius of the circle.

$$\theta_0 - \theta_1 = -\alpha \int_{\gamma} (xdy - ydx) = -2\alpha \text{Area} \quad (10)$$

where Area is the signed area of the sector formed by the arc together with the radial segments through its endpoints.

To recast the problem in terms of shape variables, put polar coordinates (r, ϕ) on the xy plane. Then eq. (9) reads

$$Id\theta + mr^2 d\phi = 0 \quad (11)$$

Shape space coordinates are r and $\psi = \phi - \theta$. Then:

$$d\theta = \frac{-mr^2}{I + mr^2} d\psi \quad (12)$$

Remark 4 History We have called this system "The Heisenberg flywheel" because it is isomorphic to a canonical control system on the Heisenberg group H^3 a special three-dimensional nilpotent Lie group. Write:

$$Q = \frac{\partial}{\partial x} + \alpha x \frac{\partial}{\partial \theta}$$

$$P = \frac{\partial}{\partial y} - \alpha y \frac{\delta}{\delta \theta}$$

$$Z = \frac{\partial}{\partial \theta}$$

Then

$$[Q, P] = \alpha Z \quad (13)$$

with all other Lie brackets zero. (13) is the Heisenberg commutation relation, with $\alpha = \frac{1}{\hbar}$. The control law is $\dot{q} = u(t) Q(q) + v(t) P(q)$, $q \in H^3$.

This model system has been extensively studied by Brockett, [6], [5].

This optimal control problem goes back to mythological times. Its solution is attributed to Dido. See L.C. Young p. 215, [32]. Actually Dido solved the dual problem: for fixed length maximize the area enclosed by the arc together with the straight line segment joining its endpoints. The solutions to this problem are the same.

2 The Dictionary

We will now present the dictionary. The reader who does not know the theory of principal bundles may need to refer to § 3. The last entry is described in § 4. For additional details see [20], [21]

<i>Control</i>	<i>Gauge Theory</i>	<i>Deformable Body</i>
control law $\dot{q} = h(q)u$	equations of parallel transport $dq + A(x, g)dx = 0$	total angular momentum is zero
state space	total space Q	configuration space
variables g transverse to contr.	fiber $\pi^{-1}(s)$	all rotations of a given shape s
	structure group G	group of rigid rotations
	Lie algebra of G	space of angular velocities
directly controlled variables x	base space S	shape space
	bundle projection π	to each config. q assigns its shape $x = \pi(q)$
	local trivialization or choice of gauge	robot coordinates are one example
controls u	tangent vectors to base space	shape deformations
$u(t)$ steers from q_0 to q_1	q_1 is the parallel translate of q_0 along $x(t)$ where $\dot{x} = u(t)$	
	the map $q_0 \mapsto q_1$ is called the holonomy when $x_0 = x_1$	$g_0 \mapsto g_1$ is the reorientation of the body
Chow's control. criterion	Ambrose-Singer Thm.	What reorientations are possible?
Lie bracket conditions on contr. vector field	curvature conditions	
optimal control for a quadratic cost function	motions of a charged particle in the master Yang-Mills field	most efficient shape deformat. yield. a desired reorient.

3 Connection on Principal Bundles

3.1 Goals and References

The goal of this section is to summarize the theory of a principal bundle with connection and to provide some details of how to get from one column of the dictionary to the other.

For a more detailed treatment of bundles see Steenrod [29]. For a treatment of connections see Bleecker [3], Chern [8], especially the appendix , or Spivak [28] .

3.2 Bundles

As we have said already, the map $\pi : Q \rightarrow S$ which assigns to each configuration its shape is an example of a principal G bundle, provided the G action is free. In definition 1 above we defined principal G bundles and local trivializations.

S is called the base space, Q the total space and the sets $\pi^{-1}(s) \subset Q$ are called the fibers. Thus Q is the union over S of its fibers. Each fiber is diffeomorphic to G , but not in a canonical way. We should think of the fibers as "affine groups", groups G with no preferred identity. This ambiguity in choice of identity is the essence of gauge theory.

A smooth choice of identity $\sigma(s) \in \pi^{-1}(s)$ for each s in some neighborhood is called a 'local section'. Thus a local section is a map

$$\sigma : U \subset S \rightarrow Q$$

satisfying

$$\pi(\sigma(s)) = s$$

for all x in U . Local sections define local trivializations (definition 1 above) according to the rule

$$\phi_U(x, g) = g\sigma(s)$$

and this defines a one-to-one correspondence between local sections and local trivializations.

In the case of the deformable body the fiber over a shape consists of all configurations having this shape. The ambiguity is that of how to realize a given shape by actually embedding it in space. A local section is thus a choice of reference configuration for each shape in some neighborhood U of shapes.

Remark 5 *In the physics literature the choice of section is often referred to as the choice of gauge. The gauge invariance of a property is then akin to coordinate invariance. A statement or property is called gauge invariant if its validity is independent of the particular choice of local section used to perform a calculation.*

In Kaluza-Klein theories of elementary particle physics S is space-time and G is a space of internal variables attached by π to each point of space-time.

There are topological obstructions to finding an isomorphism $Q \cong S \times G$. The simplest of these are the Chern classes. See [8].

3.3 Connections

The following more algebraic point of view regarding principal bundles is useful in defining connections. Given any point q in Q , let $R_q(g) = qg$ (“right multiplication by q ”). Then we have a sequence of maps

$$G \xrightarrow{R_q} Q \xrightarrow{\pi} S \quad (14)$$

of smooth maps between manifolds. We will call this the “bundle sequence”. It is an exact sequence of smooth maps in the following sense. The map R_q is one-to-one, the map π is onto, and $\pi^{-1}(s) = \text{image}(R_q)$ where $\pi(q) = s$. A local section can be thought of as a local splitting of this exact sequence.

A connection is a family of infinitesimal splittings of the bundle sequence (14). The infinitesimal version of this sequence is obtained by taking dif-

ferentials of the maps involved:

$$\text{Lie}(G) \xrightarrow{\alpha_q} T_q Q \xrightarrow{d\pi_q} T_s S \quad (15)$$

Here we have identified the Lie algebra $\text{Lie}(G)$ with $T_e G$ where e is the identity element of the group G , and α_q is the differential of the map $R_q : g \rightarrow gq$ at e . α_q is sometimes called the ‘infinitesimal action’ of G . In the case of deformable body in three-space, $G = SO(3)$, and $\text{Lie}(G)$ is the space of infinitesimal rotations. A vector $\omega \in \text{Lie}(G)$ represents an instantaneous angular velocity. Thus $\text{Lie}(G)$ can be identified with \mathbb{R}^3 . The infinitesimal action $\alpha_q(\omega)$ is the infinitesimal rotation of the configuration q about the axis ω . In symbols $\alpha_q(\omega)(X) = \omega \times \mathbf{q}(X)$. Here the X ’s serve to label the points of the deformable body, so that $\mathbf{q}(X) \in \mathbb{R}^3$ is the inertial position of the body point labelled X when the body is in the configuration q . (Please ignore the difference between bold face and plain q ’s and X ’s here. They are all vectors.)

Sequence (15) is an exact sequence of linear maps: α_q is one-to-one, $d\pi_q$ is onto, and $\ker(d\pi_q) = \text{image}(\alpha_q)$. This can be seen can be obtained by differentiating the condition of exactness of the bundle sequence (14).

Definition 2 *The image of α_q is called the ‘vertical subspace’ or simply “vertical space” at q . It represents the space of all rigid deformations of q . It is denoted by V_q .*

We have $V_q = \text{im}(\alpha_q) = \ker(d\pi_q) = T_q(Gq) = T_q(\pi^{-1}(s))$. Note that α_q provides a canonical isomorphism between $\text{Lie}(G)$ and V_q . The union of all the V_q ’s is called the ‘vertical distribution’, denoted $V \subset TQ$.

We now give four equivalent definitions of connections.

Definition 3 *A horizontal distribution (sometimes called an Ehresmann connection) is a smoothly varying family*

$$D_q \subset T_q Q$$

of linear subspaces complementary to the vertical distribution and invariant under the G action. Thus

$$T_q Q = V_q \oplus D_q$$

and

$$D_{gq} = gD_q$$

Note that $d\pi_q$ restricted to D_q is a linear isomorphism.

$T_q Q$ is linearly isomorphic to the direct sum $T_s S \oplus \text{Lie}(G)$. (This follows from linear algebra and the fact that the infinitesimal bundle sequence, diagram (15) is exact.) However there is, in general, no canonically defined splitting.

Definition 4 A connection is an equivariant splitting of the infinitesimal bundle sequence (15). Said in more detail, a connection is a family of linear isomorphisms

$$l_q : T_s S \oplus \text{Lie}(G) \rightarrow T_q Q$$

depending smoothly on $q \in Q$ and satisfying the following properties:

$$d\pi_q(l_q(v, \omega)) = v$$

$$\alpha_q \omega = l_q(0, \omega)$$

(the splitting properties) and

$$l_{gq}(v, g \cdot \omega) = g \cdot l_q(v, \omega)$$

(the equivariance property).

In these formulas $v \in T_s S$. In the last formula $g \cdot \omega$ denotes the adjoint action of $g \in G$ on ω . (Again, in the deformable body case is the standard action of $SO(3)$ on \mathbb{R}^3 .) And on the right hand side of this equation $g \cdot l_q(v, \omega)$ denotes, by abuse of notation, the differential of the map $q \mapsto gq$ applied to the vector $l_q(v, \omega) \in T_q Q$.

Definition 5 A horizontal lift is a smoothly varying family of maps

$$h(q) : T_{\pi(q)} S \rightarrow T_q Q$$

such that

$$d\pi_q \circ h(q) = \text{identity on } T_q Q$$

$$h(gq) = gh(q)$$

Definition 6 A connection one-form is a smoothly varying family

$$\Gamma_q : T_q Q \rightarrow \text{Lie}(G)$$

such that

$$\Gamma_q(\alpha_q(\omega)) = \omega$$

$$\Gamma_{gq} = g\Gamma_q g^{-1}$$

(The first g of $g\Gamma_q g^{-1}$ denotes the adjoint action of g on $\text{Lie}(G)$. The second g^{-1} represents the action of G on Q .)

These four definitions are equivalent: an object satisfying any one definition canonically defines objects satisfying all others. They are all related by linear algebra. Thus:

$$D_q = \text{im } h(q) = \text{Ker } \Gamma_q = l_q(T_{\pi(q)} S \oplus O).$$

The reader can work out other relations. For example:

$$l_q = h(q) \oplus \alpha_q$$

By abuse of language, an object satisfying any one of these properties may be called a connection.

In terms of a local trivialization over U the connection one-form Γ must have the form

$$\Gamma(x, g) = (dg + gA(x)dx)g^{-1} \quad (16)$$

Γ is uniquely determined (over U) by the $\text{Lie}(G)$ - valued one-form $A(x) \cdot dx$ on U . $A(x) = A(x, e)$ of eq. (3), (5).

Remark 6 In this last formula we expressed the adjoint action of G on $\text{Lie } G$ as $\omega \mapsto g\omega g^{-1}$ instead of the earlier $\omega \mapsto g\omega$. Here $\omega = A(x)dx$.

Warning Most mathematical texts use right principal bundles as opposed to our left bundles. That is, they write the action of the group as $(q, g) \mapsto qg$. This leads to several sign differences when formulas are compared to ours.

There are situations in which Q has a natural connection. One is when it has a Riemannian metric for which the G -action is by isometries. Then declare

$$\mathcal{D}_q = V_q^\perp$$

where $V_q = \text{im } \alpha_q$, and \perp is with respect to the metric on $T_q Q$. Note that V_q is the tangent space to the G orbit through q , which is also the fiber through q . This is the situation with our class of examples. Take for the Riemannian metric the metric defined by the Kinetic energy of the deformable body.

3.4 Angular Momentum and Riemannian Submersions

There is a nice situation in which Q has a canonical connection and our deformable body examples fits this situation. Suppose Q is a Riemannian manifold and that G acts on Q by isometries. The vertical space V_q is as before: it is the tangent space to the orbit through q . We define the horizontal distribution to be:

$$\mathcal{D}_q = V_q^\perp \tag{17}$$

the orthogonal complement to the vertical space. The invariance of \mathcal{D} under the action of G follows immediately from the fact that G acts by isometries.

If G acts freely then equation 11 defines a connection on the principal bundle $G \rightarrow Q \rightarrow S = Q/G$. Moreover S inherits a Riemannian metric from Q by declaring that, for each q , The restriction of $d\pi_q$ is an isometry

$$\mathcal{D}_q \rightarrow T_s S, \quad s = \pi(q).$$

(Exercise: Show the resulting inner product on $T_s S$ is independent of the choice of $q \in \pi^{-1}(s)$.)

This gives $Q \rightarrow S$ then structure of a "Riemannian submersion."

Definition 7 A submersion $\pi : Q \rightarrow S$ is called Riemannian if Q and S have Riemannian metrics such that $d\pi_q$ restricted to $\mathcal{D}_q = \ker(d\pi_q)^\perp$ is an isometry for each $q \in Q$.

In the case of a deformable body there is a canonical Riemannian metric defined by the kinetic energy:

$$\langle \delta q_1, \delta q_2 \rangle = \int_{X \in B} \langle \delta q_1(X), \delta q_2(X) \rangle dm(X) \quad (18)$$

where B denotes a reference body, dm is the mass distribution and $X \in B \mapsto \delta q_i(X) \in \mathbb{R}^3$, $i = 1, 2$, are two deformations of the body, i.e. $\delta q_i \in T_q Q$.

If $\delta q_2 \in V_q \subset T_q Q$ is a vertical vector then there is an $\omega \in \mathbb{R}^3$ such that

$$\delta q_2(x) = \omega \times q(X)$$

And so

$$\langle \delta q_1, \delta q_2 \rangle = \langle \omega, M(q, \delta q) \rangle$$

where

$$M(q, \delta q_1) = \int_B (q(X) \times \delta q_1(X)) dm(X)$$

is the standard expression for the total angular momentum associated to the deformation δq_1 of the configuration q . It follows that δq_1 is horizontal iff $M(q, \delta q_1) = 0$. Thus

$$D_q = \{\delta q \in T_q Q : M(q, \delta q) = 0\}.$$

This is the basic fact which makes the language of connections on principal bundles useful for the control of deformable bodies.

If the horizontal distribution D is defined by the vanishing of the angular momentum M then the connection one-form Γ has the same kernel as M . Consequently we must have :

$$\Gamma(q) = R(q)M(q, \cdot)$$

for some invertible transformation $R(q) : Lie(G) \rightarrow Lie(G)$. [Note: that Γ and M are both one-forms on Q with values in $Lie(G)$ and they obey the same transformation law with respect to the G action.] To find $R(q)$ we use the normalization condition $\Gamma(q)(\alpha_q(\omega)) = \omega$ for the connection one-form. Now

$$M(q, v) = \alpha_q^t(v)$$

where

$$\alpha_q^t : T_q Q \rightarrow \text{Lie}(G) = \mathbb{R}^3$$

is the transpose of the map

$$\alpha_q : \text{Lie}(G) \rightarrow T_q Q$$

(α_q is the infinitesimal G-action). The normalization condition becomes $R(q)\alpha_q^t\alpha_q = \text{identity}$. Therefore $R(q) = (\alpha_q^t\alpha_q)^{-1}$. Now $\alpha_q^t\alpha_q = I(q)$ is the locked inertia tensor. This yields the "master formula" equation (7), (8).

3.5 Parallel Transport: the Control Law

Definition 8 A vector or vector field is said to be horizontal if it lies in D . A path is said to be horizontal if its derivatives all lie in D .

Our control law is that curves be horizontal.

With h denoting the horizontal lift operator, this control law is our original equation, (1). Now it is called the equation of parallel transport. It can be rewritten in the following ways

$$q^*\Gamma = 0$$

(cf. with eq. (5)) or

$$\dot{q} \in D_q(t)$$

The second of these has exactly the form of eq. (3) when written out in a trivialization. (Cf. eq. (16).)

Any of these equations is called the equation of parallel transport. A solution $q(t) = (x(t), g(t))$ is called "the parallel transport of $q^{(0)}$ along $x(t)$ ". Solving the equations of parallel transport for different initial conditions $q(0) \in Q$ defines a map

$$\mathbf{IP}_\gamma : Q_0 \rightarrow Q_1$$

where

$$Q_0 = \pi^{-1}(x(0)), Q_1 = \pi^{-1}(x(1)),$$

and

$$\mathbf{IP}_\gamma q(0) = q(T).$$

Here γ denotes the curve $x[0, T]$ in shape space. \mathbf{IP}_γ is called the parallel transport operator along the curve γ . It is a crucial and easily proved fact that this operator is independent of how the curve γ is parameterized.

In the case of the deformable body \mathbf{IP}_γ describes by how much the body has rotated due to the sequence of shape changes $x(t)$. This interpretation is most meaningful when

$$x(0) = x(T)$$

so that the final and initial shapes are the same. Then the configurations $q_0 = q(0)$ and $q_1 = q(T)$ differ by a rigid rotation $g_1 = g(T)$:

$$\mathbf{IP}_\gamma \mathbf{q}(0) = \mathbf{q}(T) = \mathbf{g}(T) \mathbf{q}(0)$$

In this case the parallel transport map is called the holonomy. It describes the net reorientation of the body.

The parallel transport operators satisfy

$$\mathbf{IP}_\gamma(\mathbf{g}\mathbf{q}) = \mathbf{g}\mathbf{IP}_\gamma(\mathbf{q})$$

and

$$\mathbf{IP}_{\gamma_2 \cdot \gamma_1} = \mathbf{IP}_{\gamma_2} \circ \mathbf{IP}_{\gamma_1}$$

In terms of the deformable body, this first identity says that if we rigidly rotate our body then perform a sequence of shape changes then the resulting reorientation is the same as if we had first performed the sequence of shape changes and then rotated the body. In the second identity γ_1 and γ_2 are two paths such that the end point of γ_1 is the initial point of γ_2 . Then $\gamma_1 \cdot \gamma_2$ stands for the single path obtained by joining the two paths together at this common endpoint.

The parallel transport equations (3, 1) are solved in a local trivialization by letting $M(t)$ be the fundamental solution to equation (3). Thus $M(0) = \text{identity}$. Then $\mathbf{g}(t) = \mathbf{g}(0)M(t)$ so that

$$\mathbf{IP}(\mathbf{x}(0), \mathbf{g}(0)) = (\mathbf{x}(T), \mathbf{g}(0)\mathbf{M}(T))$$

in the given trivialization.

Remark 7 In the physics literature one finds the notation

$$M(t) = P \exp\left\{-\int_0^t A(x)dx\right\}$$

for the fundamental solution, and hence for the parallel transport generator.

" $P \exp$ " stands for "path-ordered exponential" it has the following meaning. Consider partitions $0 \leq t_1 \leq t_2 \leq \dots \leq t_n \leq t$ of $[0, t]$ for which the mesh size goes to 0 as $n \rightarrow \infty$ as in the definition of the Riemann integral. Set

$$\alpha(s) = A(x(s)) \cdot \frac{dx}{dt}(s).$$

Then

$$P \exp - \int_0^t A dx = \lim_{n \rightarrow \infty} \exp(-\alpha(t_1)) \exp(-\alpha(t_2)) \dots (\exp(-\alpha(t_n)))$$

3.6 Controllability and Curvature

Set

$$X_i(q) = h(q) \cdot e_i$$

where $\{e_i\}$ is a basis for the set of controls (a smooth local frame field for S) and where h is the horizontal lift (control law eq. (1)). For example we might take $e_i = \frac{\partial}{\partial x^i}$, the coordinate vector fields, when (x^1, \dots, x^n) are (local) coordinates on the shape space S . The X_i are smooth horizontal vector fields on Q . They form a frame (pointwise basis) for the distribution $D = \text{image } h$.

Recall the following consequence of Chow's theorem:

Theorem 1 Suppose that Q is connected. If the X_i together with all of their iterated Lie brackets $[X_i, X_j], [[X_i, [X_j, X_k]]] \dots$ eventually span the tangent space of Q at every point. Then our system 1 provides accessibility: given any two points $q_0, q_1 \in Q$ there is a control law $u(t)$ which steers from q_0 to q_1

Remark 8 This theorem is, of course, valid for a general distribution, i.e. one not necessarily coming from a connection on a bundle. For treatments of the theorem, see Chow [9], Rashevski [23], or Sussmann, §3.2 [31].

One can algebracize the computations by thinking of the Lie bracket of horizontal vector fields at q as a map

$$[\cdot, \cdot]_q : D_q \times D_q \rightarrow T_q Q / D_q.$$

To do this take $v, w \in D_q$ and extend them to smooth horizontal vector fields V, W defined in a neighborhood of q . (Recall that horizontal means takes value in D .) Then set

$$[v, w]_q = [V, W](q) \text{mod} D_q$$

. To see that this is well-defined observe that

$$[fV, gW]_q = f(q)g(q)[V, W](q) \text{mod} D_q$$

for any smooth functions f and g , i.e. the operation is "tensorial": it really only depends on $v = V(q)$ and $w = W(q)$ and not on their horizontal extensions V, W .

Similar operations can be used to make tensorial sense out of higher brackets. See Gershkovich and Vershik [12].

When D is the horizontal space of a connection on a principal bundle then we have the splitting

$$D_q \oplus \text{Lie}(G) \cong T_q Q$$

so that we can and do identify

$$T_q Q / D_q = \text{Lie}(G).$$

Since $D_q = \ker \Gamma_q$ the identification is provided by the connection one-form. Moreover we can use the horizontal lift operator h_q to identify $T_{\pi(q)} S$ with D_q .

Definition 9 The curvature form F at q is the negative of the Lie bracket $-[\cdot, \cdot]_q$ at q after applying the above identifications. Specifically:

$$F(q)(v, w) = -\Gamma_q([h_q v, h_q w]_q)$$

Here $v, w \in T_{\pi(q)}S$. Thus $F(q)$ is a skew-symmetric bilinear form (two-form) on $T_{\pi(s)}S$ with values in the vector space $\text{Lie}(G)$.

The transformation law for the connection one-form Γ implies that

$$F(gq) = gF(q)g^{-1}$$

This transformation law is equivalent to saying that F is a two-form on S with values in the "adjoint bundle" which is a certain vector bundle over S with typical fiber $\text{Lie}(G)$. Alternatively, by using $h_q \circ d\pi_q$ instead of h_q in the definition, we can think of F as a two-form on Q with values in the vector space $\text{Lie}(G)$. As such one calculates:

$$F = d\Gamma - [\Gamma, \Gamma]$$

or in a local trivialization

$$F(x, g) = g(dA - [A, A])$$

Here

$$[\Gamma, \Gamma](v, w) = [\Gamma(v), \Gamma(w)]$$

and

$$= \Gamma(v) \times \Gamma(w) \text{ for } G = SO(3)$$

with the last Lie brackets being those in $\text{Lie}(G)$.

To prove these formulas use the Cartan's formula

$$(d\Gamma)(X, Y) = (d(\Gamma(Y)))(X) - (d(\Gamma(X)))(Y) - \Gamma([X, Y])$$

for the exterior derivative d and apply it to horizontal vector fields.

Remark 9 If G is Abelian, for example $G = SO(2)$, then $gF(q)g^{-1} = F(q)$ so that the curvature is an old-fashioned two-form on S . It can be defined by $\pi^*F = d\Gamma$.

The curvature form is a covariant derivative of the connection form. We can directly relate higher covariant derivatives of the curvature form

$(D_X F)(Y, Z), (D_X D_Z F)(Y, W)$ etc., to the higher intrinsic Lie brackets $[h_q X, [h_q Y(q), h_q Z]]_q, [h_q X, [h_q Z, [h_q Y, h_q W]]]_q$

As a result of these relations we have the following consequence of Chow's theorem [9](See Hermann [16] for more details. Also, somewhere in Hermann's 24 volume set.)

Theorem 2 *[Ambrose-Singer Theorem]* Suppose that Q is connected. Suppose that for some $q \in Q$ the image of the curvature Γ_q , together with all of its covariant derivatives span $\text{Lie}(G)$. Then any two points q_0, q_1 of Q can be joined by a horizontal path.

Remark 10 The original The Ambrose-Singer theorem is more general than this. We have given a computationally useful version just as we did with the original Chow theorem.

Example 2 (again) Here $A = \alpha(xdy - ydx)$ The Lie algebra is one dimensional and so Abelian: there is no $[\Gamma, \Gamma]$ term. Thus

$$F = dA = \alpha dx \wedge dy$$

As long as $\alpha \neq 0$ any two points can be joined by a horizontal path.

4 Optimal Control

Consider a pointwise cost of the form

$$c = \frac{1}{2}R(x)(u, u)$$

and corresponding cost functional

$$C[x, u] = \int_0^T c(x(t), u(t)) dt.$$

$R(x)(\cdot, \cdot)$ is an inner product (positive definite quadratic form) on controls which depends smoothly on $x \in S$. In other words, it is

a a Riemannian metric on the shape space S . The cost C of a control strategy is thus the corresponding integrated kinetic energy. Or, what is effectively the same, the cost is the length of the corresponding projected path $x(t) = \pi(q(t))$ in S . The problem is to minimize C among all controls steering from q_0 to q_1 in time T .

Equivalently, we could fix shapes $x_0 = \pi(q_0)$, $x_1 = \pi(q_1)$ and a parallel transport operator $\mathbf{IP} \in \text{Hom}(\mathbf{Q}_0, \mathbf{Q}_1)$ where $Q_i = \pi^{-1}(x_i)$. These two types of boundary conditions are related by $\mathbf{IP}(q_0) = q_1$. Thus I call the problem the "isoparallel problem". In case $x_0 = x_1$ it is the isoholonomic problem. Example 1 is the isoarea problem which is dual to the famous isoperimetric problem.

We may think of the cost C as the 'efficiency' of a given control strategy. Thus our optimal control problem is to find the most efficient way to deform a deformable body ao as to achieve a desired reorientation. This problem was first formulated by Shapere and Wilczek [25] [26] in connection with the question of how certain microorganisms swim. They solved the corresponding linearized problem, that is, the case of infinitesimal shape deformations.

From a geometric point of view the natural cost to use is the length of the path in shape space, or what is effectively the same, its integrated kinetic energy. These are to be calculated with respect to the metric defined in §3.4. The optimal control problem then becomes the problem of finding that horizontal path which connects q_0 to q_1 in a time T and which minimizes the integrated kinetic energy over this time interval.

The basic facts regarding the extremals to this problem are

Theorem 3 [20] *The normal extremals for the above optimal control problem obey the same differential equations as those of a particle under the influence of the gauge potential A (cf. eq. (16)) travelling on the Riemannian manifold (S, R) . In case the group G is $SO(2)$ (planar robot) these are the standard Larentz equations which govern the motion of a charged particle in the magnetic field $F = dA$.*

Theorem 4 [22], [20] *The abnormal extremals are those horizontal curves*

$q(t)$ such that there exists a nonzero element μ in the dual of $\text{Lie}(G)$ such that $\mu(F(\dot{q}, \cdot)) = 0$.

In case $\dim(Q) = 3$, $\dim(G) = 1$ (for example $G = SO(2)$) so that $\dim(S) = 2$ this condition implies that the projection $x(t) = \pi(q(t))$ lies on the zero level set of the magnetic field B where $F = BdS$ and dS denotes the area form on S . If this level set is generic ($dB \neq 0$ when $B = 0$) then every sufficiently short subarc of $q(t)$ is the unique cost minimizing path between its endpoints. Moreover in this case the abnormal extremal is stable under perturbations of the cost and the control law (eq. 1).

Remark 11 (Explanation) "Abnormal" refers to the fact that we can take $\lambda_0 = 0$ when we compute the extremals of $\lambda_0 C + \lambda G$. Here C is the cost function and G represents the horizontal constraints on the paths. See Bliss [4], Hermann [16], and especially Morse and Myers [19] for more information on normal versus abnormal curves.

The optimal control Hamiltonian H is the Hamiltonian defining the differential equation for theorem 3. It is the Hamiltonian furnished us by Pontrjagin's principle in the normal case. To write it down, choose a local frame of vector fields X_μ for S and calculate the matrix of inner products

$$g_{\mu\nu}(x) = R(x)(X_\mu(x), X_\nu(x))$$

let $g^{\mu\nu}$ be the inverse matrix. Then

$$H = \frac{1}{2} g^{\mu\nu}(hX_\mu)(hX_\nu)$$

where, as before, h denotes horizontal lift. Also, the hX_μ , being vector fields on TQ , are fiber-linear functions on T^*Q . Thus H is a fiber-quadratic function on T^*Q .

5 Two Examples

Example 3 Example 1, revisited

$$X_1 = \frac{\partial}{\partial x} = (1, 0, 0)$$

$$X_2 = \frac{\partial}{\partial y}$$

$$hX_1 = \frac{\partial}{\partial x} + \alpha y \frac{\partial}{\partial \theta}$$

In coordinates these are the fiber linear functions

$$hX_1 = p_x + \alpha y p_\theta$$

$$hX_2 = p_y - \alpha x p_\theta$$

(Just replace $\frac{\partial}{\partial x}$ with p_x , etc.)

$$H = \frac{1}{2} \{(p_x + \alpha y p_\theta)^2 + (p_y - \alpha x p_\theta)^2\}$$

This is the Hamiltonian for a particle of charge p_θ in a constant magnetic field of strength α . The solutions are circles as any textbook on electromagnetism will say.

Example 4 The Falling Cat of Kane and Scher

Kane and Scher modeled the maneuver by which the falling cat, dropped from upside down with zero angular momentum, rights itself. Their model consists of two identical axially symmetric rigid bodies joined along their symmetry axes by a special kind of joint ("no-twist").

If the joint were instead a ball-and-socket joint (three degrees of freedom) and if we ignored collisions of the bodies then the configuration space would be

$$Q_{b-s} = SO(3)_f \times SO(3)_b.$$

The subscripts "f" and "b" stand for "front" and "back". The subscript "b-s" is for ball-and-socket.

Kane and Scher make the following beautiful choice of coordinates. Let 1_f and 1_b denote the axis of symmetry, the backbone, of the front and back body halves. Let ψ denote the angle between 1_b and 1_f . As long as $\psi \neq 0, \pi$. 1_f and 1_b define a plane \mathcal{P} the plane which contains them and the joint. The angles θ_f and θ_b will denote the angles made by the respective feet axes

2_f and 2_b and this plane. The orientations are chosen so that θ_f increases as the front rotates about the 1_f axes, orientated toward the head, according to the right hand rule. And θ_b increases as the back half rotates about the 1_b axis, oriented to point toward the tail. Then $(\psi, \theta_f, \theta_b)$ coordinatize the shape space S_{b-s} , where the subscript stands for ball-and-socket joint. To define coordinates on Q_{b-s} , we need a local section $s : S_{b-s} \rightarrow Q_{b-s}$. This induces coordinates as in §3.1 :

$$(\psi, \theta_f, \theta_b, g) \longmapsto g s(\psi, \theta_f, \theta_b).$$

We choose the section s (choice of gauge) by insisting that the plane \mathcal{P} is the yz plane, that 1_f makes an angle $\psi/2$ with the positive z -axis, that 1_b makes the angle $-\psi/2$ with the same, that when the coordinates $\theta_f = \theta_b = 0$ then the cat's legs are lying in the plane, and that if in addition $\psi = \pi$ so that the cat is fully extended (backbone straight) then the feet are pointed straight up (cat upside-down). The negative z axis is the direction of gravity.

Let $\{\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3\}$ be the orthonormal frame in the inertial frame associated to these axes. e (no bold-face) will denote the identity matrix corresponding to choice of section s . Set

$$c = \cos(\psi/2) \quad s = \sin(\psi/2)$$

Let I_3 denote the moment of inertia of either body for resisting spinning about its symmetry (1_f or 1_b) axis. Then the total angular momentum M generated by the deformation $(\psi, \theta_f, \theta_b, e) \rightarrow (\psi + d\psi, \theta_f + d\theta_f, \theta_b + d\theta_b, e)$ is

$$M = I_3 \{[c\mathbf{e}_3 + s\mathbf{e}_2]d\theta_f + [c\mathbf{e}_3 - s\mathbf{e}_2]d\theta_b\}.$$

The moment of inertia tensor of the configuration $s(\psi, \theta_f, \theta_b)$ with respect to the inertial frame is the diagonal matrix

$$I(\psi) = 2I_1 \text{diag}[c^2 + (1 + \beta)s^2, c^2 + \alpha s^2, \alpha c^2 + (1 + \beta s^2)]$$

where $I_1 = I_2$ is the moment of inertia of either body half with respect to any axis orthogonal to the symmetry axis, and

$$\alpha = \frac{I_3}{I_1}$$

is the ratio of moments of inertia and

$$\beta = \frac{ml^2}{I_1}$$

where l is the distance of the center of mass of either body half from the joint and m is the mass of either body half.

Plugging these results into the master formula (8) for the connection one-form where x stands for $(\psi, \theta_f, \theta_b)$ we obtain

$$A(x) = \frac{\alpha}{2} \frac{c(d\theta_f + d\theta_b)}{\alpha c^2 + (1 + \beta)s^2} \mathbf{e}_3 + \frac{\alpha s(d\theta_f - d\theta_b)}{c^2 + \alpha s^2} \mathbf{e}_2$$

Recall that the vector coefficients \mathbf{e}_i refer to infinitesimal rotations about these axes.

The no-twist condition of Kane and Scher is

$$d\theta_f = -d\theta_b.$$

We can think of this as saying that the cat is not allowed to break its own back. Integrating we obtain

$$\theta_f = -\theta_b$$

and so we set

$$\theta := \theta_f.$$

We would like to thank Mark Enos at this point for pointing out that the no-twist condition is **not** the same as connecting the two halves by a U-joint (sometimes called a Hooke's joint)!

The full connection-one form $dg + gA(x)$ for the no-twist joint is then

$$\{d\chi + \Phi(\psi)d\theta\}\mathbf{e}_2$$

where

$$\Phi(\psi) = \alpha \frac{s}{c^2 + \alpha s^2}$$

and where we have written $g = \exp(\chi \mathbf{e}_2)$ = rotation about the y-axis by an amount χ .

We have just proved the remarkable fact, observed by Kane and Scher, that for no-twist deformations the connection one-form is Abelian. Consequently the re-orientation $\Delta\chi$ of the model cat due to a given shape deformation can be calculated by a single quadrature. In bundle-theoretic language, what has happened is that by restricting the joint to be no-twist we reduce the structure group of the bundle from $SO(3)$ to the (disconnected) group $O(2)$.

The map $\pi : Q \rightarrow S$ in the no-twist case is the Hopf fibration $SO(3) \rightarrow \mathbf{RP}^2$, where \mathbf{RP}^2 denotes the real projective plane. This is another surprise: the shape space for a no-twist joint is the real projective plane! This fact can be seen algebraically by using Euler angle (i.e. exponential) coordinates on $SO(3)$, the ball-and-socket shape space, to express the no-twist joint constraint. Then the no-twist shape space becomes identified with $\exp(\mathbf{R}^2) \subset SO(3)$ where $\exp : \mathbf{R}^3 = \text{Lie}(SO(3)) \rightarrow SO(3)$ is the usual exponential map and \mathbf{R}^2 is the xz plane. If we do not let the body halves pass through each other then we destroy all topology: the shape space becomes diffeomorphic to a disc. In fact, if we imagine the body halves to be infinitely thin rods then shapes we must delete are those with $\psi = 0$. This is a circle of shapes (vary θ) and corresponds to the line at infinity in the classical conception of the projective plane. Thus the shape space in this case is a classical affine plane.

The normal extremals for the optimal control problem are the trajectories of charged particles travelling on the real projective plane with metric defined by the pointwise cost function. For more details concerning this problem consult a forthcoming paper or write the author.

The abnormal extremals are the horizontal lifts of certain lines of latitude $\psi = \text{constant}$ where the constants are the solutions to the equation

$$\frac{d\Phi}{d\psi} = 0.$$

6 Restrictions and Uses of the Dictionary

6.1 Restrictions

The control law must be linear in the control in order to be able to use the dictionary. A partial extension exists for laws which are affine in control.

We require the control law $u \mapsto h(q)u$ to be maximal rank for all q . In some cases this could be a significant restriction. We require the number n of state variables to be greater than the number m of controls. (The case $m \geq n$ is trivial from a theoretical point of view.) Any control system satisfying these properties can always be put in local bundle form of equation (3).

The dictionary is most useful for systems with some symmetry, namely that of equation (4). It is even more useful if the number $n - m$ (the dimension of G) is much less than the number m of controls.

6.2 Utility

Some computations become extremely easy. For example we have shown that using ideas from gauge theory it can be very easy to check controllability. The dictionary allows easy identification of important special curves and submanifolds. For example, singular arcs lie on the surfaces {curvature = 0}. The dictionary can be used as an aid to intuition. For example, solutions to the naturally formulated optimal control problems (quadratic cost) are trajectories in a magnetic field defined by the curvature.

By using the dictionary, or at least its geometric philosophy, we can find the correct formulations of some open problems in nonholonomic path planning, for example the problem of feedback stabilization. See the next section.

7 Feedback Stabilization

Suppose the point $q_0 \in Q$ is the desired goal. For example, it might represent the cat's desired configuration: feet pointing down back slightly arched. In feedback stabilization we try to design a feed back control law

$$u = u(q)$$

which steers all points in some neighborhood of q_0 to q_0 . Such a feedback law is called a feedback stabilization strategy. We now cite the following basic theorem of Brockett. [6].

Theorem 5 [Brockett] *Let $\dot{q} = h(q, u)$ be the control law. If the map $(x, u) \mapsto h(x, u)$ does not map a neighborhood of $\{q_0\} \times \text{controls}$ onto a neighborhood of 0, then there is no continuous feedback stabilization strategy.*

In our situation h can always be put in the form:

$$(x, g; u) \mapsto (u, -A(x, g)u) \in \mathbb{R}^m \times \mathbb{R}^{n-m}$$

Points of the form $(0, \xi), \xi \neq 0$ are never in the image of this map. Thus our systems never admit continuous feedback stabilization laws.

However, if we give up the idea of stabilizing to a point and instead pick a desired subvariety to stabilize onto we can achieve victory.

Example 5 Rewrite the Heisenberg flywheel in polar coordinates: (r, ϕ) for the xy-plane. Let us set $z = \theta$. We will think of (r, ϕ, z) as cylindrical coordinates on three-space for the purposes of visualization. The system becomes

$$\begin{aligned}\dot{r} &= u_1 \\ \dot{\phi} &= u_2 \\ \dot{z} &= -\frac{1}{2}r^2u_2\end{aligned}$$

Our goal is to stabilize onto a circle of radius r_1 at a height z_1 . Set

$$u_1 = -c(r - r_1)$$

$$u_2 = z - z_1$$

One easily calculates that the distance squared

$$V = (r - r_1)^2 + (z - z_1)^2$$

of a point from our objective circle is a Liapanov function. That is, $\frac{dV}{dt} \leq 0$ everywhere and $\frac{dV}{dt}(q) = 0$ only for points on the unit cylinder. Thus we have stabilized the system onto our circle by means of a feedback stabilization law. Note that in order to do this we gave up all control of our position on that circle.

In terms of our mass fly-wheel model we can feedback stabilize the distance r of the mass from the fly wheel and the inertial angle of the flywheel but not the angle of the rod. By an alternative strategy we could have stabilized instead the rod angle or the difference of the two angles ψ .

Example 6 Brockett has proposed the following generalization of Heisenberg's flywheel:

$$\dot{x} = u$$

$$\dot{\xi} = x \wedge u$$

Here $x, u \in \mathbb{R}^n$ and $\xi \in \text{Lie}(SO(n)) = \Lambda^2 \mathbb{R}^n$. The feedback law

$$u = -\xi x$$

stabilizes the system onto the subvariety

$$\{(x, \xi) : \xi x = 0\}.$$

A general framework is as follows. Suppose that our objective is to feedback stabilize control system (1) onto the submanifold $N \subset Q$. We say that N is transverse to the horizontal distribution D if

$$T_q N + D_q = T_q Q \text{ for all } q \in N.$$

Let us suppose that N is defined by some equation: $N = p^{-1}(y_0)$ where

$$Q \xrightarrow{p} Y$$

is a smooth map and y_0 is a regular value of Y , that is, $dp(T_q Q) = T_{p(q)} Y$ whenever $p(q) = y_0$. If N is a smooth manifold with trivial normal bundle then it is always possible to write N in this way. We think of Y as the objectives. In our first stabilization example $p(r, \phi, z) = (r, z)$.

The transversality condition on D and N translates to

$$dp_q D_q = T_{p(q)} Y$$

for $q \in N$. Now we suppose that $\dim(Y) = \text{rank}(D)$ which is the same as

$$\dim(N) + \text{number of controls} = \dim(Q).$$

Then $dp_q \circ h(q)$ is a linear isomorphism of the controls u onto the objectives $T_y Y$. The control law (1) induces the law

$$\dot{y} = \frac{\partial p}{\partial q}(q) h(q) u$$

for the objectives. ($\frac{\partial p}{\partial q}$ and dp_q are different symbols for the same thing.) Choose coordinates so that $y_0 = 0$. Then the feedback law

$$u(q) = -\left[\frac{\partial p}{\partial q}(q) \circ h(q)\right]^{-1} \cdot p(q) \quad (19)$$

yields the differential equation $\dot{y} = -y$. We have proved:

Theorem 6 Consider control law (1). Suppose $N \subset Q$ is a submanifold which is transverse to the control distribution and whose normal bundle is trivial. Then feedback law (19) (locally) stabilizes the system onto N .

As a trivial illustration of the theorem take $p : Q \rightarrow Y$ to be the bundle projection $\pi : Q \rightarrow S$ to be the bundle projection. One easily checks the hypothesis. The induced control system on S is just $\dot{x} = u$ which is obviously feedback stabilizable.

Remark 12 Independent of this work, Bloch and Rehyhanoglu and McClamroch came up with essentially the same idea for stabilization onto submanifolds. See their preprint [2].

As a final remark we would like to advertise a very recent result of Coron which is much deeper than the just proved theorem.

Theorem 7 [*Coron's Theorem*] Suppose that the distribution D is bracket generating and that $Q = \mathbb{R}^n$. Then there exists a time dependent feedback control

$$u = u(x, t)$$

which is periodic in time:

$$u(x, t) = u(x, t + \tau)$$

and under which the origin of Q becomes globally asymptotically stable. The period τ can be any fixed positive number.

Remark 13 The proof of Brockett's theorem is topological. The index of the vector field $x \mapsto h(x)u(x)$ is an obstruction to stabilization. It seems that the essence of Coron's idea is that by suspending the control system to one on $Q \times S^1$ the obstructions vanish. Coron's theorem is of course also true if Q is a general manifold and global stability is replaced by local stability.

References

- [1] W. Ambrose and I.M. Singer, **A Theorem on holonomy** Trans. of A.M.S. , 428-453, (1953).
- [2] A. Bloch, M. Reyhanoglu, and N. McClamroch, *Control and Stabilization of Nonholonomic Dynamic Systems* preprint (1991).
- [3] D. Bleecker, **Gauge Theory and Variational Principles**, Addison-Wesley, (1981).
- [4] G.A. Bliss, **Lectures on Calculus of Variations**, Univ. of Chicago Press, (1946).
- [5] R.W. Brockett, *Nonlinear Control Theory and Differential Geometry*, Proc. of the Int. Congress of Mathematicians, Warszawa ,(1983).

- [6] R.W. Brockett, *Control Theory and Singular Riemannian Geometry*, in New Directions in Applied Mathematics, P.J. Hilton and G.S. Young, editors, Springer, (1981).
- [7] C. Carathéodory, **Calculus of Variations and Partial Differential Equations of the First Order**, vol. 2 Holden-Day, S.F., CA, (1967).
- [8] S.S. Chern, **Complex Manifolds without Potential Theory**, Springer-Verlag, (1979).
- [9] W.L. Chow, *Über Systeme van Linearen partiellen Differentialgleichungen erster Ordnung*, Math. Ann., vol. 117, 98-105, (1939).
- [10] J.-M. Coron, *Global Asymptotic Stabilization for Controllable Systems Without Drift*, preprint, (1991).
- [11] V. Ya Gershkovich and A.M. Vershik, *Non-holonomic Riemannian Manifolds*, in **Dynamical Systems** v. 7, part of the Mathematical Encyclopaedia series, vol.16 (in Russian), MIR pub., 1988.
- [12] V. Ya Gershkovich and A.M. Vershik, *Non-holonomic Manifolds and Nilpotent Analysis*, J. Geom. and Phys. vol. 5, no. 3, (1988).
- [13] O. Granchina and A.M. Vershik, *Reduction of Non-holonomic Variational Problems to Isoperimetric Ones and Connections in Principal Bundles*, preprint (in Russian) (1990).
- [14] A. Guichardet *On rotation and vibration motions of molecules* Ann. Inst. H. Poincaré, Phys. Theor., **40** no.3 329-342 (1984).
- [15] U. Hamenstädt, *Some Regularity Theorems for Carnot-Caratheodory Metrics*, J.D.G. , v. 32, 819-850 (1990).
- [16] R. Hermannn, *Some Differential Geometric Aspects of the Lagrange Variational Problem*, Indiana Math. J., 634-673 (1962).
- [17] R. Hermannn, *Geodesics of Singular Riemannian metrics*, Bull. AMS, vol. 79, no.4, 780-782, (1973).
- [18] T.R. Kane and M.P. Scher *A dynamical explanation of the falling cat phenomenon*, Int'l H. Solids and Structures, **5** 663-670, (1969).

- [19] M. Morse and S. Myers, *The Problesm of Lagrange and Mayer with Variable End Points* in **Selected Papers** ed. by R. Bott, Springer-Verlag (1981).
- [20] R. Montgomery , *The Isoholonomic Problem and Some Applications*, Comm. Math. Phys., vol. 128, 565-592, (1990).
- [21] R. Montgomery , *Optimal Control of Deformable Bodies and Its Relation to Gauge Theory* in **The Geometry of Hamiltonian Systems**, proceedings of a Workshop held in June 1988 at M.S.R.I., T. Ratiu ed., Springer-Verlag, (1991).
- [22] R. Montgomery, *Geodesics Which Do Not Satisfy the Geodesic Equations*, preprint , (1991).
- [23] P. K. Rashevski, *About connecting two points of complete nonholonomic space by admissible curve* (in Russian), Uch. Zapiski ped. inst. Libknexta, no. 2, 83-94, (1938).
- [24] C.B. Rayner *The Exponential Map for the Lagrange Problem on Differentiable Manifolds*, Phil. Trans. of the Royal Soc. of London, ser. A, Math. and Phys. Sci. no. 1127, v. 262, 299-344, (1967).
- [25] A. Shapere and F. Wilczek, *Self-propulsion at low Reynolds number* , Physical Review Letters, vol. 58, 2051-2054, (1987).
- [26] A. Shapere and F. Wilczek ed., **Geometric Phases in Physics**, Adv. Series in Math Phys v. 5, World Sci. Press, (1989).
- [27] A. Shapere, *Self-propulsion at Low Reynolds Numbers*, PhD thesis, Physics, Princeton Univ. (1989).
- [28] M. Spivak **Introduction to Differential Geometry**, vol. 2, Pub. or Perish Press, (1979).
- [29] N. Steenrod **Topology of Fibre Bundles** Princeton U. Press (1951).
- [30] R. Strichartz, *Sub-Riemannian Geometry*, J. Diff. Geom. 24, 221-263, (1983).
- [31] H. Sussmann, *Lie Brackets, Real Analyticity and Geometric Control in Differential Geometric control Theory* a conf. proceedings ed. by R.W. Brockett, R. Millman, and H.J. Sussmann, Birkäuser, (1983).

- [32] L.C. Young **Lectures on the Calculus of Variations and Optimal Control Theory**, Chelsea, (1980).

10

Optimal Nonholonomic Motion Planning for a Falling Cat

C. Fernandes, L. Gurvits and Z.X. Li*

Robotics Research Laboratory

Courant Institute of Mathematical Sciences

New York University, NY, NY 10012

February 6, 1992

Abstract

How does a falling cat change her orientation in midair without violating the angular momentum constraint? This has become an interesting question to both control engineers and robotists. In this paper, we address this problem together with a constructive solution. First, we show that *a falling cat problem* is equivalent to the *constructive nonlinear controllability problem*. Thus, the same principle and algorithm used by a falling cat can be used for space robotic applications such as reorientation of a satellite using rotors, attitude control of a space structure using internal motion and for other robotic tasks such as dexterous manipulation with multifingered robotic hands and nonholonomic motion planning for mobile robots. Then, using ideas from Ritz

*This work was supported in part by NSF grants No. CDA-9018673, No. IRI-9003986 and No. MSS-9010900.

Approximation Theory we develop a simple algorithm for motion planning by a falling cat. Finally, we test the algorithm through simulation on two widely accepted models of a falling cat. It is interesting to note that one set of simulated trajectories is close to trajectories used by a real cat.

1 Introduction

It is well known that a cat, when released from an upside down configuration starting from rest, is able to land on her feet. However, we also know from classical mechanics that the angular momentum of a falling cat is conserved. Thus, *how does a falling cat change her orientation without violating angular momentum conservation* becomes an interesting problem not only to physiologists ([Mcd55]), mechanical engineers ([KS69]), control engineers ([LS90], [SL91], [Bro81], [BD91] and [WK89]), mathematicians and physicists ([Mon89] and references therein), but also to robotists ([FGL91] and [MS90]).

Recently, it has been understood that *a falling cat problem*, in essence, is equivalent to a *nonholonomic variational problem* studied by mathematicians ([VG88] and [AKN85]), a *Gauge theoretic problem* studied by physicists ([Ber85]), a *Nonholonomic Motion Planning* (NMP) problem studied by robotists ([LC90], [Lau92], [PD90] and [NM90]) and a *constructive nonlinear controllability problem* studied by control theoreticians ([Bro81], [BM89], [HH70] and [Isi85]). It is shown in ([FGL91] and references therein) that the same principle used by a falling cat can be used for space robotic systems such as reorientation of a satellite using only two rotors, attitude control of a space structure (e.g., the Hubble Space Telescope, the Space Station Freedom and an Autonomous Space Robotic Servicer) using internal motion provided by manipulators attached to the structure, or solar panels, for dexterous manipulation by multifingered robotic hands ([Li89]) and for steering mobile robots subject to classic nonholonomic constraints ([FGL91]).

The problem of *constructive nonlinear controllability* has been a long standing one. Since the work of Chow ([Cho40]) there has been a great deal of research activity on this problem, most noticeably by Haynes and Hermes ([HH70]), Brockett ([Bro81]), Hermann and Krener ([HK77]), Sussmann ([SJ72]), and recently by Bloch and McClamroch ([BMR90]), Brockett and Dai ([BD91]), Sussmann and Lafferriere ([LS90] and [SL91]), and Murray and Sastry ([MS90]). Their approaches can be classified as (1) *the stabilization via nonsmooth feedback approach* ([BMR90]), as smooth feedbacks do not in general exist ([Kaw89] and [SS80]), (2) *the open loop motion planning approach* ([LS90], [SL91] and [HH70]), and (3) *the harmonic analysis approach* ([Bro81] and [MS90]).

There are certain advantages and disadvantages associated with each of these approaches. For example, while the *harmonic analysis approach* is capable of computing optimal solutions it assumes a certain structure in the system. On the other hand, while the *open loop motion planning approach* takes a general form of a system the solution it computes is not guaranteed to be optimal.

In this paper, by combining the advantages of Approach (2) and (3) we develop a numerical algorithm (called the Basis Algorithm) that computes approximate optimal solutions of the problem. The guide line of the Basis Algorithm is Ritz Approximation Theory ([CH53]) which approximates solution of an ∞ -dimensional optimization problem by solutions of a family of finite-dimensional optimization problems.

The paper is structured as follows: In Section 2, we use dynamical models of a falling cat and conservation of angular momentum to formulate the *falling cat problem* as a nonlinear control problem without drift. The control vector fields here correspond to directions of motion which satisfy the angular momentum *constraint*. In Section 3, we present the Basis Algorithm, where a family of finite-dimensional optimization problems are introduced via proper choice of a basis for the control space and restricting the control input to the first few harmonics of the basis. Special techniques are then developed for solving these finite-dimensional problems. In Section 4, we present simulation results of the Basis

Algorithm on two models of a falling cat. Surprisingly enough, one set of simulation results closely resembles the real trajectories employed by a falling cat.

2 Dynamical Models

It has been observed in [KS69] that the tail of a cat has negligible effect on the maneuvering process. Thus, a reasonable model of a falling cat is consisted of the upper body and the lower body, both treated as rigid, coupled to each other by either a ball-in-socket joint or a universal joint. If modeled by a ball-in-socket joint, the upper body would have three degrees of freedom (dof) relative to the lower body, and 2 dof otherwise. In this paper, we will consider both models as a ball-in-socket joint model would give us a system with 6 states and 3 control inputs and universal-joint model a system with 5 states and 2 control inputs.

2.1 A Ball-in-socket Joint Model

Consider the system shown in Figure 1. The coordinate frames C_0, C_1 and C_2 are called, respectively, the inertial reference frame, the body fixed frame to body-1 (the upper body) and the body fixed frame to body-2 (the lower body). We assume that the origins of the body frames coincide with the mass centers of the respective bodies. For body $i = 1, 2$, we denote by $(r_i, A_i) \in \mathbb{R}^3 \times SO(3)$ the position and orientation of frame C_i relative to the inertial frame C_0 , and also by $r \in \mathbb{R}^3$ the position vector of the mass center of the whole system.

If $\{(r_1(t), r_2(t), A_1(t), A_2(t))\} \in \mathbb{R}^3 \times \mathbb{R}^3 \times SO(3) \times SO(3)$ represent a trajectory of the system, the kinetic energy of the system has the form

$$K = \frac{1}{2} \sum_{i=1}^2 \langle I_i w_i, w_i \rangle + \frac{1}{2} \sum_{i=1}^2 m_i \|\dot{r}_i\|^2 \quad (1)$$

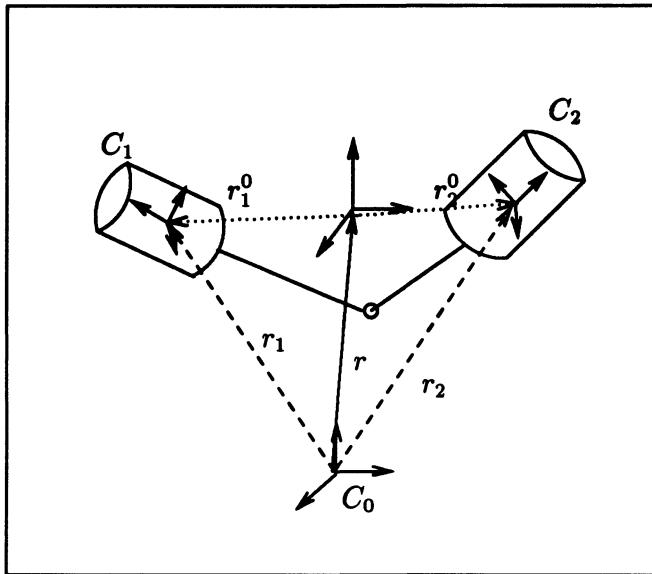


Figure 1: A rigid-body model of a falling cat

where

$$\begin{aligned} I_i &\in \Re^{3 \times 3} = \text{inertia tensor of body-}i, \\ m_i &= \text{mass of body-}i, \\ \hat{w}_i &= A_i^T \dot{A}_i, \text{ angular velocity of body } i. \end{aligned}$$

and $\hat{\cdot}$ is the operator identifying a 3-vector with a 3 by 3 skew-symmetric matrix.

From the figure we have the following kinematic relations

$$r_1 = r + r_1^0, \quad r_2 = r + r_2^0 \quad (2)$$

and

$$m_1 r_1^0 + m_2 r_2^0 = 0, \quad r_2^0 = r_1^0 + A_1 s_1 - A_2 s_2 \quad (3)$$

where $s_i \in \Re^3, i = 1, 2$, is the hinge position vector relative to frame C_i and $m = m_1 + m_2$ is the total mass of the system.

Applying Eqs (2) and the first equation of (3) to Eq. (1) yield

$$K = \frac{1}{2} \sum_{i=1}^2 \langle I_i \hat{w}_i, \hat{w}_i \rangle + \frac{1}{2} \sum_{i=1}^2 m_i \|\dot{r}_i^0\|^2 + \frac{1}{2} m \|\dot{r}\|^2. \quad (4)$$

Using Eqs. (3) the kinetic energy expression can be further simplified to

$$K = \frac{1}{2} \sum_{i=1}^2 \langle I_i w_i, w_i \rangle + \frac{1}{2} \|\dot{A}_1 s_1 - \dot{A}_2 s_2\|^2 + \frac{1}{2} m \|\dot{r}\|^2, \quad \epsilon = m_1 m_2 / m.$$

Since

$$\dot{A}_i s_i = A_i \hat{w}_i s_i = -A_i \hat{s}_i w_i, \quad i = 1, 2,$$

we finally have

$$K = \frac{1}{2} [w_1^T, w_2^T] \begin{bmatrix} J_1 & J_{12} \\ J_{12}^T & J_2 \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} + \frac{1}{2} m \|\dot{r}\|^2, \quad (5)$$

where

$$J_i = I_i + \epsilon \hat{s}_i^T \hat{s}_i, \quad i = 1, 2,$$

and

$$J_{12} = \epsilon \hat{s}_1 A_1^T A_2 \hat{s}_2.$$

It is now clear from (5) that motion of the system splits into translational motion of the mass center and rotational motion about the mass center of the system. The equations governing translational motion obey Newton's Second Law

$$m \ddot{r} = - \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix}$$

where g is the gravity constant. Thus, if a cat is dropped at a height $r(0) = (0, 0, r_z(0))^T$, the duration of time she has to complete the maneuver is given by

$$T = r_z(0)/2g.$$

For rotational motion, the configuration space is $Q = SO(3) \times SO(3)$ together with the following kinetic energy

$$K(\dot{q}, \dot{q}) = \frac{1}{2} [w_1^T, w_2^T] \begin{bmatrix} J_1 & J_{12} \\ J_{12}^T & J_2 \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}.$$

The equations of motion for the system have been derived using *Variational Principles* in ([WK89]) and have the form

$$\begin{bmatrix} J_1 & J_{12} \\ J_{12}^T & J_2 \end{bmatrix} \begin{bmatrix} \dot{w}_1 \\ \dot{w}_2 \end{bmatrix} + \begin{bmatrix} w_1 \times J_1 w_1 + \epsilon \hat{s}_1 A \hat{w}_2 \hat{s}_2 w_2 \\ w_2 \times J_2 w_2 + \epsilon \hat{s}_2 A^T \hat{w}_1 \hat{s}_1 w_1 \end{bmatrix} = \begin{bmatrix} -A \\ I \end{bmatrix} \tau \quad (6)$$

where $A = A_1^T A_2$ is the *shape* of the system and $\tau \in \mathfrak{R}^3$ the actuation torque in the ball-in-socket joint in body-2 frame. Since the coefficient matrix of τ , often identified as the decoupling matrix, has rank only 3 the system is not able to track an arbitrary trajectory in the configuration space. On the other hand, by projecting the dynamics into the shape space an arbitrary shape space trajectory can be realized using a “computed-torque” type of control algorithm (it is called the *virtual manipulator control* in [VD87]).

In the absence of external torque, which is the case for a falling cat and many other space robotic systems (see [FGL91] for more examples), the (spatial) angular momentum of the system is conserved. In this case, the conservation of angular momentum can be treated as *constraint* in planning motion for the system.

To compute the angular momentum, we can use either the classic approach as in [Gol80] or the modern geometric approach as in [AM78]. Following [AM78], we let the rotational group $SO(3)$ act on the configuration space $Q = SO(3) \times SO(3)$, lift the action to the cotangent bundle of Q and compute the momentum map of the lifted action. As a result we have the following expression for the angular momentum, $\mu \in \mathfrak{R}^3$, of the system ([FGL91])

$$\mu = (A_1 J_1 + A_2 J_{12}^T) w_1 + (A_2 J_2 + A_1 J_{12}) w_2. \quad (7)$$

If a cat is released with *zero angular momentum*, then by the *conservation law of angular momentum*, the velocity vector (w_1, w_2) , of the cat along a path $q(t) = (A_1(t), A_2(t)) \in Q$ has to satisfy the constraint

$$\underbrace{[A_1 J_1 + A_2 J_{12}^T, A_2 J_2 + A_1 J_{12}]}_{\mathcal{A}(q)} \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} = 0. \quad (8)$$

Thus, a path is said to be admissible if it satisfies constraint (8), or equivalently, its velocity vector lies in the kernel of $\mathcal{A}(q)$.

We can compute a basis spanning the kernel of $\mathcal{A}(q)$. First, notice that what a cat has control over in the air is its hinge velocity (or the relative velocity between the upper body and the lower body), which can be treated as a pseudo-control input. Using the relation

$$A_2 = A_1 A$$

the velocity of body 2 can be expressed in terms of the velocity of body 1 and the hinge velocity, $\hat{w} = A^T \dot{A}$:

$$w_2 = A^T w_1 + \hat{w}. \quad (9)$$

Applying Eq. (9) to Eq. (8) yields

$$\begin{cases} I_l A_1 w_1 = -(A_2 J_2 + A_1 J_{12}) w \\ I_l A_2 w_2 = (A_1 J_1 + A_2 J_{12}^T) A w \end{cases} \quad (10)$$

where

$$I_l = A_1 J_1 A_1^T + A_2 J_2 A_2^T + A_2 J_{12}^T A_1^T + A_1 J_{12} A_2^T$$

is the locked-body inertia tensor of the system at shape A .

Introduce Cayley parameters $\alpha \in \Re^3$ for the orientation of body 1, i.e., let $\alpha_0 = \frac{1}{1+\|\alpha\|^2}$,

$$A_1 = \alpha_0 \begin{bmatrix} 1 + \alpha_1^2 - \alpha_2^2 - \alpha_3^2 & 2(\alpha_1 \alpha_2 - \alpha_3) & 2(\alpha_1 \alpha_3 + \alpha_2) \\ 2(\alpha_1 \alpha_2 + \alpha_3) & 1 - \alpha_1^2 + \alpha_2^2 - \alpha_3^2 & 2(\alpha_2 \alpha_3 - \alpha_1) \\ 2(\alpha_1 \alpha_3 - \alpha_2) & 2(\alpha_2 \alpha_3 + \alpha_1) & 1 - \alpha_1^2 - \alpha_2^2 + \alpha_3^2 \end{bmatrix}$$

Thus, w_1 is related to $\dot{\alpha}$ through a locally nonsingular Jacobian matrix $U(\alpha) \in \Re^{3 \times 3}$

$$w_1 = U(\alpha) \dot{\alpha}.$$

Similarly, we define Cayley parameters $\beta \in \Re^3$ for body 2. If we let

$$u \triangleq \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix}$$

denote the three components of pseudo-control input, then Eq. (10) can be written in the form

$$B_1(q)\dot{q} = B_2(q)u \quad (11)$$

where $q = (\alpha, \beta) \in \mathbb{R}^6$ are local coordinates of the configuration space,

$$B_1(q) = \begin{bmatrix} I_l A_1 U(\alpha) & 0 \\ 0 & I_l A_2 U(\beta) \end{bmatrix}, \quad B_2(q) = \begin{bmatrix} -(A_2 J_2 + A_1 J_{12}) \\ (A_1 J_1 + A_2 J_{12}^T) A \end{bmatrix}. \quad (12)$$

Since $B_1(q)$ is nonsingular, Eq. (11) can also be written in the form

$$\dot{q} = B(q)u, \quad q \in \mathbb{R}^6, \quad u \in \mathbb{R}^3 \quad (13)$$

where $B(q) = B_1(q)^{-1} B_2(q)$ defines a 3-dimensional distribution on the configuration space, and the columns of $B(q)$ form a basis for the null space of $\mathcal{A}(q) \cdot \text{Diag}(U(\alpha), U(\beta))$.

From Eq. (13) motion of a falling cat is described in the form of a nonlinear control system, with 6 states and 3 control inputs. A falling cat is interested in solving the following problem.

Problem 2.1 Given two configurations $q_0 = (\alpha_0, \beta_0), q_f = (\alpha_f, \beta_f) \in \mathbb{R}^6$, find control input $u(t) \in \mathbb{R}^3, t \in [0, T]$, of minimal cost, such that the solution of Eq. (11), or Eq. (19), starting from q_0 ends at q_f at time T .

Here, a choice of cost functional would be the L_2 -norm of u ,

$$\|u\|_2^2 = \int_0^T \langle u(t), u(t) \rangle dt.$$

2.2 A Universal-Joint Model

A ball-in-socket joint model permits twisting motion of a cat. We can prevent twisting by replacing the ball-in-socket joint by a universal joint.

Under this new model, the system for rotational motion alone has 5 degrees-of-freedom: 3 for the orientation of body-1 and 2 for the joint angles of the universal joint. To obtain a nonlinear control system description of the system, notice that the shape of the system is given by

$$A = \begin{bmatrix} \cos \theta_1 & 0 & \sin \theta_1 \\ 0 & 1 & 0 \\ -\sin \theta_1 & 0 & \cos \theta_1 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 \\ \cos \theta_2 & 0 & \sin \theta_2 \\ \sin \theta_2 & 0 & -\cos \theta_2 \end{bmatrix} \quad (14)$$

where θ_1, θ_2 are the angles of the universal joint.

From Eq. (14) the hinge velocity of the system is

$$w = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \dot{\theta}_2 + \begin{bmatrix} \cos \theta_2 \\ 0 \\ \sin \theta_2 \end{bmatrix} \dot{\theta}_1 \triangleq b_1 u_1 + b_2 u_2 \quad (15)$$

where $u_1 = \dot{\theta}_1$ and $u_2 = \dot{\theta}_2$.

Multiply the first equation of (10) by A_1^T and apply Eq. (15) to the result we have

$$\tilde{I}_l U(\alpha) \dot{\alpha} = -(AJ_2 + J_{12})(b_1 u_1 + b_2 u_2). \quad (16)$$

where $J_{12} = \epsilon \hat{s}_1 A \hat{s}_2$ and

$$\tilde{I}_l = (J_1 + AJ_2A^T + AJ_{12}^T + J_{12}A^T)$$

is the locked-body inertia of the system from body-1.

Using Eq. (16) and the definitions of θ_1 and θ_2 , motion of a falling cat with universal-joint model is described by a nonlinear-control system of the form

$$B_1(q)\dot{q} = B_2(q)u, \quad (17)$$

where $u \in \Re^2$ is the control input, $q = (\theta_1, \theta_2, \alpha^T)^T \in \Re^5$ are the local coordinates of the configuration space,

$$B_1(q) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \tilde{I}_l U(\alpha) \end{bmatrix}$$

and

$$B_2(q) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ -(AJ_2 + J_{12})b_1 & -(AJ_2 + J_{12})b_2 \end{bmatrix}.$$

Again, $B_1(q) \in \Re^{5 \times 5}$ is nonsingular and Eq. (17) can be rewritten in the form

$$\dot{q} = B(q)u \quad (18)$$

where $B(q) = B_1(q)^{-1}B_2(q)$ defines a 2-dimensional distribution on the configuration space.

Problem 2.2 Consider system (18). Let $q_0 = (\theta_1^0, \theta_2^0, \alpha^0)^T$ and $q_f = (\theta_1^f, \theta_2^f, \alpha^f)^T$ be two given configurations. Find control input $u \in \Re^2$ of optimal cost, linking q_0 to q_f .

We will call both Problem (2.1) and (2.2) Nonholonomic Motion Planning (NMP) problems as the angular momentum conservation equation (8) is non-integrable, or the distribution spanned in local coordinates by the columns of $B(q)$ is nonholonomic ([VG88], [Mon89], [FGL91] and [Bro81]). These problems are typical of a class of motion planning problems one runs into in the study of dexterous manipulation by multifingered robotic hands and control of space robots (see [FGL91] for more details).

3 The Basis Algorithm for Optimal NMP

Following the lines of Ritz approximation theory ([CH53] and [BB91]) we develop in this section a *Basis Algorithm* for computing near-optimal solutions of a NMP problem. We can assume without loss of generality that the problem has been formulated in the form

$$\dot{x} = B(x)u, \quad x_0, x_f \in \Re^n \quad (19)$$

where $x \in \Re^n$ is the configuration variable, $u \in \Re^m$ the control input, $B(x) \in \Re^{n \times m}$ a regular m -dimensional distribution and the

cost function to be minimized is

$$J = \int_0^T \langle u, u \rangle dt.$$

We assume that the system is controllable¹ and thus there exists a solution $u^* \in L_2([0, T])$ for the problem. Here, $L_2([0, T])$ denotes the Hilbert space of measurable vector-valued functions of the form $u(t) = (u_1(t), \dots, u_m(t))^T, t \in [0, T]$. The inner product of two functions $f, g \in L_2([0, T])$ is given by

$$\langle f, g \rangle = \int_0^T \langle f(t), g(t) \rangle dt$$

and the norm, $\|f\|$, of f by $\|f\|^2 = \langle f, f \rangle$.

Let us rescale time so that $T = 2\pi$ and denote by $\{\mathbf{e}_i\}_{i=1}^\infty$ an orthonormal basis for $L_2([0, T])$. A well-known example of such a basis is the Fourier basis and another example is the one constructed from the polynomial basis using the Gram-Schmidt orthogonalization process. In terms of an orthonormal basis $\{\mathbf{e}_i\}_{i=1}^\infty$, a function $u \in L_2([0, T])$ can be expressed as

$$u = \sum_{i=1}^{\infty} \alpha_i \mathbf{e}_i; \quad (20)$$

for some sequence $\alpha = (\alpha_1, \alpha_2, \dots) \in l_2$, the ∞ -dimensional Hilbert space, the objective function becomes

$$J = \int_0^T \langle u, u \rangle dt = \sum_{i=1}^{\infty} \alpha_i^2 \stackrel{\Delta}{=} \|\alpha\|^2$$

and the NMP problem can be rephrased as: *Given a NMP system of the form*

$$\dot{x} = B(x)u, \quad u = \sum_{i=1}^{\infty} \alpha_i \mathbf{e}_i(t), \quad x_0, x_f \in \Re^n, \quad (22)$$

find $\alpha \in l_2$ of minimum cost such that the solution of (22) starting from x_0 reaches x_f at $t = T$. In other words, a NMP problem is equivalent to a nonlinear optimization problem in an ∞ -dimensional space. Certainly, solving for the exact solution of such

¹In other words, the system satisfies the so-called Lie-Algebra Rank Condition (LARC): The Lie algebra generated by the columns of $B(q)$ has rank equal to the number of states (see [Isi85], [Bro76] and [HK77])

a problem, if one exists, poses considerable computational difficulties. The idea of Ritz approximation theory is to approximate the solution using solutions of some finite-dimensional problems. As dimension of the finite-dimensional problems goes to infinity, the approximated solutions converge to the true solution.

The crucial problems here are, *(1) how do we introduce the finite-dimensional problems? and (2) how to efficiently solve these finite-dimensional problems?*

For the first part we will demonstrate, using solution of a simple example, that *due to nature of nonholonomic systems* it is reasonable to introduce finite-dimensional problems by restricting control input to the original system to the first few basis elements. For the second part, we will develop a numerical algorithm which requires only solution of a set of well-defined initial-value problems.

Example 3.1 Let us consider solution of Brockett's canonical problem ([Bro81] and [MS90]) in terms of the Fourier basis.

$$\begin{aligned}\dot{x}_1 &= u_1 \\ \dot{x}_2 &= u_2 \\ \dot{x}_3 &= -x_2 u_1 + x_1 u_2\end{aligned}\tag{23}$$

with boundary conditions

$$x_0 = (0, 0, 0)^T \text{ and } x_f = (0, 0, \delta)^T, \quad \delta \geq 0.$$

Since the boundary conditions for the first and second coordinates are the same, the control inputs in terms of the Fourier basis (note that $b_0 = 0$ and $\bar{b}_0 = 0$) have the form

$$\begin{aligned}u_1 &= \sum_{n=1}^{\infty} a_n \sin nt + \sum_{n=1}^{\infty} b_n \cos nt \\ u_2 &= \sum_{n=1}^{\infty} \bar{a}_n \sin nt + \sum_{n=1}^{\infty} \bar{b}_n \cos nt.\end{aligned}\tag{24}$$

Integrating the last equation of (23) and dividing the result by 2π yields

$$\begin{aligned}\frac{\delta}{2\pi} = \frac{x_3(2\pi) - x_3(0)}{2\pi} &= \frac{1}{2\pi} \int_0^{2\pi} (-x_2 u_1 + x_1 u_2) dt \\ &= \sum_{n=1}^{\infty} \frac{1}{n} (\bar{a}_n b_n - \bar{b}_n a_n).\end{aligned}\quad (25)$$

The objective is to minimize the cost

$$\|\alpha\|^2 = \sum_{n=1}^{\infty} a_n^2 + b_n^2 + \bar{a}_n^2 + \bar{b}_n^2 \quad (26)$$

subject to constraint equation (25).

To compute the optimal solutions of the system, let us introduce the following lemma.

Lemma 3.1 (a) Let $Z_0 \in \mathbb{R}^3$ be given. Then, the set of solutions $X, Y \in \mathbb{R}^3$ which satisfies the constraint

$$X \times Y = Z_0$$

and minimizes the cost

$$\|X\|^2 + \|Y\|^2$$

has the form

$$\{(X, Y) : X \cdot Y = 0, X \cdot Z_0 = 0, Y \cdot Z_0 = 0, \|X\|^2 = \|Y\|^2 = \|Z_0\|\} \quad (27)$$

and the optimal cost is $2\|Z_0\|$.

(b) Let $Z_0 \in \mathbb{R}^3$ be given. Then, the set of solutions $\{X_i, Y_i \in \mathbb{R}^3\}_{i=1}^{\infty}$ which satisfies the constraint

$$Z_0 = \sum_{i=1}^{\infty} c_i (X_i \times Y_i), \text{ for } c_1 = 1, c_i < c_1, i \geq 2 \quad (28)$$

and minimizes the cost

$$\sum_{i=1}^{\infty} \|X_i\|^2 + \|Y_i\|^2$$

has the form

$$\{X_i = Y_i = 0 \text{ for } i \geq 2\}$$

and (X_1, Y_1) solves part (a) of the lemma.

Proof. Part (a) can be proved using either the Lagrangian multiplier's technique or by observing the following inequalities.

First,

$$\|Z_0\| = \|X\| \|Y\| \sin \theta \leq \|X\| \|Y\|$$

where θ is the angle between X and Y . On the other hand, we have that

$$\|X\|^2 + \|Y\|^2 \geq 2\|X\| \|Y\|.$$

Thus, the cost is minimal if the conditions of part (a) are satisfied. Note that X and Y lie on the plane perpendicular to Z_0 but are otherwise arbitrary. *The solutions are not unique.*

To show part (b), let $Z_n = c_n(X_n \times Y_n)$ and we have that

$$\|Z_0\| = \left\| \sum_{n=1}^{\infty} Z_n \right\| \leq \sum_{n=1}^{\infty} \|Z_n\|.$$

The cost is bounded below by

$$\begin{aligned} \sum_{n=1}^{\infty} \|X_n\|^2 + \|Y_n\|^2 &\geq 2 \sum_{n=1}^{\infty} \|X_n\| \|Y_n\| \\ &\geq 2 \sum_{n=1}^{\infty} d_n \|Z_n\|, \quad d_n = \frac{1}{c_n} \geq 1 \\ &= 2 \left(\sum_{n=1}^{\infty} \|Z_n\| + \sum_{n=2}^{\infty} (d_n - 1) \|Z_n\| \right) \\ &\geq 2\|Z_0\|. \end{aligned}$$

The last inequality can be made an equality by choosing $Z_n = 0, n \geq 2$ and Z_1 as in part (a). Thus the minimal cost results from a choice satisfying the conditions of part (b). \square

We now return to Brockett's canonical problem. First, let us define by

$$Z_0 = \begin{bmatrix} 0 \\ 0 \\ \delta/2\pi \end{bmatrix}, \quad X_n = \begin{bmatrix} \bar{a}_n \\ \bar{b}_n \\ 0 \end{bmatrix}, \quad \text{and} \quad Y_n = \begin{bmatrix} a_n \\ b_n \\ 0 \end{bmatrix}.$$

Then, constraint equation (25) is equivalent to

$$Z_0 = \sum_{n=1}^{\infty} \frac{1}{n} (X_n \times Y_n)$$

and the objective function (26) becomes

$$\sum_{n=1}^{\infty} \|X_n\|^2 + \|Y_n\|^2.$$

Applying part (b) of the lemma we conclude that

$$X_n = Y_n = 0, \text{ for } n \geq 2,$$

and from part (a) we have that the set of optimal controls is given by

$$\begin{aligned} u_1(t) &= \left[\sqrt{\delta/2\pi}, 0 \right] R(\phi)^T \begin{bmatrix} \sin t \\ \cos t \end{bmatrix} \\ u_2(t) &= \left[0, \sqrt{\delta/2\pi} \right] R(\phi)^T \begin{bmatrix} \sin t \\ \cos t \end{bmatrix} \end{aligned}$$

where

$$R(\phi) = \begin{bmatrix} \cos \phi & -\sin \phi \\ \sin \phi & \cos \phi \end{bmatrix}, \quad \phi \in [0, 2\pi],$$

is the rotation matrix on the plane. Observe that the optimal solutions are not unique in this case. \square

Notice that in the above example the first few harmonics of the Fourier basis play more important roles than the high frequency components. Thus, it is not unreasonable to restrict the control to a finite-dimensional space spanned by the *first* N basis elements so as to avoid solving the ∞ -dimensional problem. Let

$$u = \sum_{i=1}^N \alpha_i e_i(t) \tag{29}$$

where $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_N)^T \in \mathbb{R}^N$ is to be determined, we optimize the following functional

$$J(\alpha, \gamma) = \sum_{i=1}^N \alpha_i^2 + \gamma \|x(1) - x_f\|^2. \tag{30}$$

Here, $x(1) \in \mathbb{R}^n$ is the solution of the system

$$\dot{x} = B(x)u \quad (31)$$

at $T = 1$, with initial condition x_0 and control input given by Eq. (29).

Clearly, the terminal condition, $x(1) \in \mathbb{R}^n$, is a function of the $\alpha \in \mathbb{R}^N$ and we can associate to it a map

$$f : \mathbb{R}^N \longrightarrow \mathbb{R}^n$$

by $f(\alpha) = x(1)$. Thus, once γ is chosen, the cost given by Eq. (30) is a function of $\alpha \in \mathbb{R}^N$

$$J(\alpha) = \langle \alpha, \alpha \rangle + \gamma \|f(\alpha) - x_f\|^2. \quad (32)$$

Our problem now is to find $\alpha \in \mathbb{R}^N$ such that the cost function in Eq. (32) is minimized. Based on Ritz approximation theory (see Section 3.2 and [FGL91]) we have that as N and γ tend to ∞ , solutions obtained by solving this finite-dimensional problem tend to solutions of the original system. In other words, this approach will give us *approximate optimal solutions* of a NMP problem.

If the function $f(\alpha)$ is known and $J(\cdot)$ is convex, an efficient approach to minimize $J(\alpha)$ is by *quadratic programming*. Let's review the basic idea of quadratic programming using a simple example.

Example 3.2 Give an algorithm which minimizes the following function

$$I(x) = \langle Qx, x \rangle - 2\langle x_0, x \rangle, \quad x \in \mathbb{R}^n$$

where $Q \in \mathbb{R}^{n \times n}$ is symmetric and positive definite and $x_0 \in \mathbb{R}^n$.

Solution. Computing the Taylor series expansion of the function about a point y yields

$$\begin{aligned} I(y + \delta) &= I(y) + \langle \frac{\partial I}{\partial x}, \delta \rangle + \frac{1}{2} \langle \frac{\partial^2 I}{\partial x^2} \delta, \delta \rangle + O(\|\delta\|^3) \\ &= I(y) + 2\langle Qy - x_0, \delta \rangle + \langle Q\delta, \delta \rangle. \end{aligned}$$

From which we obtain

$$\frac{\partial I}{\partial x}|_y = 2(Qy - x_0), \text{ and } \frac{\partial^2 I}{\partial x^2}|_y = 2Q.$$

If \bar{x} is a minimum, then it is necessarily true that the differential of I evaluated at \bar{x} be zero

$$g(y) \triangleq \frac{\partial I}{\partial x}|_y : \Re^n \longrightarrow \Re^n.$$

It is also sufficient when the cost function, $I(\cdot)$, is convex, so minimizing $I(\cdot)$ is equivalent to solving for the root(s) of the function

$$g(y) = 0.$$

Solution of the latter problem can be computed using the popular Newton's algorithm, or variations of Newton's algorithm.

$$\begin{aligned} x_{n+1} &= x_n - \left(\frac{\partial g}{\partial x} \right)^{-1} g(x_n) = x_n - \left(\frac{\partial^2 I}{\partial x^2} \right)^{-1} \frac{\partial I}{\partial x} \\ &= x_n - Q^{-1}(Qx_n - x_0). \end{aligned} \quad (33)$$

Since our function $I(\cdot)$ is quadratic and globally convex, Newton's algorithm converges in a single step to the global minimum $x_{n+1} = Q^{-1}x_0$. \square

Since Newton's algorithm is so powerful we can still apply the algorithm or one of its many variations to the minimization of $J(\cdot)$ even when $J(\cdot)$ is not globally convex. The latter is a very strong condition to assume for a nonlinear function.

Computing the Taylor series expansion of $J(\cdot)$ about a point α_n yields

$$\begin{aligned} J(\alpha_n + \delta) &= J(\alpha_n) + \langle \frac{\partial J}{\partial \alpha}, \delta \rangle + \frac{1}{2} \langle \frac{\partial^2 J}{\partial \alpha^2} \delta, \delta \rangle + O(\|\delta\|^3) \\ &= J(\alpha_n) + 2\langle \gamma A^T(f(\alpha_n) - x_f) + \alpha_n, \delta \rangle \\ &\quad + \langle (I + \gamma A^T A + \gamma \sum_{i=1}^n (f_i(\alpha_n) - x_i^f) H_i)) \delta, \delta \rangle + O(\|\delta\|^3) \end{aligned} \quad (34)$$

where

$$A = \frac{\partial f}{\partial \alpha}|_{\alpha_n} \in \Re^{n \times N} \text{ and } H_i = \frac{\partial^2 f_i}{\partial \alpha^2}, \quad i = 1, \dots, n$$

are, respectively, the Jacobian of f and the Hessians of the component functions of f , i.e., $f = (f_1, \dots, f_n)^T$.

From Eq. (34) we have that

$$\frac{\partial J}{\partial \alpha}|_{\alpha_n} = 2 \left(\alpha_n + \gamma A^T(f(\alpha_n) - x_f) \right) \quad (35)$$

and

$$\frac{\partial^2 J}{\partial \alpha^2}|_{\alpha_n} = 2 \left(I + \gamma A^T A + \gamma \sum_{i=1}^n (f_i(\alpha_n) - x_i^f) H_i \right). \quad (36)$$

Since $(I + \gamma A^T A)$ is positive definite already and it is also difficult to compute the Hessians of the component functions, we use the following modified Newton's algorithm to update α .

$$\alpha_{n+1} = \alpha_n - \mu \left[\sigma I + A^T A \right]^{-1} \left[\sigma \alpha_n + A^T(f(\alpha_n) - x_f) \right] \quad (37)$$

where $\sigma = 1/\gamma$ and $0 < \mu < 1$ is a parameter.

Remark 3.1 The choice of $\mu \in (0, 1)$ in Eq. (37) is very important in order for the algorithm to converge. In our simulation experiments, a small μ is used when α is far from its optimal solution, and is gradually increased to 1 when α comes close to its solution. As a result, our algorithm is convergent for almost all examples we have tested so far. This phenomena has the following interpretations. First, let's define

$$\psi(\mu) = J(\alpha_{n+1}(\alpha_n, \mu)).$$

We claim that

$$\frac{\partial \psi(\mu)}{\partial \mu}|_{\mu=0} < 0$$

as long as $\frac{\partial J(\alpha)}{\partial \alpha}|_{\alpha_n} \neq 0$. In other words, when α is far from its solution, by choosing μ small we can always make the cost function $J(\cdot)$ decrease. To see this, denote by

$$v \triangleq (\sigma I + A^T A)^{-1} [\sigma \alpha_n + A^T(f(\alpha_n) - x_f)]$$

and evaluate the derivative of ψ at 0 we obtain

$$\begin{aligned}\frac{\partial \psi}{\partial \mu} \Big|_{\mu=0} &= -2\langle v, (\alpha_n + \gamma A^T(f(\alpha_n) - x_f)) \rangle \\ &= -\frac{1}{2} \langle (I + \gamma A^T A)^{-1} \frac{\partial J}{\partial \alpha}, \frac{\partial J}{\partial \alpha} \rangle.\end{aligned}$$

Since $(\sigma I + A^T A)$ is positive definite and $\frac{\partial J}{\partial \alpha} \neq 0$ we conclude that $\frac{\partial \psi}{\partial \mu} \Big|_{\mu=0} < 0$. Note also that when $\mu \leq 1$, the algorithm is similar to quadratic programming, and when $\mu \ll 1$, it is similar to the gradient method. \square

Remark 3.2 Eq. 37 can be modified to reduce computation time when the penalty coefficient γ is large, i.e. when $\sigma \ll 1$. Instead of computing the inverse of the $N \times N$ matrix, $[\sigma I + A^T A]$, the computation of an $n \times n$ matrix, $[AA^T]$, suffices. This can be seen by rewriting the previous equation as follows :

$$\alpha_{n+1} = \alpha_n - \mu(C\alpha_n + D(f(\alpha_n) - x_f))$$

where $C = \sigma[\sigma I + A^T A]^{-1}$ is $N \times N$ and $D = [\sigma I + A^T A]^{-1} A^T$ is $N \times n$. We replace C with the projector on the nullspace of A , i.e. with $[I - A^T[AA^T]^{-1}A]$ and D with the pseudoinverse of A , i.e. with $A^T[AA^T]^{-1}$, where both replacements can be computed via the inversion of an $n \times n$ matrix, $[AA^T]$. The validity of these replacements as $\sigma \rightarrow 0$ is easily seen by considering the representation of transformation A in a basis where it has the form

$$\begin{bmatrix} a & 0 \end{bmatrix},$$

for invertible $n \times n$ matrix a . Note that A (and hence a) is assumed to have full row rank, which is generically the case.

Then the representation of C is

$$\begin{bmatrix} \sigma[a^T a + \sigma I]^{-1} & 0 \\ 0 & I \end{bmatrix},$$

which tends to the projector on the nullspace of A as $\sigma \rightarrow 0$. Similarly the representation of D in this basis is

$$\begin{bmatrix} [a^T a + \sigma I]^{-1} a^T \\ 0 \end{bmatrix},$$

which tends to the pseudoinverse of A ,

$$\begin{bmatrix} a^{-1} \\ 0 \end{bmatrix}.$$

□

It has been shown in [FGL91] that there exists $\sigma_0 > 0$ such that the cost function

$$J(\alpha, \sigma) = \sigma \langle \alpha, \alpha \rangle + \langle f(\alpha) - x_f, f(\alpha) - x_f \rangle$$

is locally convex near a solution of the problem for any $0 < \sigma \leq \sigma_0$. The important consequence of this result is the superlinear ([Lue84]) convergence of Eq. (37) if the starting point is close enough to the global minimum. If the starting point is far away from the global minimum, the algorithm can get stuck in some local minimum, although our numerical experience did not give such an example. We are now working on an approach which helps in avoiding local minima.

Before we attempt to apply Eq. (37) to a NMP problem, we should be aware that the function $f(\alpha)$ is in general not known. Consequently, A is not available. We shall now develop a numerical approach for computing the function $f(\alpha)$ and its Jacobian.

3.1 Computation of Jacobians

In order to implement Eq. (37), we need to compute the function $f(\alpha_n)$ and its Jacobian, $A = \frac{\partial f}{\partial \alpha}|_{\alpha_n} \in \Re^{n \times N}$. Here, we present a numerical algorithm, via the integration of two differential equations, to this problem.

Let us define by Φ the $n \times N$ matrix whose columns are the basis elements, $\{\mathbf{e}_i(t)\}_{i=1}^N, t \in [0, T]$,

$$\Phi = [\mathbf{e}_1(t), \mathbf{e}_2(t), \dots, \mathbf{e}_N(t)] \quad (38)$$

and rewrite the NM system as

$$\dot{x} = B(x)u = B(x)\Phi\alpha, \quad \alpha \in \mathbb{R}^N.$$

Clearly, A is given by evaluating the matrix function

$$Y(t) = \frac{\partial x(t)}{\partial \alpha}$$

at $t = T$. Also note that

$$Y(0) = \lim_{t \rightarrow 0} Y(t) = 0.$$

The differential equation for $Y(t)$ is given by

$$\begin{aligned} \dot{Y} &= \frac{\partial}{\partial t} \frac{\partial x}{\partial \alpha} = \frac{\partial}{\partial \alpha} \frac{\partial x}{\partial t} = \frac{\partial}{\partial \alpha} \dot{x} \\ &= \frac{\partial}{\partial \alpha} \left(\sum_{i=1}^m B^i u_i \right) = \sum_{i=1}^m \left(\frac{\partial B^i}{\partial \alpha} u_i + B^i \frac{\partial u_i}{\partial \alpha} \right) \\ &= \sum_{i=1}^m \frac{\partial B^i}{\partial \alpha} u_i + B\Phi = \sum_{i=1}^m \left(\frac{\partial B^i}{\partial x} \frac{\partial x}{\partial \alpha} \right) u_i + B\Phi \\ &= \sum_{i=1}^m \left(\frac{\partial B^i}{\partial x} Y \right) u_i + B\Phi = \left(\sum_{i=1}^m \frac{\partial B^i}{\partial x} u_i \right) Y + B\Phi. \end{aligned}$$

Jacobians of the columns of B are not difficult to compute if the form of B is known.

Thus, at each time step we have to integrate the following set of differential equations from $t = 0$ to $t = T$ in order to update α .

$$\begin{cases} \dot{x} = B(x)\Phi\alpha_n, \quad x(0) = x_0 \\ \dot{Y} = \left(\sum_{i=1}^m \frac{\partial B^i}{\partial x} u_i \right) Y + B\Phi, \quad Y(0) = 0. \end{cases} \quad (39)$$

We set $f(\alpha_n) = x(T)$ and $A = Y(T)$.

Let's summarize our algorithm given by the Basis Approach.

Algorithm 3.1 (Basis Algorithm)

Input: 1. Initial and final configurations: x_0 and $x_f \in \mathbb{R}^n$
 2. $B(x) \in \mathbb{R}^{n \times m}$.

Output: Control input $u(t) \in \mathbb{R}^m, t \in [0, T]$, linking x_0 to x_f .

Step 0: 1. Choose an orthonormal basis and retain the first N elements $\{\mathbf{e}_i(t)\}_{i=1}^N$. Define Φ as in Eq. (38).
 2. Initialize $\alpha_0 \neq 0$ by some random process.

Step 1: 1. Choose $\gamma > 0$ and $\mu > 0$.
 2. Solve the set of differential equations given by (39).
 3. Set

$$f(\alpha_n) = x(T) \text{ and } A = Y(T).$$

$$\alpha_{n+1} = \alpha_n - \mu [\sigma I + A^T A]^{-1} [\sigma \alpha_n + A^T (f(\alpha_n) - x_f)]$$

Step 2: Examine $x(T)$ and cost $J_{N,\gamma}$. If the results are not satisfactory, repeat Step 1, otherwise exit.

Remark 3.3 According to our experience with the algorithm, initially γ and μ are chosen to be small. When α comes close to its solution, i.e., $x(T)$ is near x_f and the cost ceases to decrease, increase γ and μ . It is also important that $\alpha_0 \neq 0$, otherwise the program will not initiate.

3.2 Correctness of the Basis Algorithm

In this subsection, we state a result about correctness of the *Basis Algorithm*. let $S^* = (x^*(t), u^*(t)), t \in [0, T]$, be the set of optimal

solutions of the original system²

$$P_* : \quad \dot{x} = B(x)u, \quad x_0, x_f \in \Re^n, \quad u = \sum_{i=1}^{\infty} \alpha_i \mathbf{e}_i(t) \quad (40)$$

with cost function

$$J(u) = \int_0^1 \langle u(t), u(t) \rangle dt = \sum_{i=1}^{\infty} \alpha_i^2$$

where $\{\mathbf{e}_i(t)\}_{i=1}^{\infty}, t \in [0, 1]$ is an orthonormal basis for $L_2([0, 1])$, the Hilbert space of measurable vector-valued functions, and $S_{N,\gamma} = (x^N(t), u^N(t))$ the set of optimal solutions of the *approximated system*

$$P_{N,\gamma} : \quad \dot{x} = B(x)u^N(t), \quad x_0, x_f \in \Re^n, \quad u^N(t) = \sum_{i=1}^N \alpha_i \mathbf{e}_i(t) \quad (41)$$

with cost function

$$J_{N,\gamma}(u^N) = \sum_{i=1}^N \alpha_i^2 + \gamma \|x^N(1) - x_f\|^2,$$

then, under some appropriately defined *measures*, the solution set of the *approximated system* converges to the solution set of the *original system*.

We need the following assumption about the system under consideration.

Assumption 3.1 *There exists a function $\phi(\delta), \delta \geq 0$, for system (40) such that if $\|u\|_2 \leq \delta$, then $\|x\|_{C[0,1]} \leq \phi(\delta) < \infty$. In other words, the system is bounded input and bounded state (BIBS) stable.*

It was shown in ([GL90]) that if there exist two constants, β_0 and β_1 such that $\|B(x)\| \leq \beta_0 \|x\| + \beta_1$, the so-called *linear growth* condition, then for any $u \in L_2[0, 1]$ applied to the system there

²In other words, $u^*(t)$ is control input to system P_* and $x^*(t)$ the corresponding state.

exists a unique x belonging to $C_{[0,1]}$ and Assumption (3.1) holds. In other words, both the solution set $S_{N,\gamma}$ and S_* are in the Banach space $C_{[0,1]} \oplus L_2[0, 1]$.

Let $X, Y \subset C_{[0,1]} \oplus L_2[0, 1]$. We define the distance, $d(X, Y)$, of X to Y as follows

$$d(X, Y) = \sup_{(x,u) \in X} \inf_{(\tilde{x},\tilde{u}) \in Y} (\|x - \tilde{x}\|_{C_{[0,1]}} + \|u - \tilde{u}\|_2).$$

Using this distance measure, we now have

Theorem 3.1 *Let $S_* = (x^*, u^*) \subset C_{[0,1]} \oplus L_2[0, 1]$ be the set of optimal solutions of system P_* with optimal cost J_* , and $S_{N,\gamma} = (x^N, u^N) \subset C_{[0,1]} \oplus L_2[0, 1]$ be the optimal solution set of $P_{N,\gamma}$, with optimal cost $J_{N,\gamma}$. Then, $S_{N,\gamma}$ converge to S_* in the sense that*

$$\lim_{\gamma \rightarrow \infty} \lim_{N \rightarrow \infty} d(S_{N,\gamma}, S_*) = 0$$

and $J_{N,\gamma}$ converges to J_* in the sense that

$$\lim_{\gamma \rightarrow \infty} \lim_{N \rightarrow \infty} J_{N,\gamma} = J_*.$$

Proof of the theorem is rather technical and can be found in [FGL91], or inferred from [CH53].

4 Simulation Results

To test the Basis Algorithm, we have performed simulation experiments on several NMP systems, including a unicycle moving on a plane subject to rolling constraint, parking a front-wheel drive cart, attitude control of a satellite using two rotors, reorientation of a space station using a Puma manipulator and, of course, our falling cat systems. In this section, we will describe simulation results on the falling cat system. The reader is referred to [FGL91] for simulation results on the other systems.

4.1 The Universal-Joint Model

The falling cat, when modeled as two rigid bodies coupled by a universal joint, has two control inputs and the configuration space has dimension 5. The equation describing the system is given in Eq. (17), which is of the form

$$B_1(q)\dot{q} = B_2(q)u \quad (42)$$

where $q = (\theta_1, \theta_2, \alpha_1, \alpha_2, \alpha_3)^T$, $B_1(q) \in \mathbb{R}^{5 \times 5}$ is invertible, and $u(t) = \Phi(t)\gamma$, $\gamma \in \mathbb{R}^N$, Φ is the $n \times N$ matrix whose columns are the basis elements, $\{\mathbf{e}_i(t)\}_{i=1}^N$, $t \in [0, 2\pi]$.

To proceed with simulation using the *Basis Algorithm* we require Jacobians, $\frac{\partial q}{\partial \gamma}|_{2\pi} \triangleq Y(2\pi)$. We have

$$\dot{Y} = \frac{\partial}{\partial t} \frac{\partial q}{\partial \gamma} = \frac{\partial}{\partial \gamma} \frac{\partial q}{\partial t} = \frac{\partial \dot{q}}{\partial \gamma}.$$

Since the symbolic inversion of B_1 is difficult, we cannot directly use the procedure of Section 3.1 to obtain \dot{Y} . Instead, we obtain the differential equation to compute the required Jacobians, starting from the equations of the (general) form in Eq.(42), by differentiating both sides of (42) :

$$\begin{aligned} \frac{\partial}{\partial \gamma} B_1(q)\dot{q} &= \frac{\partial}{\partial \gamma} B_2(q)u \\ \Rightarrow \frac{\partial}{\partial \gamma} \sum_{i=1}^n B_1^i \dot{q}_i &= \frac{\partial}{\partial \gamma} \sum_{i=1}^m B_2^i u_i \\ \Rightarrow \sum_{i=1}^n \dot{q}_i \frac{\partial B_1^i}{\partial \gamma} + B_1 \frac{\partial \dot{q}}{\partial \gamma} &= \sum_{i=1}^m u_i \frac{\partial B_2^i}{\partial \gamma} + B_2 \frac{\partial u}{\partial \gamma} \\ \Rightarrow \sum_{i=1}^n \dot{q}_i \frac{\partial B_1^i}{\partial q} \frac{\partial q}{\partial \gamma} + B_1 \dot{Y} &= \sum_{i=1}^m u_i \frac{\partial B_2^i}{\partial q} \frac{\partial q}{\partial \gamma} + B_2 \Phi \\ \Rightarrow \left(\sum_{i=1}^n \dot{q}_i \frac{\partial B_1^i}{\partial q} \right) \frac{\partial q}{\partial \gamma} + B_1 \dot{Y} &= \left(\sum_{i=1}^m u_i \frac{\partial B_2^i}{\partial q} \right) \frac{\partial q}{\partial \gamma} + B_2 \Phi \\ \Rightarrow \left(\frac{\partial B_1}{\partial q} \otimes \dot{q} \right) Y + B_1 \dot{Y} &= \left(\frac{\partial B_2}{\partial q} \otimes u \right) Y + B_2 \Phi \\ \Rightarrow B_1 \dot{Y} &= \left(\frac{\partial B_2}{\partial q} \otimes u - \frac{\partial B_1}{\partial q} \otimes \dot{q} \right) Y + B_2 \Phi. \end{aligned} \quad (43)$$

Here \otimes denotes the special multiplication of a vector with the derivative of a matrix with respect to a vector variable.

Thus, we have to integrate the following set of differential equations from $t = 0$ to $t = 2\pi$ at each iteration of the *Basis Algorithm*³, where γ is the value during the current iteration, of the Fourier representation of the point in control space.

$$\begin{cases} \dot{q} = B_1^{-1}(q)B_2(q)\Phi\gamma, & q(0) = q_0, \\ \dot{Y} = B_1^{-1}\left(\frac{\partial B_2}{\partial q} \otimes u - \frac{\partial B_1}{\partial q} \otimes \dot{q}\right)Y + B_1^{-1}B_2\Phi, & Y(0) = 0. \end{cases} \quad (44)$$

Integrating Eq. (44) from time 0 to time 2π to obtain values of $q(2\pi)$ and $\frac{\partial q(2\pi)}{\partial \gamma}$, and using the simulation procedure described in the earlier sections, the *Basis Algorithm* was tried on the following examples.

The initial and final configurations are

$$q_0 = \begin{bmatrix} -1.14159 \\ 0.0 \\ -0.999997 \\ 1.83048 \\ 0.546302 \end{bmatrix}, \quad q_f = \begin{bmatrix} -1.14159 \\ 0 \\ -1 \\ -0.546305 \\ -1.83049 \end{bmatrix}.$$

If we draw the falling-cat on a vertical plane, then q_f is a rotation from q_0 about a horizontal axis by 180° (upside down configuration to landing configuration). Simulation results are presented using controls from two control subspaces, one of dimension 10 and the other a larger (containing the first) subspace of dimension 22.

(A) 10-D Control Space

In this program run, we look for optimal control in a 10 dimensional subspace of control space. The elements of a basis for the subspace are

$$\mathbf{e}_1 = \begin{bmatrix} 0.5 \\ 0 \end{bmatrix}, \quad \mathbf{e}_2 = \begin{bmatrix} \sin t \\ 0 \end{bmatrix}, \quad \mathbf{e}_3 = \begin{bmatrix} \cos t \\ 0 \end{bmatrix},$$

³If the DASSL package ([BCP89]) is available, then Eqs. (42) and (43) can be integrated in their present form, without inverting the $B_1(q)$ matrix.

$$\mathbf{e}_4 = \begin{bmatrix} \sin 2t \\ 0 \end{bmatrix}, \mathbf{e}_5 = \begin{bmatrix} \cos 2t \\ 0 \end{bmatrix}$$

and the elements obtained by permuting rows of the above.

In Figure 2, a sequence of 19 snapshots of the motion of the cat from the initial configuration to the final configuration is shown. Starting from the initial configuration at top-left, the snapshots are taken at equally spaced points in the time interval $[0, 2\pi]$ and are displayed in column major order. In the bottom-right snapshot, the desired and the actual final configuration are superimposed. As can be seen, the error is small.

Figure 3 shows plots of the optimal control inputs. Figure 4 shows the θ_1 and θ_2 components of the hinge optimal trajectory and the α_1 , α_2 and α_3 components of the optimal trajectory of the upper body linking q_0 to q_f .

(B) 22-D Control Subspace

In the example which was simulated, we look for optimal control in a 22 dimensional subspace of control space. The basis elements for the subspace are

$$\begin{aligned} \mathbf{e}_1 &= \begin{bmatrix} 0.5 \\ 0 \end{bmatrix}, \mathbf{e}_2 = \begin{bmatrix} \sin t \\ 0 \end{bmatrix}, \mathbf{e}_3 = \begin{bmatrix} \cos t \\ 0 \end{bmatrix}, \\ \mathbf{e}_4 &= \begin{bmatrix} \sin 2t \\ 0 \end{bmatrix}, \mathbf{e}_5 = \begin{bmatrix} \cos 2t \\ 0 \end{bmatrix}, \mathbf{e}_6 = \begin{bmatrix} \sin 3t \\ 0 \end{bmatrix}, \mathbf{e}_7 = \begin{bmatrix} \cos 3t \\ 0 \end{bmatrix}, \\ \mathbf{e}_8 &= \begin{bmatrix} \sin 4t \\ 0 \end{bmatrix}, \mathbf{e}_9 = \begin{bmatrix} \cos 4t \\ 0 \end{bmatrix}, \mathbf{e}_{10} = \begin{bmatrix} \sin 5t \\ 0 \end{bmatrix}, \mathbf{e}_{11} = \begin{bmatrix} \cos 5t \\ 0 \end{bmatrix} \end{aligned}$$

and the elements obtained by permuting rows of the above.

In Figure 5, a sequence of 19 snapshots of the motion of the cat from the initial configuration to the final configuration is shown. Starting from the initial configuration at top-left, the snapshots are taken at equally spaced points in the time interval $[0, 2\pi]$ and are displayed in column major order. In the bottom-right snapshot, the desired and the actual final configuration are superimposed. As can be seen, the error is small.

Figure 6 shows plots of the optimal control inputs. Figure 7 shows the θ_1 and θ_2 components of the hinge optimal trajectory and the α_1 , α_2 and α_3 components of the optimal trajectory of the upper body linking q_0 to q_f .

4.2 Ball-in-socket Joint

The equation describing the ball-in-socket joint model is also of the form

$$B_1(q)\dot{q} = B_2(q)u$$

where $q = (\alpha^T, \beta^T)^T \in \mathbb{R}^6$, $u \in \mathbb{R}^3$ and $B_1(q) \in \mathbb{R}^{6 \times 6}$, $B_2(q) \in \mathbb{R}^{6 \times 3}$ are given in Eq. (12).

The Jacobians are obtained in a way similar to the universal-joint model case, except that the results are much more complicated. The initial and final configurations in terms of the Cayley parameters are

$$q_0 = \begin{bmatrix} -0.298446 \\ 0.546302 \\ 0.546302 \\ 1 \\ -.546302 \\ -1.83 - 49 \end{bmatrix}, q_f = \begin{bmatrix} -5.50899 \\ -3.00959 \\ -1.83049 \\ 1.64414 \\ -3.00957 \\ 0.546302 \end{bmatrix}.$$

If we draw the falling-cat on a plane, then q_f is a rotation from q_0 about a horizontal axis by 180° (upside down configuration to landing configuration). We have used 21 basis elements in the simulation. These are

$$\mathbf{e}_1 = \begin{bmatrix} 0.5 \\ 0 \\ 0 \end{bmatrix}, \mathbf{e}_2 = \begin{bmatrix} \sin t \\ 0 \\ 0 \end{bmatrix}, \mathbf{e}_3 = \begin{bmatrix} \cos t \\ 0 \\ 0 \end{bmatrix}, \mathbf{e}_4 = \begin{bmatrix} \sin 2t \\ 0 \\ 0 \end{bmatrix}$$

$$\mathbf{e}_5 = \begin{bmatrix} \cos 2t \\ 0 \\ 0 \end{bmatrix}, \mathbf{e}_6 = \begin{bmatrix} \sin 3t \\ 0 \\ 0 \end{bmatrix}, \mathbf{e}_7 = \begin{bmatrix} \cos 3t \\ 0 \\ 0 \end{bmatrix}$$

and the remaining basis elements are obtained by permuting the rows of the above elements.

Figure 8 shows plots of the optimal control inputs. Figure 9 shows the a_1 , a_2 , a_3 components of the optimal trajectory of the upper body and the b_1 , b_2 and b_3 components of the optimal trajectory for the lower body, linking q_0 to q_f .

In Figure 10, a sequence of snapshots of the 3D cat from the initial configuration to the final configuration is shown. In the lowest snapshot in this figure, the desired and the actual final configuration are superimposed. As can be seen, the error is small. This figure can be compared with Figure 6 in ([KS69]), where photographs of the motion of a falling cat were overlaid on computer-drawn pictures.

5 Conclusion

In this paper, we studied optimal nonholonomic motion planning for a falling cat, or more specifically, a system of coupled rigid bodies subject to angular momentum conservation. First, utilizing the mechanics of the system we formulated the motion planning problem as a nonlinear control problem without drift. Consequently, some available tools from nonlinear control theory can be used. For example, by computing the Lie-algebra generated by the vector fields of $B(q)$, one can determine the existence of admissible path between two given configurations ([Cho40] and [Isi85]) using Chow's theorem.

Using ideas from Ritz approximation theory we have developed a Basis Algorithm for solving the *optimal motion planning (or nonlinear control) problem*. The algorithm does not assume any specific structure of the $B(q)$ matrix, nor does it require calculations of the Phillip-Hall coordinates.

We have performed simulation experiments of the Basis Algo-

rithm on a number of NMP problems (see [FGL91] for more details), in particular the ball-in-socket joint model and the universal-joint model of the falling cat. The algorithm is shown to be efficient in finding near-optimal solutions. For simulation of NMP problems, the use of DASSL is especially recommended.

The techniques used here for problem formulation and the Basis Algorithm can be directly applied to space robotics, e.g., reorientation of a satellite using rotors, attitude control of a space structure (i.e., the Hubble Space Telescope, the Space Station Freedom) using internal motion.

References

- [AKN85] V. I. Arnold, V. V. Kozlov, and A.I. Neishtadt. *Dynamical Systems, III*. Springer-Verlag, 1985.
- [AM78] R. Abraham and J. Marsden. *Foundations of Mechanics*. Benjamin/Cummings Publishing Company, 2nd edition, 1978.
- [BB91] S. Boyd and C. Barratt. *Linear Controller Design*. Prentice Hall, 1991.
- [BCP89] K.E. Brenan, S.L. Campbell, and L.R. Petzold. *Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations*. North-Holland, 1989.
- [BD91] R. Brockett and L. Dai. Non-holonomic kinematics and the role of elliptic functions in constructive controllability. Technical report, Applied Science Division, Harvard University, 1991.
- [Ber85] M. Berry. Classical adiabatic angles and quantal adiabatic phase. *J. Phys. A: Math. Gen.*, 18:15–27, 1985.
- [BM89] A. Bloch and N.H. McClamroch. Control of mechanical systems with classic nonholonomic constraints. In *Conf. on Decision and Control*, pages 201–205, 1989.

- [BMR90] A. Bloch, N.H. McClamroch, and M. Reyhanoglu. Controllability and stabilizability properties of a nonholonomic control system. In *Conf. on Decision and Control*, pages 1312–1314, 1990.
- [Bro76] R.W. Brockett. Nonlinear systems and differential geometry. *Proceedings of the IEEE*, 64(1):61–72, 1976.
- [Bro81] R. Brockett. Control theory and singular riemannian geometry. In P. Hilton and G. Young, editors, *New Directions in Applied Mathematics*, pages 11 –27. Springer-Verlag, 1981.
- [CH53] R. Courant and D. Hilbert. *Methods of Mathematical Physics*, volume 1. John Wiley & Sons, 1953.
- [Cho40] W.L. Chow. Ueber systeme von linearen partiellen differentialgleichungen erster ordnung. *Math. Ann.*, 117(No.1):98–105, 1940.
- [FGL91] C. Fernandes, L. Gurvits, and Z.X. Li. Foundations of nonholonomic motion planning. Technical report, Robotics Research Laboratory, Courant Insititute of Mathematical Sciences, 1991.
- [GL90] L. Gurvits and Z.X. Li. Theory and applications of nonholonomic motion planning. Technical report, Courant Institute of Mathematical Sciences, 1990.
- [Gol80] H. Goldstein. *Classic Mechanics*. Addison-Wesley, 2nd edition edition, 1980.
- [HH70] G.W. Haynes and H. Hermes. Nonlinear controllability via lie theory. *SIAM J. Control*, 8(4):450–460, 1970.
- [HK77] R. Hermann and A.J. Krener. Nonlinear controllability and observability. *IEEE Trans. on Automatic Control*, 22:728–740, 1977.
- [Isi85] A. Isidori. *Nonlinear control systems: An introduction*. Lecture notes in control and information sciences. Springer-Verlag, 1985.

- [Kaw89] M. Kawski. Stabilization of nonlinear systems in the plane. *Systems and Control Letter*, 12:169–175, 1989.
- [KS69] T.R. Kane and M.P. Scher. A dynamical explanation of the falling cat phenomenon. *Int. J. Solid Structures*, Vol. 5, pp.663-670, 1969.
- [Lau92] J. Laumond. Singularities and topological aspects in nonholonomic motion planning. In Z.X. Li and J. Canny, editors, *Progress in Nonholonomic Motion Planning*. Kluwer Academic Publisher, 1992.
- [LC90] Z. Li and J. Canny. Motion of two rigid bodies with rolling constraint. *IEEE Trans. on Robotics and Automation*, RA2-06:62–72, 1990.
- [Li89] Z. Li. *Kinematics, Planning and Control of Dextrous Robot Hands*. PhD thesis, Dept. of EECS, Univ. of Calif. at Berkeley, 1989.
- [LS90] G. Lafferriere and H. Sussmann. Motion planning for controllable systems without drift: A preliminary report. Technical report, Rutgers University Systems and Control Center Report SYCON-90- 04, 1990.
- [Lue84] D. Luenberger. *Linear and Nonlinear Programming*. Addison-Wesley Publishing Company, 1984.
- [Mcd55] D.A. McDonald. How does a falling cat turn over. *J. Physiol. Paris*, 129:34–35, 1955.
- [Mon89] R. Montgomery. Optim. contr. of deform. body, isohol. prob., and sub-riem. geom. Technical report, MSRI, UC Berkeley, 1989.
- [MS90] R. Murray and S. Sastry. Grasping and manipulation using multifingered robot hands. Technical report, University of California at Berkeley, 1990.
- [NM90] Y. Nakamura and R. Mukherjee. Nonholonomic path planning of space robotics via bidirectional approach. In *IEEE Int. Conf. on Robotics and Automation*, 1990.

- [PD90] E. Papadopoulos and S. Dubowsky. On the nature of control algorithms for space mounted manipulators. In *IEEE Int. Conf. on Robotics and Automation*, pages 1102–1108, 1990.
- [SJ72] H.J. Sussmann and V. Jurdjevic. Controllability of nonlinear systems. *Journal of Differential Equations*, 12:95–116, 1972.
- [SL91] H. Sussmann and W. Liu. Limits of highly oscillatory controls and the approximation of general paths by admissible trajectories. Technical report, Rutgers University Center for System and Control Report, SYCON-91-02, 1991.
- [SS80] E.D. Sontag and H. Sussmann. Remarks on continuous feedback. In *Proc. of IEEE Conf. on Decision and Control*, 1980.
- [VD87] Z. Vafa and S. Dubowsky. On the dynamics of manipulators in space using the virtual manipulator approach. In *IEEE Int. Conf. on Robotics and Automation*, 1987.
- [VG88] A. M. Vershik and V. Ya. Gershkovich. Nonholonomic problems and the theory of distribution. *ACTA Applicandae Mathematicae*, Vol. 12, No. 2:181–210, 1988.
- [WK89] L.S. Wang and P.S. Krishnaprasad. A multibody analog of dual-spin problems. Technical report, System Research Center, University of Maryland, College Park, 1989.

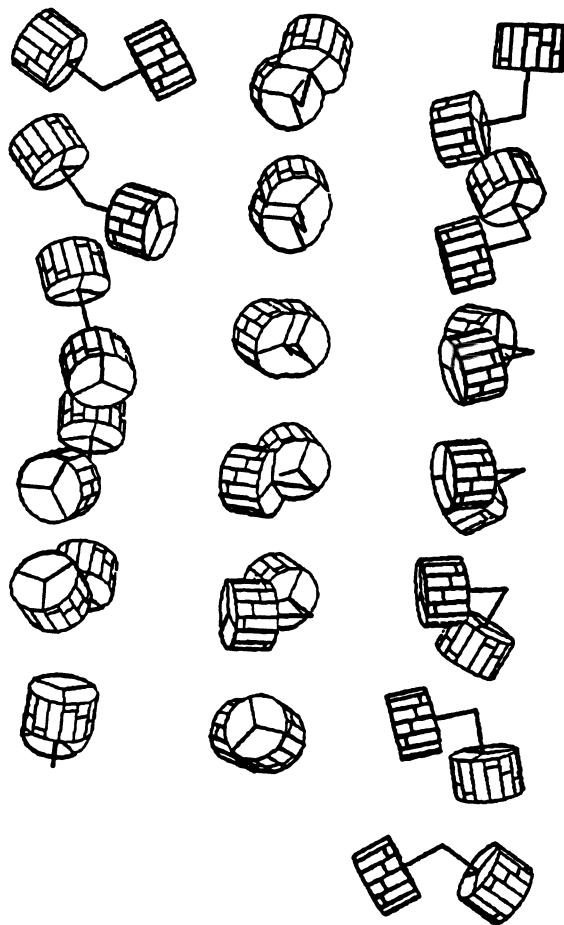


Figure 2: Simulated motion for the Universal Joint model ($N = 10$).

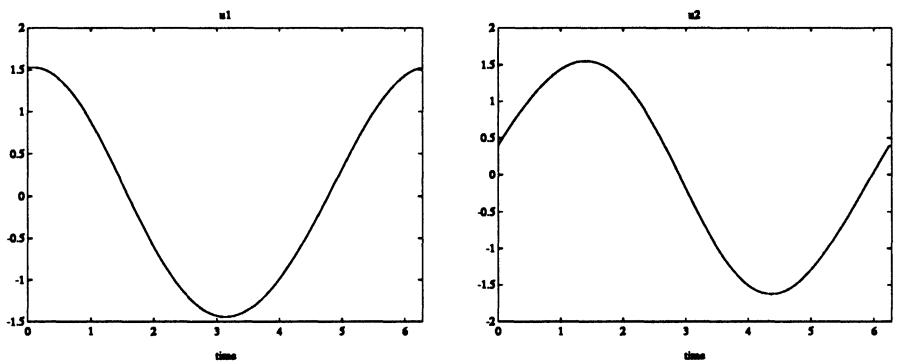


Figure 3: Optimal control input for Universal Joint model ($N = 10$).

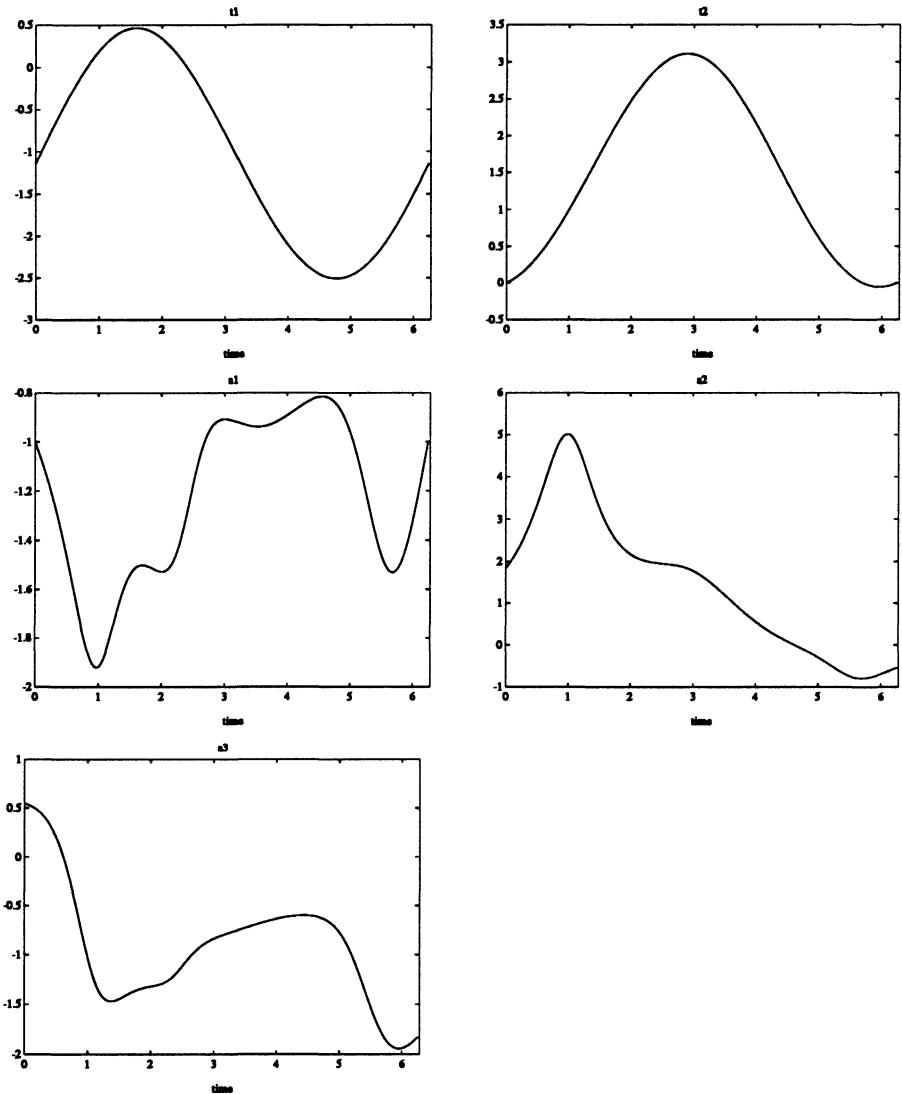


Figure 4: Optimal Trajectory for Universal Joint model ($N = 10$).

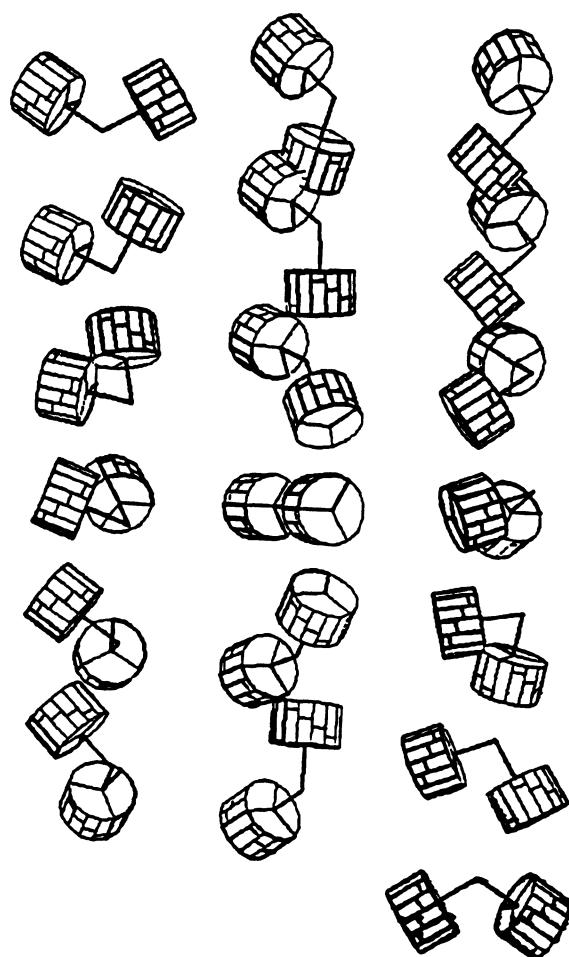


Figure 5: Simulated motion for the Universal Joint model ($N = 22$).

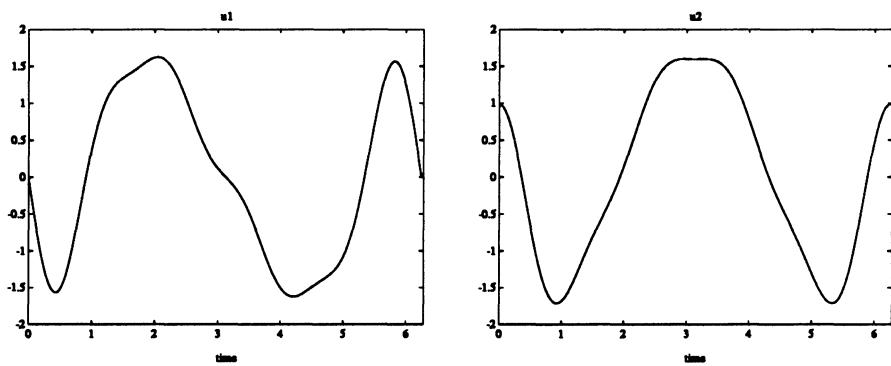


Figure 6: Optimal control input for Universal Joint model ($N = 22$).

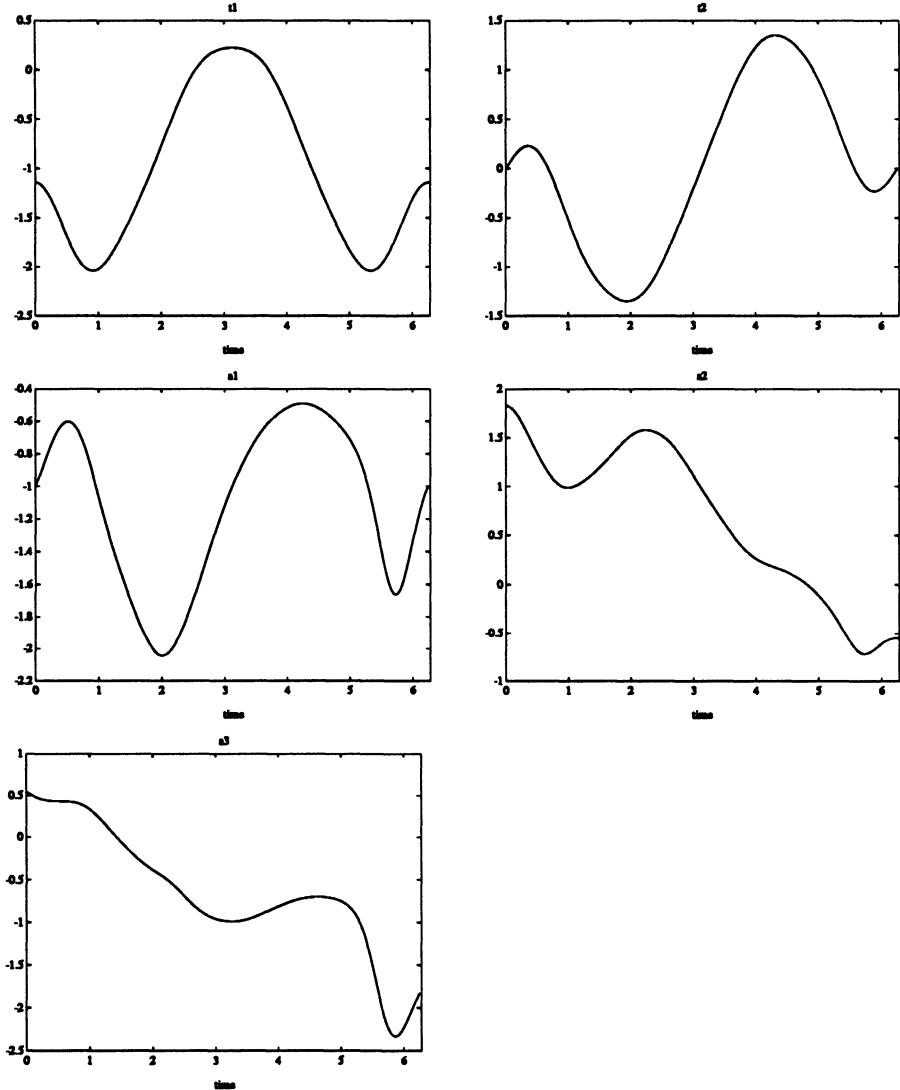


Figure 7: Optimal Trajectory for Universal Joint model ($N = 22$).

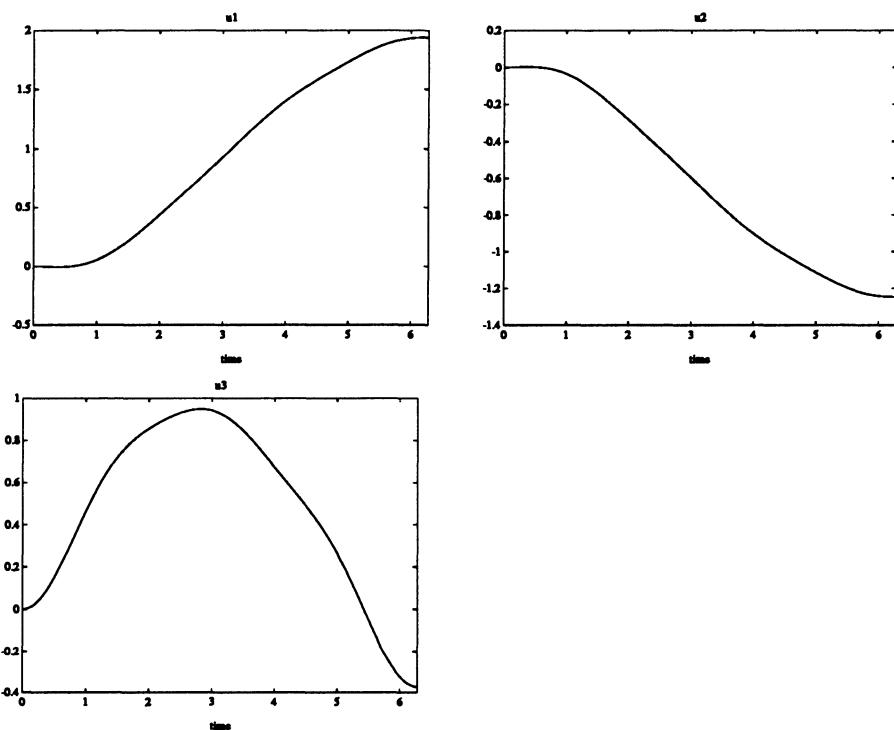


Figure 8: Optimal control input for Ball-in-Socket Joint model.

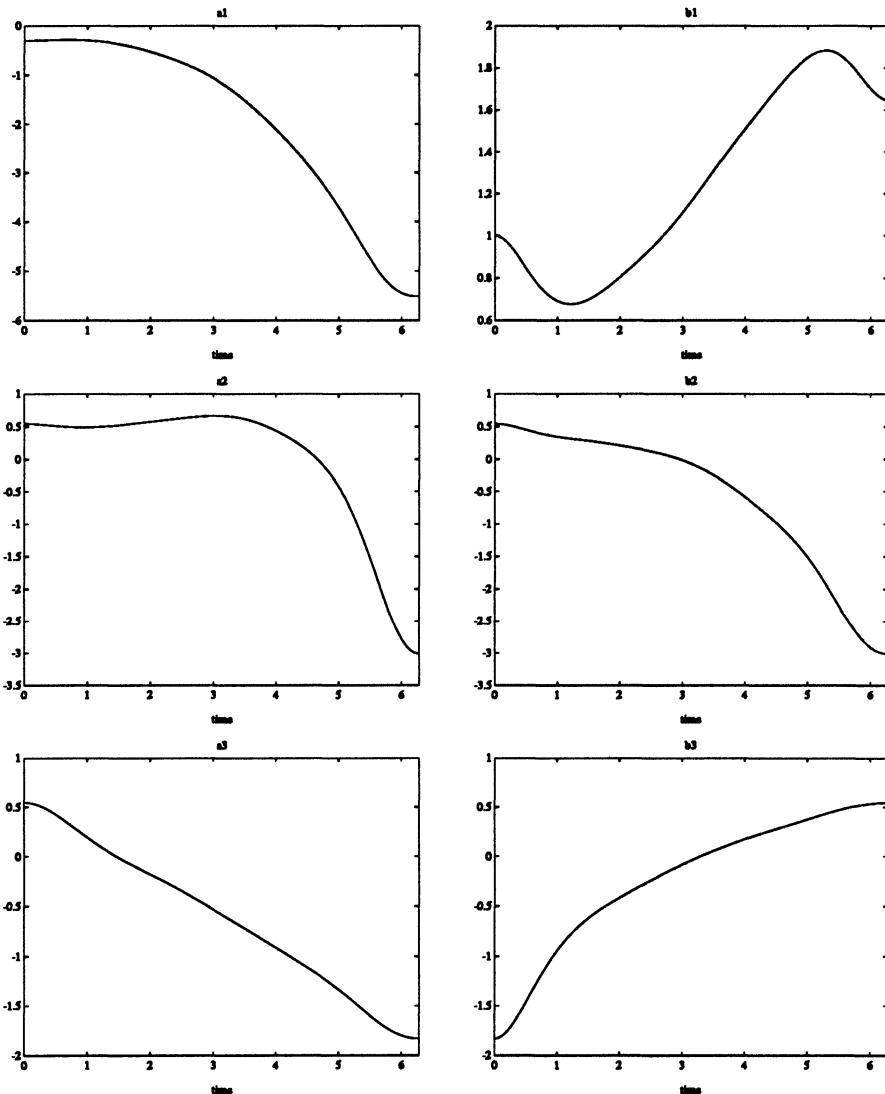


Figure 9: Optimal Trajectory for Ball-in-Socket Joint model.

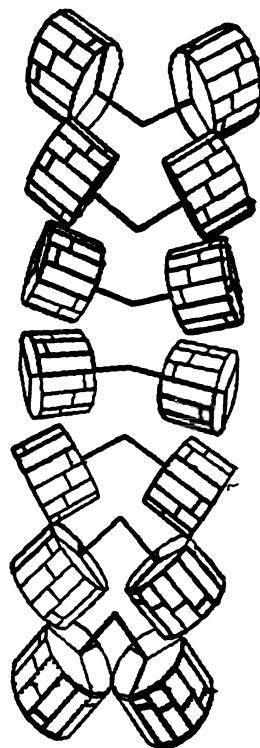


Figure 10: Simulated motion for the Ball-in-Socket Joint model.

11

Nonholonomic Behavior in Free-floating Space Manipulators and its Utilization

Evangelos G. Papadopoulos

McGill Research Centre for Intelligent Machines
3480 University St., Montreal, PQ, Canada H3A 2A7

Abstract

The kinematics and dynamics of free-floating manipulators are examined from a fundamental point of view. The dynamic coupling between an uncontrolled spacecraft and its manipulator can make a system dynamically singular at configurations which cannot be predicted by the system's kinematic properties. Nonholonomic behavior is observed in free-floating systems, and is due to the nonintegrability of the angular momentum. A workspace point can be singular or not, depending on the path taken to reach it. Trouble-free Path Independent Workspaces are defined. Nonholonomy in space manipulators is utilized by planning techniques that permit the control of a spacecraft's attitude by means of joint manipulator motions, the simultaneous control of the joint angles of a manipulator and of the attitude of its spacecraft, using joint motions only, and finally, the effective use of a system's reachable workspace by planning paths that avoid dynamically singular configurations.

I. Introduction

Space robotic devices are envisioned to assist in the construction, repair and maintenance of future space stations and satellites. To increase the mobility of such devices, *free-flying* systems in which one or more manipulators are mounted on a thruster-equipped spacecraft, have been proposed [1-6]. In such a system, dynamic

coupling between the manipulator and its spacecraft exists, and manipulator motions induce disturbances to the system's spacecraft. Thruster jets can compensate for these disturbances, but their extensive use limits severely a system's useful life span [2-4]. To increase a system's life, operation in a *free-floating* mode has been considered [3-6]. In this mode of operation, spacecraft thrusters are turned off, and the spacecraft is permitted to translate and rotate in response to its manipulator motions. In practice, this mode of operation can be feasible if the total system momentum is zero; if nonzero momentum develops, the system's thrusters must be used to eliminate it.

Free-floating systems exhibit nonholonomic behavior, which is due to the nonintegrability of the angular momentum [8,11]. This property complicates the planning and control of such systems. Joint space planning techniques that take advantage of the nonholonomy in such systems were proposed [2,7,8]. A Self Correcting Planning technique allows the control of a spacecraft's attitude using the manipulator's joint motions [2]. Lyapunov techniques were explored to achieve simultaneous control of a spacecraft's attitude and its manipulator's joint angles, using the manipulator's actuators only. Convergence problems were reported in some cases [7,8]. Various control algorithms were designed for the motion control of free-floating systems, and some of them were experimentally verified [4,5]. However, control algorithm instabilities were observed, [6], which were shown to be due to the existence of dynamic singularities [9,11].

In this paper, the fundamental kinematic and dynamic nature of free-floating manipulators is analyzed, and the nonintegrability of the angular momentum is discussed. Based on this analysis, the controllability of a free-floating system is examined in the joint and Cartesian space. It is shown that a free-floating manipulator is controllable in its joint space, but can be locally uncontrollable in the Cartesian workspace. This is due to the existence of *dynamic singularities*. Unlike to fixed-based systems, dynamically singular configurations cannot be predicted by the kinematic structure of the system, and instead depend upon its mass properties. It is shown that dynamic singularities are *path dependent* and a particular workspace point can induce a dynamic singularity or not, depending upon the path taken to reach it. Path Independent Workspaces are defined as the regions in which no dynamic singularities occur. It is shown that the nonintegrability of the angular momentum introduces nonholonomic behavior in free-floating systems. Joint space path planning techniques that take into advantage the nonholonomic behavior of free-floating systems, like the Self Correcting Planning technique, and Lyapunov-based techniques, are reviewed. Potential problems in using these techniques are identified. Finally, a Cartesian space path-planning technique is presented. This technique avoids dynamically singular configurations, and hence permits the effective use of the full reachable workspace of a free-floating system.

II. Kinematic and Dynamic Modeling of Free-floating Manipulators

A. Kinematic Modelling

This section develops the kinematic and dynamic equations needed to model a rigid free-floating manipulator system, see Figure 1. A key feature of this modeling is expressing the key kinematic and dynamic variables of the system as functions of a set of constant length, body-fixed *barycentric vectors*. The dynamics are written using a Lagrangian approach.

The body 0 in Figure 1 represents the spacecraft; the bodies k ($k=1,\dots,N$) represent the N manipulator links. $N+1$ reference frames are introduced, each one attached to the Center of Mass (*CM*) of each body, with axes parallel to the body's principal axes. Hence, the body inertia matrix expressed in this frame is diagonal. The manipulator joint angles and velocities are represented by the $N \times 1$ column vectors \mathbf{q} and $\dot{\mathbf{q}}$. The spacecraft can translate and rotate in response to manipulator movements. The manipulator is assumed to have revolute joints and an open chain kinematic configuration so that, in a system with an N degree-of-freedom (DOF) manipulator, there will be $6+N$ DOF.

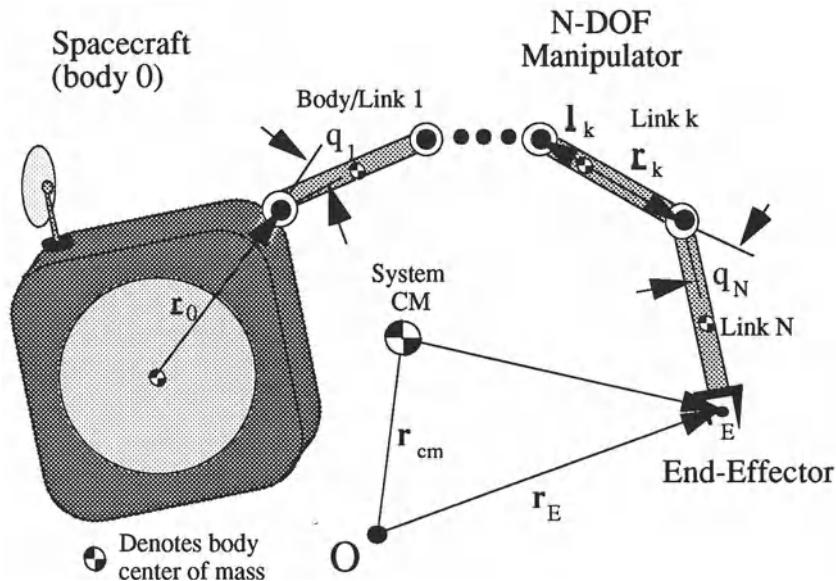


Figure 1. A spatial free-floating manipulator system.

First, the end-effector position, \mathbf{r}_E , is written with respect to the inertially fixed origin O . It can be shown that \mathbf{r}_E is given by [10,11]:

$$\mathbf{r}_E = \mathbf{r}_{cm} + \sum_{i=0}^N \mathbf{v}_{iN} + \mathbf{r}_N = \mathbf{r}_{cm} + \sum_{i=0}^N \mathbf{v}_{iN,E} \quad (1)$$

where \mathbf{r}_{cm} is the position vector of the system's *CM* with respect to the origin \mathbf{O} , vectors \mathbf{v}_{iN} are barycentric vectors fixed on body i , and given in Appendix A by Equation (A1), \mathbf{r}_N is the vector from the *CM* of the last link to the end-effector, and $\mathbf{v}_{iN,E} = \mathbf{v}_{iN} + \delta_{iN} \mathbf{r}_N$, where δ_{iN} is the Kronecker's delta, [12,13,11]. Assuming that no external forces act on the system, the system *CM* does not accelerate, and the system linear momentum $\mathbf{p} = M\mathbf{r}_{cm}$ is constant. With the further assumption of zero initial linear momentum, $\dot{\mathbf{r}}_{cm}$ is also a constant, and can be taken to be zero without loss of generality.

The orientation of link i is described by the transformation matrix \mathbf{T}_i , given by:

$$\mathbf{T}_i(\Theta, q_1, \dots, q_i) = \mathbf{T}_0(\Theta) {}^0\mathbf{T}_i(q_1, \dots, q_i) \quad (2)$$

where Θ represents the orientation of the spacecraft, \mathbf{T}_0 is a 3×3 transformation matrix that describes the orientation of the spacecraft frame with respect to the inertial frame, and ${}^0\mathbf{T}_i$ is a 3×3 transformation matrix, function of the joint angles (q_1, \dots, q_i), that describes the orientation of the i^{th} frame with respect to the spacecraft frame. The orientation of the N^{th} link, is the orientation of the end-effector.

The end-effector inertial linear velocity, $\dot{\mathbf{r}}_E$, is obtained by differentiation of Equation (1). Since the barycentric vectors are body-fixed, $\dot{\mathbf{r}}_E$ is given simply by:

$$\dot{\mathbf{r}}_E = \dot{\mathbf{r}}_{cm} + \sum_{i=0}^N \omega_i \times \mathbf{v}_{iN,E} = \sum_{i=0}^N \omega_i \times \mathbf{v}_{iN,E} \quad (3)$$

where ω_i is the inertial angular velocity of body i , $\dot{\mathbf{r}}_{cm}$ is the velocity of the system's *CM* taken equal to zero, and the \times converts a vector to the cross-product skew-symmetric matrix, see Equation (A4). The inertial angular velocity ω_k is written as a function of the spacecraft's angular velocity expressed in the 0^{th} frame, ${}^0\omega_0$, and the manipulator joint rates, $\dot{\mathbf{q}}$:

$$\omega_k = \mathbf{T}_0({}^0\omega_0 + {}^0\mathbf{F}_k \dot{\mathbf{q}}) \quad k = 1, \dots, N \quad (4)$$

where ${}^0\mathbf{F}_k$ is a $3 \times N$ matrix given by:

$${}^0\mathbf{F}_k = [{}^0\mathbf{T}_1^{-1} \mathbf{u}_1, {}^0\mathbf{T}_2^{-2} \mathbf{u}_2, \dots, {}^0\mathbf{T}_k^{-k} \mathbf{u}_k, \mathbf{0}] \quad k = 1, \dots, N \quad (5)$$

The vector ${}^i\mathbf{u}_i$ is the unit column vector in frame i parallel to the revolute axis through joint i , and $\mathbf{0}$ is a $3 \times (N-k)$ zero element matrix. The end-effector angular velocity is simply given by:

$$\omega_E = \omega_N \quad (6)$$

It can be shown that in the absence of external torques, and for zero initial $\dot{\mathbf{r}}_{cm}$, the constant angular momentum of the system is given by the sum [9,11]:

$$\mathbf{h} = \sum_{j=0}^N \sum_{i=0}^N \mathbf{D}_{ij} \omega_j \quad (7)$$

where \mathbf{D}_{ij} are mixed inertia matrices, functions of the barycentric vectors, and given by Equation (A5). Assuming zero initial angular momentum, Equation (7) can be solved for ${}^0\omega_0$ to yield:

$${}^0\omega_0 = - {}^0\mathbf{D}^{-1} {}^0\mathbf{D}_q \dot{\mathbf{q}} \quad (8)$$

where ${}^0\mathbf{D}$ is the 3×3 inertia matrix of the system expressed in the spacecraft frame at the system CM , and ${}^0\mathbf{D}_q$ is a $3 \times N$ matrix that corresponds to the inertia of the system's moving parts, see Equations (A7). Note that the inverse of ${}^0\mathbf{D}$ always exists because the system inertia matrix is positive definite.

Equation (8) can be used to eliminate the spacecraft angular velocity ${}^0\omega_0$ from Equations (3) and (6), and hence to derive a free-floating system's Jacobian \mathbf{J}^* , defined by:

$$[\dot{\mathbf{r}}_E, \omega_E]^T = \mathbf{J}^* \dot{\mathbf{q}} \quad (9)$$

and given as:

$$\mathbf{J}^*(\Theta, \mathbf{q}) = \text{diag}(\mathbf{T}_0, \mathbf{T}_0) {}^0\mathbf{J}^*(\mathbf{q}) \quad (10)$$

$${}^0\mathbf{J}^*(\mathbf{q}) = \begin{bmatrix} {}^0\mathbf{J}_{11} & {}^0\mathbf{D}^{-1} {}^0\mathbf{D}_q + {}^0\mathbf{J}_{12} \\ {}^0\mathbf{D}^{-1} {}^0\mathbf{D}_q + {}^0\mathbf{J}_{22} & \end{bmatrix} \quad (11)$$

where the matrices ${}^0\mathbf{J}_{11}$, ${}^0\mathbf{J}_{12}$, and ${}^0\mathbf{J}_{22}$, are defined in Appendix A.

Note that these Jacobians are *basic Jacobians*, that is Jacobians independent of the particular parameter set used to describe the end-effector orientation [14]. Kinematic equations related to the particular orientation representation also must be used. Equation (9) describes the effect of joint motions on end-effector velocities, and its form is the same to the form that of that for fixed-based systems. Comparing the structure of ${}^0\mathbf{J}^*$ to the structure of the Jacobian \mathbf{J} , that would be written for the same manipulator but with a fixed base, it is easy to see that these are the same. Indeed, both depend on the configuration \mathbf{q} , and have the same size, $6 \times N$. The same observations hold for \mathbf{J}^* , with the exception that in addition, this is also a function of the spacecraft orientation. However, this orientation can be estimated or measured, and hence, \mathbf{J}^* can be used in the place of \mathbf{J} in well-established control or planning algorithms derived for fixed-based systems.

Note that \mathbf{J}^* depends not only on the kinematic properties of the system, but also on configuration dependent mass properties, eg. inertias. This observation suggests that singular configurations for a free-floating system, i.e. ones in which ${}^0\mathbf{J}^*$ has rank less than six, will not be the same to the ones for fixed based systems. This is indeed the case, as we will see later.

B. Dynamic Modelling

To derive the equations of motion of a free-floating system, the system kinetic energy is expressed as a function of the generalized coordinates and their velocities, [10,11].

$$T = \frac{1}{2} \sum_{j=0}^N \sum_{i=0}^N \omega_i^T D_{ij} \omega_j \quad (12)$$

It can be shown that under the same assumptions as above, T is given by [10,11]:

$$T = \frac{1}{2} \dot{\mathbf{q}}^T \mathbf{H}^*(\mathbf{q}) \dot{\mathbf{q}} \quad (13)$$

where $\mathbf{H}^*(\mathbf{q})$ is the *reduced* system inertia matrix, given by:

$$\mathbf{H}^*(\mathbf{q}) \equiv {}^0\mathbf{D}_{qq} - {}^0\mathbf{D}_q^T {}^0\mathbf{D}^{-1} {}^0\mathbf{D}_q \quad (14)$$

The matrices ${}^0\mathbf{D}$, ${}^0\mathbf{D}_q$, and ${}^0\mathbf{D}_{qq}$ are defined by Equations (A7). It is easy to show that the system inertia matrix, \mathbf{H}^* , is an NxN positive definite symmetric inertia matrix, which depends on \mathbf{q} and the system mass and inertia properties. All elements of \mathbf{H}^* are functions of the manipulator joint angles q_i ($i=1,\dots,N$) only, since ${}^0\mathbf{D}$, ${}^0\mathbf{D}_q$, and ${}^0\mathbf{D}_{qq}$ are functions of only the q_i 's and not of the spacecraft attitude; hence the system inertia matrix \mathbf{H}^* has the same structural properties as the inertia matrices that correspond to fixed-base manipulators.

In the absence of gravity, the potential energy of a rigid system is zero, and the system's dynamic equations are given by:

$$\frac{d}{dt} \left\{ \frac{\partial T}{\partial \dot{\mathbf{q}}} \right\} - \frac{\partial T}{\partial \mathbf{q}} = \boldsymbol{\tau} \quad (15)$$

where $\boldsymbol{\tau}$ is the generalized force vector which, in this case, is equal to the torque vector $[\tau_1, \tau_2, \dots, \tau_N]^T$. Applying Equation (15) to the kinetic energy given by Equation (13) results in a set of N equations of motion of the form:

$$\mathbf{H}^*(\mathbf{q}) \ddot{\mathbf{q}} + \mathbf{C}^*(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{q}} = \boldsymbol{\tau} \quad (16)$$

where $\mathbf{H}^*(\mathbf{q})$, is the system inertia matrix defined by Equation (14) and $\mathbf{C}^*(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{q}}$ contains nonlinear centrifugal and Coriolis terms. Note that the equations of motion are written as functions of the joint variables only, and not of the spacecraft variables. This results from the fact that the system kinetic energy does not depend on a spacecraft's attitude nor on its angular or linear velocity, when the system initial angular momentum is zero, and the system is free of external torques. The spacecraft's contribution to the system's kinetic energy, T, enters in through the inertia matrices ${}^0\mathbf{D}$ and ${}^0\mathbf{D}_q$, which depend on its mass m_0 and inertia \mathbf{I}_0 .

If it is assumed that a particular task requires motion control of the end-effector, then Equations (9) and (16) can be used to design a controller. Based on the structural similarity of these equations to the ones derived for a fixed based system, Reference [10] suggested that if singularities of \mathbf{J}^* can be avoided, nearly any control algorithm applied to fixed-based systems can be used in free-floating systems. The nature of free-floating system singularities and workspaces, in conjunction to the nonintegrability of the angular momentum, is addressed next.

III. Controllability, Dynamic Singularities and Workspaces

A. Controllability in the Joint Space

Assume that one task requires control of the system configuration \mathbf{q} , only. Then, a linearizing feedforward control law of the form $\tau = \mathbf{H}^*(\mathbf{q}) \mathbf{u} + \mathbf{C}^*(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{q}}$, where $\mathbf{u} \in \mathbb{R}^N$ is an auxiliary control input, reduces the equations of motion to a controllable decoupled second order system. This can be done because \mathbf{H}^* is a positive definite matrix, and proves that the system is controllable in its joint space.

B. Nonintegrability of the Angular Momentum

The angular momentum, given by Equation (8), cannot be integrated to yield the spacecraft's orientation as a function of the system's configuration, \mathbf{q} , with the exception of a planar two body system [11]. Obviously, this equation can be integrated numerically, but in such a case the resulting final spacecraft orientation will be a function of the path taken in the joint space. In other words, different paths in the joint space, with the same initial and final points, will result in different spacecraft orientations. Due to Equation (9), the same applies to workspace paths; i.e. moving from one workspace location to another one via different paths results in different final spacecraft orientations. Closed paths in the joint space or the workspace can change the system's attitude.

It is this nonintegrability property that introduces nonholonomic characteristics to free-floating systems. However, this nonholonomic behavior results from the particular *dynamic* structure of the system, and is not due to *kinematic* nonintegrable constraints, like the ones experienced by a rolling disk. The use of this nonholonomic behavior to achieve various tasks is described in the following sections.

C. Controllability in the Cartesian Space

Assume next that the task is to move the end-effector to some position and orientation, and that $N = 6$. Since the system is controllable in its joint space, any $\mathbf{q}, \dot{\mathbf{q}}$ can be obtained. The question that arises next is whether this is enough to obtain any \mathbf{r}_E, ω_E , and eventually any position and orientation. The answer to this question is affirmative if the Jacobian \mathbf{J}^* is of full rank, i.e. six. Similar observations hold for the more general case of an N DOF manipulator. Since the transformation matrix \mathbf{T}_0 is not singular, (with the exception of possible representation singularities), then \mathbf{J}^* loses its full rank when:

$$\det[^0\mathbf{J}^*(\mathbf{q})] = 0 \quad (17)$$

The above condition shows that singularities in free-floating systems are fixed in joint space. However, since ${}^0\mathbf{J}^*$ is a function of configuration dependent inertia matrices, these singularities are different than the ones for fixed base systems, and

their location in joint space depend in addition on the dynamic properties of the system; for these reasons, they were called *dynamic singularities* [9].

It is interesting to examine the location of the dynamic singularities in a system's workspace. To do this we need a one to one correspondence from the joint space to the cartesian workspace. However, such a correspondence does not exist, even in the case of a six DOF manipulator, because its end-effector position \mathbf{r}_E , and orientation \mathbf{T}_E , are not only functions of the system's configuration \mathbf{q} , but also of the path dependent spacecraft orientation, Θ , see also Equations (1) and (2):

$$\mathbf{r}_E(\Theta, \mathbf{q}) = \mathbf{T}_0(\Theta) \sum_{i=0}^6 {}^0\mathbf{T}_i {}^i\mathbf{v}_{i6,E} \quad (18)$$

$$\mathbf{T}_E(\Theta, \mathbf{q}) = \mathbf{T}_0(\Theta) {}^0\mathbf{T}_6(\mathbf{q}_1, \dots, \mathbf{q}_6) \quad (19)$$

The ${}^i\mathbf{v}_{i6,E}$ are constant vectors. Out of all the pairs (Θ, \mathbf{q}) with which a workspace point can be reached, some may correspond to a singular configuration, \mathbf{q}_s . Then a workspace point may or may not induce a dynamic singularity, depending on the joint space path taken to reach it.

To resolve this ambiguity, *Path Dependent Workspaces* (PDW) were defined to contain all workspace locations that may induce a dynamic singularity [9,11]. To find these points, note that the distance of a workspace location from the system *CM* does not depend on the spacecraft's orientation:

$$R = R(\mathbf{q}) = \left\| \sum_{i=0}^N {}^0\mathbf{T}_i {}^i\mathbf{v}_{i6,E} \right\| \quad (20)$$

This equation represents a spherical shell in the workspace. All the singular configurations \mathbf{q}_s are mapped through Equation (20) to a set of shells, whose union gives the PDW. If we subtract the PDW from the reachable workspace, contained in a spherical volume defined by:

$$R_{\min} = \min_{\mathbf{q}} R(\mathbf{q}) \quad (21a)$$

$$R_{\max} = \max_{\mathbf{q}} R(\mathbf{q}) \quad (21b)$$

we get the *Path Independent Workspace*, (PIW). All points in the PIW are guaranteed not to induce dynamic singularities. Then, any point in the PIW can be reached from all other points in the PIW, by any path that belongs entirely to the PIW.

If the system is in a dynamically singular configuration, the end-effector is constrained to move on a manifold of dimension lower than six. This means that some workspace points are not reachable with small $\delta\mathbf{q}$, whatever $\delta\mathbf{q}$ is. In other words, the system is locally uncontrollable. However, it may still be possible to reach any PDW point from any other workspace point, by choosing an appropriate path. This will be demonstrated in the following sections.

C. Controllability in the State Space

A system with a 6 DOF manipulator has 12 DOF, the additional six corresponding to the position and orientation of the spacecraft. Assuming an Euler angle representation of a spacecraft's orientation, a state space can be formed containing the position and orientation of the spacecraft, the joint angles or the end-effector position and orientation, and the corresponding velocities or rates. The dimension of such a state space is 2·12. However, since the linear momentum can be integrated to yield a spacecraft's position as a function of the manipulator links, one cannot set the position of the spacecraft independently of the position of the manipulator links. This shows that the controllable subspace has dimension at most 2·9. In addition, due to the angular momentum given by Equation (8), a spacecraft's angular velocity cannot be specified independently of the joint rates \dot{q} . Hence, the controllable subspace is at most of dimension 2·9-3=15. Although the exact dimension of the controllable subspace has not been established yet, it has been shown by specific examples that it is possible to control a spacecraft's orientation, in addition to the manipulator joint variables, by employing special joint space paths.

IV. Example

Consider the planar free-floating space manipulator shown in Figure 2. The system parameters are given in Table I. For this system, vectors r_i and l_i that connect a link's *CM* to its two joints, see Figure 1, are parallel to the x axis of the i^{th} frame. Hence, only the x -component of the barycentric vectors in Equation (A1) is non-zero, and is given by:

$$\begin{aligned} {}^0v_{02,E} &\equiv \alpha = \frac{1}{M} r_0 m_0 = 0.426 \text{ m} \\ {}^1v_{12,E} &\equiv \beta = \frac{1}{M} \{ r_1(m_0+m_1) + l_1 m_0 \} = 0.894 \text{ m} \\ {}^1v_{22,E} &\equiv \gamma = \frac{1}{M} l_2(m_0+m_1) + r_2 = 0.968 \text{ m} \end{aligned} \quad (22)$$

where M is the total system mass, $M = m_0 + m_1 + m_2$. For this system, only the position of the end-effector, $r_E = [x, y]^T$, is controlled. This position is written using Equation (18) as:

$$\begin{aligned} x &= \alpha \cos(\theta) + \beta \cos(\theta+q_1) + \gamma \cos(\theta+q_1+q_2) \\ y &= \alpha \sin(\theta) + \beta \sin(\theta+q_1) + \gamma \sin(\theta+q_1+q_2) \end{aligned} \quad (23)$$

where θ , q_1 and q_2 , are defined in Figure 2. As shown in [9], the system Jacobian is:

$$J^*(\theta, q) = T_0(\theta) {}^0J^*(q) = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} {}^0J^*(q) \quad (24a)$$

$${}^0\mathbf{J}^*(\mathbf{q}) = \frac{1}{D} \begin{bmatrix} -(\beta s_1 + \gamma s_{12}) D_0 & \beta s_1 D_2 - \gamma s_{12} (D_0 + D_1) \\ -\alpha(D_1 + D_2) + (\beta c_1 + \gamma c_{12}) D_0 & -(\alpha + \beta c_1) D_2 + \gamma c_{12} (D_0 + D_1) \end{bmatrix} \quad (24b)$$

where $s_1 = \sin(q_1)$, $c_{12} = \cos(q_1 + q_2)$ etc. The inertia scalar sums D , D_0 , D_1 and D_2 are defined in Appendix B, see Equation (B2). Since each D_i ($i=0,1,2$) and D are functions of \mathbf{q} , the Jacobian elements are more complicated functions of the \mathbf{q} than their fixed-base counterparts. Note that D represents the inertia of the whole system with respect to its *CM* and thus, is always a positive number.

Table I. The system parameters.

Body	l_i (m)	r_i (m)	m_i (Kg)	I_i (Kg m ²)
0	.5	.5	40	6.667
1	.5	.5	4	0.333
2	.5	.5	3	0.250

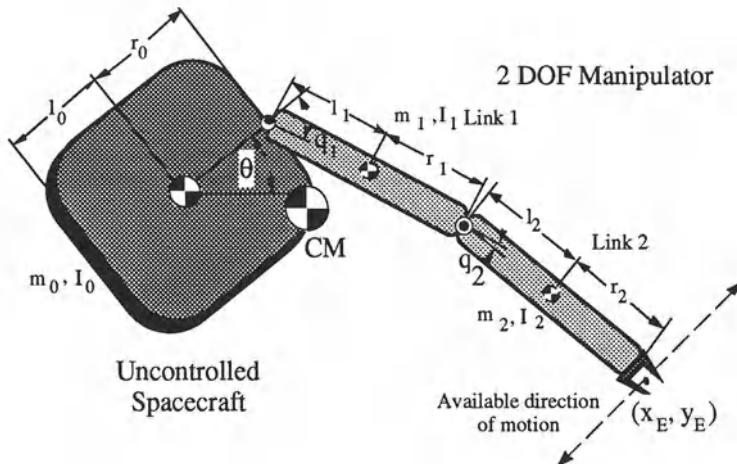


Figure 2. A planar 2 DOF free-floating manipulator, shown in a dynamically singular configuration.

The system inertia matrix, \mathbf{H}^* , is found according to Equations (14), (B1), and (B2). The result is:

$$\mathbf{H}^*(\mathbf{q}) = \begin{bmatrix} {}^0d_{11} + 2{}^0d_{12} + {}^0d_{22} - \frac{(D_1 + D_2)^2}{D} & {}^0d_{12} + {}^0d_{22} - \frac{D_2(D_1 + D_2)}{D} \\ {}^0d_{12} + {}^0d_{22} - \frac{D_2(D_1 + D_2)}{D} & {}^0d_{22} - \frac{D_2^2}{D} \end{bmatrix} \quad (25)$$

where the mixed inertia terms ${}^0d_{ij}$ are defined in Appendix B. Note that H^* is a 2x2 positive definite symmetric matrix whose elements are functions of the joint angles q_1 and q_2 , and its size and structure is the same to the inertia matrix of a fixed base system. Hence, as discussed above, it is easy to show that the example system is controllable in its joint space.

The zero angular momentum for this system is written using Equation (8) as:

$$\dot{D}\theta + (D_1 + D_2)\dot{q}_1 + D_2\dot{q}_2 = 0 \quad (26)$$

Multiplying both sides by dt , a Pfafian equation results. This Pfafian can only be integrated if the following condition holds [15]:

$$D \left\{ \frac{\partial(D_1 + D_2)}{\partial q_1} - \frac{\partial D_2}{\partial q_1} \right\} - (D_1 + D_2) \frac{\partial D}{\partial q_2} + D_2 \frac{\partial D}{\partial q_1} = 0 \quad (27)$$

However, after some algebra one can show that this condition does not hold, and therefore, the angular momentum cannot be integrated to yield θ as a function of q_1 and q_2 . Nonholonomic behavior is expected for this system.

The system Jacobian becomes singular, when its determinant is zero. This condition results in the following equation:

$$\alpha\beta D_2 \sin(q_1) + \beta\gamma D_0 \sin(q_2) - \alpha\gamma D_1 \sin(q_1 + q_2) = 0 \quad (28)$$

The values of q_1 and q_2 which satisfy Equation (28) and result in dynamically singular configurations can be plotted in joint space as shown in Figure 3. This figure also shows that conventional kinematic singularities like $q_1 = k\pi$, $q_2 = k\pi$, $k=0,\pm 1, \dots$ still satisfy Equation (28). However, infinitely more dynamically singular configurations exist which cannot be predicted from the kinematic structure of the manipulator.

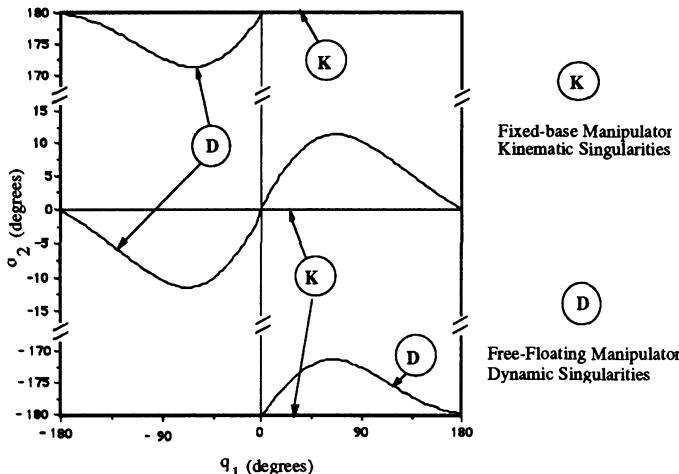


Figure 3. Dynamically singular configurations for the system shown in Figure 2.

Figure 3 shows the system in the singular configuration at $q_1 = -65^\circ$, $q_2 = -11.41^\circ$, and the spacecraft attitude at $\theta = 40^\circ$. In this configuration, the local inertial motion of the end-effector will be shown in the figure, no matter how the joint actuators are driven, and the system is locally uncontrollable. The best a control algorithm can do at such a point is to follow the available direction. All algorithms that use a Jacobian inverse, such as the resolved rate or resolved acceleration control algorithms, fail at such a point. Ones that use a pseudoinverse Jacobian or a Jacobian transpose will likely follow the available direction, but may result in large errors.

To find the limits of the reachable workspace, the distance R of the end-effector from the system CM given by Equation (20) is written as:

$$R = R(q) = \sqrt{\alpha^2 + \beta^2 + \gamma^2 + 2\alpha\beta\cos(q_1) + 2\alpha\gamma\cos(q_1+q_2) + 2\beta\gamma\cos(q_2)} \quad (29)$$

For this example, the reachable workspace is the area confined between two circles with radii:

$$R_{\min} = 0.352 \text{ m} = \alpha + \beta - \gamma \quad (30a)$$

$$R_{\max} = 2.288 \text{ m} = \alpha + \beta + \gamma \quad (30b)$$

while the PIW is confined between the circles with radii:

$$R_1 = 0.554 \text{ m} \quad (31a)$$

$$R_2 = 1.436 \text{ m} \quad (31b)$$

Figure 4 depicts the reachable, PDW, and PIW spaces for this example. When the end-effector path has points belonging to the PDW, such as path B in Figure 4, the manipulator may assume a dynamically singular configuration, depending on the path. On the other hand, paths totally within the PIW region, such as path A, can never lead to dynamically singular configurations.

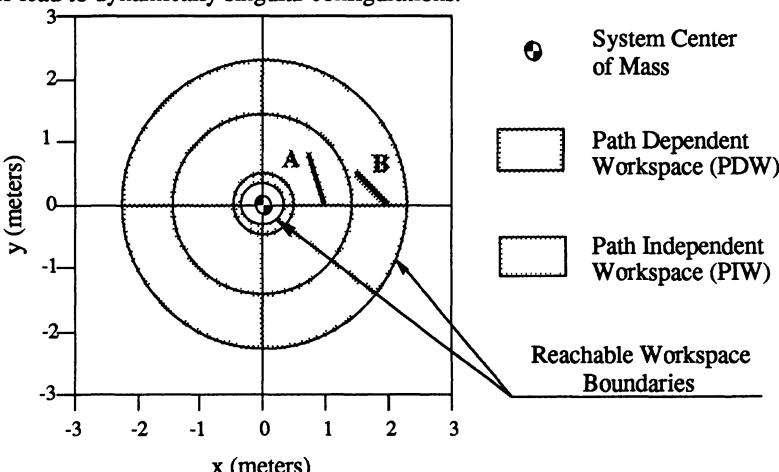


Figure 4. The reachable, Path Dependent, and Path Independent Workspaces, for the system shown in Figure 2.

In the next section we explore the use of the nonholonomic behavior of a free-floating system in planning to achieve various tasks.

V. Path Planning in Joint Space

a. Self Correcting Planning

A free-floating system may operate under the Spacecraft-Referenced End-Point Motion Control, in which either the manipulator end-point is commanded to move to a location fixed to its own spacecraft, or a simple joint motion is commanded, such as when the manipulator is to be driven at its stowed position [10]. In general, a manipulator's motion in joint space will change the spacecraft's orientation, as discussed above. However, there are many cases in which this phenomenon may be highly undesirable. For example, the spacecraft may be required to maintain a constant orientation for communication purposes. Therefore, it would be useful to control a system's orientation, without using limited thruster fuel.

Due to the nonholonomic behavior of a free-floating system, different joint space paths with the same initial and final points, will result in different spacecraft orientations. In addition, a closed path in the manipulator's joint space, will result in a net change in the spacecraft's orientation. Based on these observations, a Self Correcting Planning technique that can correct for any deviations from a desired orientation by executing closed joint space paths, has been proposed [2]. Here, the basic elements of this technique are reviewed.

If a spacecraft's orientation is described by the 3-2-1 Euler angles, $\Theta = [\theta_1, \theta_2, \theta_3]^T$, then these are written using Equation (8) as [16]:

$$\dot{\Theta} = -S^{-1}(\Theta) \omega_0 = -S^{-1}(\Theta) T_0^0 D^{-1} {}^0 D_q \dot{q} = G(\Theta, q) \dot{q} \quad (32)$$

where $S^{-1}(\Theta)$ is a nonsingular matrix, except at some isolated points, and is given by:

$$S^{-1}(\Theta) = \begin{bmatrix} 1 & \sin\theta_1 \sin\theta_2 / \cos\theta_2 & \cos\theta_1 \sin\theta_2 / \cos\theta_2 \\ 0 & \cos\theta_1 & -\sin\theta_1 \\ 0 & \sin\theta_1 / \cos\theta_2 & \cos\theta_1 / \cos\theta_2 \end{bmatrix} \quad (33)$$

For small changes in the configuration q , Equation (32) is written as:

$$\delta\Theta = G(\Theta, q) \delta q \quad (34)$$

where G is a $3 \times N$ matrix. Using a Taylor series expansion of Equation (34), and assuming a joint space closed path along the vectors δV , δW , $-\delta V$, $-\delta W$, the resulting change in the Euler angles $\delta\Theta$ is given by [2]:

$$\delta\theta_i \approx \sum_{l=1}^N \sum_{m=1}^N \left[\sum_{n=1}^3 \left(\frac{\partial G_{im}}{\partial \theta_n} G_{nl} - \frac{\partial G_{il}}{\partial \theta_n} G_{nm} \right) + \frac{\partial G_{im}}{\partial q_l} - \frac{\partial G_{il}}{\partial q_m} \right] \delta V_l \delta W_m \quad (i = 1, 2, 3) \quad (35)$$

Equation (35) can be used to find the joint space path, as described by vectors δV , and δW , to achieve a correction in the spacecraft's orientation by $\delta\Theta$. Note that this is possible if at least one of the terms in brackets in Equation (35) is nonzero.

If the dimension of the manipulator's joint space is three, Equation (35) represents three equations in six unknowns, i.e. δV_i , δW_i , for $i=1,2,3$. The additional constraints:

$$\delta V^T \delta W = 0 \quad (36a)$$

$$\delta V^T \delta V = \delta W^T \delta W \quad (36b)$$

$$\delta V_3 = (\delta V_1 + \delta V_2)/2 \quad (36c)$$

allow complete determination of the required joint space path. This technique works well if $\delta\Theta$ is small. If a large correction is required, this is broken in smaller ones, and more than one correction cycles are performed.

Example

Consider the system introduced in Section IV. It is desired to estimate the number of joint space closed square paths required to achieve a specified change in the spacecraft's orientation. For this system, G is a function of the configuration q only:

$$G(q) = [G_1, G_2] = \left[-\frac{(D_1+D_2)}{D}, -\frac{D_2}{D} \right] \quad (37)$$

According to Equations (36), vectors δV and δW are chosen to be:

$$\delta V = [\delta q, 0] \quad \delta W = [0, \delta q] \quad (38)$$

where δq represents a small change in a joint angle. Equation (35) reduces to:

$$\delta\Theta = \left(\frac{\partial G_2}{\partial q_1} - \frac{\partial G_1}{\partial q_2} \right) \delta q^2 = g(q_1, q_2) \delta q^2 \quad (39)$$

where $g(q_1, q_2)$ is a measure of the influence of a closed joint path on a spacecraft's orientation, and given by:

$$g(q_1, q_2) = -2 \frac{{}^0d_{01}D_2 \tan(q_1) + {}^0d_{12}D_0 \tan(q_2) - {}^0d_{02}D_1 \tan(q_1 + q_2)}{D^2} \quad (40)$$

Assuming that at some particular configuration $g(q_1, q_2)$ is nonzero, Equation (40) yields the change in orientation of the spacecraft as a function of the area of the closed joint space path. If this path is a square with side δq , the number of paths required to achieve a change $\Delta\theta$ in the orientation, is obtained from Equation (39) as:

$$m \approx \frac{\Delta\theta 180^\circ}{(\delta q)^2 \pi \bar{g}} \quad (41)$$

where both $\Delta\theta$ and δq are in degrees, \bar{g} is the value of $g(q_1, q_2)$ evaluated at $(q_1 + \delta q/2, q_2 + \delta q/2)$, and $\pi = 3.14$.

To demonstrate the use of Equation (41), assume that the system is at $(\theta, q_1, q_2) = (14^\circ, -48^\circ, 145^\circ)$. Then $\bar{g} = g(-43^\circ, 150^\circ) = -0.0495$. If the desired final θ is 10° , then $\Delta\theta = -4^\circ$. Assuming a square joint path of side $\delta q = 10^\circ$, and using Equation (41), we find that the required number of square paths is $m = 46$. Figure 5 shows the orientation as a function of m . After the execution of 46 closed joint paths, the spacecraft's orientation becomes 10.06° .

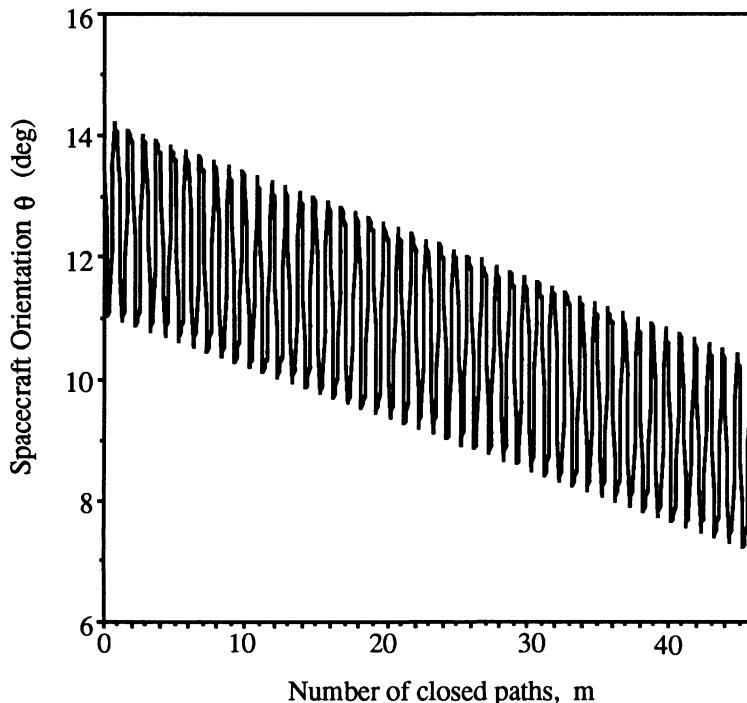


Figure 5. Changes in the spacecraft orientation θ , during the execution of closed joint space paths.

As discussed earlier, this self-correcting technique can be used if \bar{g} is nonzero. However, \bar{g} becomes zero for certain configurations q , shown in Figure 6. When the system manipulator is in one of these configurations, its spacecraft orientation cannot be affected by small joint space closed paths. These configurations can be mapped to cartesian space areas, using Equation (29) and the same procedure used for constructing the PDW. For this example, one can show that g is nonzero everywhere in the system's PIW. In other words, if the end-effector is in the PIW, the spacecraft orientation is always affected by closed joint space paths.

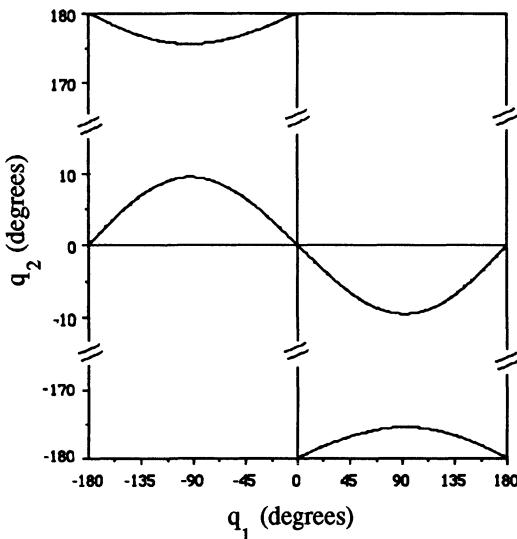


Figure 6. Configurations at which joint motions have no effect on the spacecraft's orientation θ .

b. Lyapunov-based Planning Techniques

In some cases, it is desirable to be able to control both a system's configuration \mathbf{q} , and the orientation of its spacecraft, using joint motions only. A Lyapunov-based technique designed to achieve this goal was presented in References [7,8], and is reviewed here.

First, a vector $\mathbf{y} = [\Theta, \mathbf{q}]^T \in \mathbb{R}^{N+3}$, that contains the variables to be controlled is assembled. If the constant vector \mathbf{y}_d denotes the desired \mathbf{y} , the distance of the current \mathbf{y} from the desired one is given by $\Delta\mathbf{y}$:

$$\Delta\mathbf{y} = \mathbf{y}_d - \mathbf{y} \quad (42)$$

Next, a Lyapunov function is constructed as follows:

$$V = \frac{1}{2} \Delta\mathbf{y}^T \mathbf{A} \Delta\mathbf{y} \quad (43)$$

where \mathbf{A} is a positive definite matrix. V is positive, becoming zero only when $\Delta\mathbf{y}$ is zero. Since \mathbf{y}_d is a constant, the time derivative of V is given by:

$$\dot{V} = -\Delta\mathbf{y}^T \mathbf{A} \dot{\mathbf{y}} \quad (44)$$

Using Equation (32), the following equation can be written for $\dot{\mathbf{y}}$:

$$\dot{\mathbf{y}} = [\mathbf{G}(\Theta, \mathbf{q}), \mathbf{I}]^T \dot{\mathbf{q}} = \mathbf{K}(\Theta, \mathbf{q}) \dot{\mathbf{q}} \quad (45)$$

where \mathbf{I} an $N \times N$ identity matrix, and \mathbf{K} is an $(N+3) \times N$ matrix. Combining Equations (44) and (45), V becomes:

$$\dot{V} = -\Delta y^T A K \dot{q} \quad (46)$$

and therefore, if one chooses \dot{q} according to:

$$\dot{q} = (A K)^T \Delta y \quad (47)$$

the time derivative of V is non-positive, since $A K (A K)^T$ is positive semidefinite:

$$\dot{V} = -\Delta y^T A K (A K)^T \Delta y \leq 0 \quad (48)$$

To guarantee that Δy will converge to zero, one must show that for nonzero Δy , \dot{V} is negative. However, as discussed in Reference [8], $(A K)^T$ has a null space, and Δy falls into it. In such a case, the motion stops, although Δy is nonzero.

To avoid this problem, a technique called the bi-directional approach has been proposed [8]. In this approach, two identical systems called 1 and 2, start at $t=0$ from the initial and final y . The error Δy is then defined as $y_1 - y_2$, where y_i corresponds to system i . If Δy is driven to zero, then system 1 follows in reverse the path followed by system 2, to reach the desired y_d . Although this approach may reduce the chance of being caught by the null space of K , it does not eliminate it, and hence such a method should be employed with caution.

VI. Path-planning in the Cartesian Workspace

The previous techniques can be used to find joint paths that either correct a spacecraft's orientation during a manipulator's motion, or simultaneously control the spacecraft orientation, and the manipulator's configuration. However, in many important applications, the system will operate under an Inertially-Referenced End-Point Motion Control mode [10]. Here, the primary task is to move the end-effector of the manipulator, from one inertial location to another. As was shown in Section IV, this may be a problem if the path has segments in the PDW. These problems become even more serious when a load is captured by the end-effector, because in such a case, the PIW is reduced [11]. To avoid these problems, either the workspace should be restricted to the PIW, or a planning technique that avoids dynamic singularities should be employed. In this section, one such technique is developed.

Assume that the task is to move the end-effector from point A to point D, without encountering dynamic singularities that will prevent reaching the destination point. Then, the following strategy can be used:

- (a) Start from the final desired spacecraft orientation and end-effector position/orientation, and move under joint space control to some point C of the PIW. Such a motion is not subject to the effects of dynamic singularities, because these affect the cartesian motion, only. Record the path taken. The system reaches point C with q_{DC} and Θ_{DC} .

(b) Start from the initial desired spacecraft orientation and end-effector position/orientation, and move under joint space control to some point B of the PIW. The system reaches point C with \mathbf{q}_{AB} and Θ_{AB} .

(c) Move from point B to point C, using any path. The system reaches point C with \mathbf{q}_{AC} and Θ_{AC} . In general, these are different than \mathbf{q}_{DC} and Θ_{DC} .

(d) Using small cyclical motions of the end-effector, change the spacecraft attitude from Θ_{AC} to Θ_{DC} . The configuration changes from \mathbf{q}_{AC} to \mathbf{q}_{DC} , since the end-effector does move around the same point in cartesian space.

(e) Use the recorded path during step (a), to move to point D.

The fact that small cyclical motions in the cartesian space can change a spacecraft's orientation is due to the following equation, obtained by combining Equations (10) and (32), and using an Euler angle representation for the end-effector orientation:

$$\delta\Theta = \mathbf{G}(\Theta, \mathbf{q}) \{ \text{diag}(\mathbf{I}, \mathbf{S}^{-1}(\Theta_E)) \mathbf{J}^* \}^{-1} \delta\mathbf{x}_E = \mathbf{G}^*(\Theta, \mathbf{x}_E) \delta\mathbf{x}_E \quad (49)$$

where $\delta\mathbf{x}_E = \delta[\mathbf{r}_E, \Theta_E]^T$ is a small change in the end-effector position/orientation. The 3×6 matrix \mathbf{G}^* is written as a function of Θ , and \mathbf{x}_E , because if these are given, and if $N=6$, then \mathbf{q} can be computed by inverting Equations (18) and (19). Note that Equation (49) has the same structure to Equation (35), though more complicated. Since \mathbf{J}^* is invertible in the PIW, \mathbf{G}^* exists and hence, closed paths in the cartesian space will result in changes in the orientation of a system's spacecraft. This technique is illustrated below by an example.

Example

Consider again the example system introduced in Section IV. The end-effector is initially at point A: $(x,y) = (2,0)$, which belongs in the system's PDW, see Figure 7. The initial configuration of the system is $(q_1, q_2) = (-58^\circ, 60.3^\circ)$ which corresponds to an initial spacecraft orientation $\theta = 21^\circ$. Assume that the end-effector is commanded to reach point D: $(x,y) = (1.5, 1.5)$. As the end-effector moves on a straight line from the initial to the desired location, a dynamic singularity occurs at point E where $(\theta, q_1, q_2) = (-32.4^\circ, 74.24^\circ, 10.6^\circ)$, see Figure 7. The end-effector stops at this point if an inverse Jacobian planning or control algorithm is used, or deviates from the desired final point if a transposed Jacobian control algorithm is used [10,11].

Next, the algorithm introduced above is applied. The task is to reach point D, with $\theta \approx 3^\circ$. This θ corresponds to $(q_1, q_2) = (39.4^\circ, 22.2^\circ)$. First, the end-effector is moved from the desired point D, to some PIW point C: $(0.8, 0.5)$, see path DC in Figure 7. Here a straight line motion is used, and C is reached with $\theta_{DC}=49.1^\circ$, see Figure 8. Next, the end-effector is moved from the initial point A, to point B, which for simplicity is taken equal to point C. The end-effector reaches point B: $(0.8, 0.5)$ with $(\theta, q_1, q_2) = (14.5^\circ, -49.4^\circ, 145.9^\circ)$. The next task is to change the orientation of

the spacecraft, from $\theta_{AC}=14.5^\circ$, to $\theta_{DC}=49.1^\circ$. To this end, the end-effector is commanded to follow 11 circular paths, with radius .2m, as shown in Figure 7.

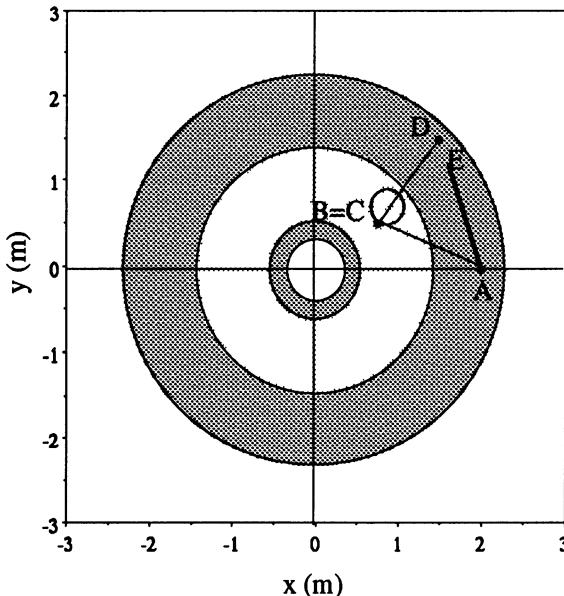


Figure 7. A Dynamic Singularity at point E does not allow the end-effector to move from point A to D. Path ABCD avoids singularities by employing small circles at point B.

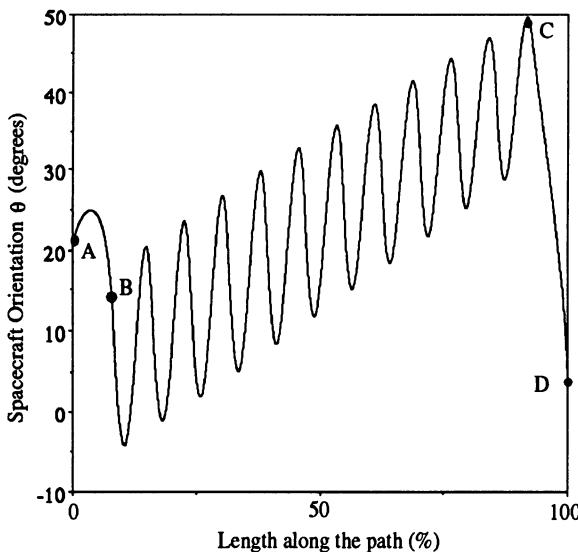


Figure 8. The orientation of the spacecraft θ as a function of the path ABCD, shown in Figure 7.

The required number of circles has been found by trial and error. After the execution of these circles, the orientation θ changes to 48.9° . Next, the end-effector is moved to D, following the prerecorded path DC in the opposite direction, and reaches D with $(\theta, q_1, q_2) = (3.3^\circ, 38.9^\circ, 22.7^\circ)$. Note that not only point D, but also a final spacecraft orientation quite close to the desired one, have been reached. If a closer match in orientation is required, a smaller circle radius and more circles should be employed. Figure 8 depicts the change of θ during as a function of the length of the total path ABCD.

VII. Conclusions

The kinematics and dynamics of free-floating manipulators were examined from a fundamental point of view. It was shown that the dynamic coupling between the uncontrolled spacecraft and its manipulator can make the system dynamically singular at configurations which cannot be predicted by the system's kinematic properties. The nonintegrability of the angular momentum introduces nonholonomic behavior. A workspace point can induce a singularity or not, depending on the path taken to reach it. Trouble-free Path Independent Workspaces were defined. Two planning techniques that use nonholonomy to control a spacecraft's orientation by manipulator joint motions were reviewed. It was shown that in some system configurations, joint manipulator motions cannot affect a spacecraft's orientation. Finally, a planning method was presented that permits the effective use of a system's reachable workspace by planning paths which avoid dynamically singular configurations.

References

- [1] Longman, R., Lindberg, R., and Zedd, M., "Satellite-Mounted Robot Manipulators-New Kinematics and Reaction Moment Compensation," *The Int. J. of Robotics Research*, Vol. 6, No. 3, pp. 87-103, Fall 1987.
- [2] Vafa, Z., and Dubowsky, S., "On the Dynamics of Space Manipulators Using the Virtual Manipulator, with Applications to Path Planning," *J. of Astr. Sciences*, Vol. 38, No. 4, October-December 1990, pp. 441-472.
- [3] Vafa, Z., and Dubowsky, S., "The Kinematics and Dynamics of Space Manipulators: The Virtual Manipulator Approach," *International Journal of Robotics Research*, Vol. 9, No. 4, August 1990, pp. 3-21.
- [4] Alexander, H., and Cannon, R., "Experiments on the Control of a Satellite Manipulator," *Proc. 1987 American Control Conf.*, Seattle, WA, June 1987.
- [5] Umetani, Y., and Yoshida, K., "Experimental Study on Two Dimensional Free-Flying Robot Satellite Model," *Proc. NASA Conf. on Space Telerobotics*, Pasadena, CA, January 1989.
- [6] Masutani, Y., Miyazaki, F., and Arimoto, S., "Sensory Feedback Control for Space Manipulators," *Proc. of the IEEE Int. Conf. on Robotics and Automation*, Scottsdale, AZ, May 1989.

- [7] Nakamura, Y., and Mukherjee, R., "Nonholonomic Path Planning of Space Robots," *Proc. of the IEEE Int. Conf. on Robotics and Automation*, Scottsdale, AZ, May 1989.
- [8] Nakamura, Y., and Mukherjee, R., "Nonholonomic Path Planning of Space Robots via Bi-directional Approach," *Proc. of the IEEE Int. Conf. on Robotics and Automation*, Cincinnati, OH, May 1990.
- [9] Papadopoulos, E., and Dubowsky, D., "On the Dynamic Singularities in the Control of Free-Floating Space Manipulators," *Proc. of the ASME Winter Annual Meeting*, San Francisco, ASME DSC-Vol. 15, December 1989.
- [10] Papadopoulos, E., and Dubowsky, S., "On the Nature of Control Algorithms for Free-floating Space Manipulators," *IEEE Trans. on Robotics and Automation*, Vol. 7, No. 6, December 1991, pp.750-758.
- [11] Papadopoulos, E., "On the Dynamics and Control of Space Manipulators," Ph.D. Thesis, Department of Mechanical Engineering, MIT, October 1990.
- [12] Hooker, W., and Margulies, G., "The Dynamical Attitude Equations for an n-Body Satellite," *J. of Astr. Sciences*, Vol XII, No 4, Winter 1965.
- [13] Wittenburg, J., *Dynamics of Rigid Bodies*, B.G. Teubner, Stuttgart, 1977.
- [14] Khatib, O., "A Unified Approach for Motion and Force Control of Robot Manipulators: The Operational Space Formulation," *IEEE Journal of Robotics and Automation*, Vol. RA-3, No. 1, February 1987, pp. 43-53.
- [15] Korn, G., and Korn, T., *Mathematical Handbook for Scientists and Engineers*, Second Edition, McGraw-Hill, New York, NY, 1968.
- [16] Hughes, P., *Spacecraft Attitude Dynamics*, John Wiley, New York, NY, 1986.

Appendix A

The general form of the barycentric vectors \mathbf{v}_{ik} ($i,k = 0 \dots N$) is [9,11]:

$$\mathbf{v}_{ik} \equiv \begin{cases} \mathbf{r}_i^* = -\mathbf{l}_i \sum_{j=0}^{i-1} \frac{m_j}{M} + \mathbf{r}_i \sum_{j=0}^i \frac{m_j}{M} & i < k \\ \mathbf{c}_i^* = -\mathbf{l}_i \sum_{j=0}^{i-1} \frac{m_j}{M} - \mathbf{r}_i \left(1 - \sum_{j=0}^i \frac{m_j}{M}\right) & i=k \\ \mathbf{l}_i^* = \mathbf{l}_i \left(1 - \sum_{j=0}^{i-1} \frac{m_j}{M}\right) - \mathbf{r}_i \left(1 - \sum_{j=0}^i \frac{m_j}{M}\right) & i > k \end{cases} \quad (A1)$$

where \mathbf{l}_i and \mathbf{r}_i are defined in Figure 1, m_i is the mass of body i , and M is the total system mass. Since \mathbf{l}_i and \mathbf{r}_i are body-fixed vectors, \mathbf{v}_{ik} are also body-fixed. If \mathbf{v}_{ik} are the barycentric vectors expressed in the i^{th} frame, then \mathbf{v}_{ik} are constant vectors, which can be computed only once. These are transformed in the inertial frame as follows:

$$\mathbf{v}_{ik} = \mathbf{T}_0^0 \mathbf{v}_{ik} = \mathbf{T}_0^0 \mathbf{T}_i^i \mathbf{v}_{ik} \quad (A2)$$

The vectors $\dot{\mathbf{v}}_{iN}$ used in Equation (1) are obtained by setting $k=N$.

The matrices required for the construction of the Jacobian \mathbf{J}^* , see Equation (11), are functions of the vectors $\dot{\mathbf{v}}_{iN,E} = \dot{\mathbf{v}}_{iN} + \delta_{iN} \dot{\mathbf{r}}_N$:

$${}^0\mathbf{J}_{11} \equiv -\sum_{i=0}^N [{}^0\mathbf{T}_i \dot{\mathbf{v}}_{iN,E}]^x \quad {}^0\mathbf{J}_{12} \equiv -\sum_{i=1}^N [{}^0\mathbf{T}_i \dot{\mathbf{v}}_{iN,E}]^x {}^0\mathbf{F}_i \quad {}^0\mathbf{J}_{22} \equiv {}^0\mathbf{F}_N \quad (\text{A3})$$

where ${}^0\mathbf{F}_i$ ($i = 1, \dots, N$) are defined by Equation (5). The x symbol operates on a column vector \mathbf{e} , to form a skew-symmetric matrix which corresponds to a cross product:

$$\mathbf{e}^x = \begin{bmatrix} 0 & -e_z & e_y \\ e_z & 0 & -e_x \\ -e_y & e_x & 0 \end{bmatrix} \quad (\text{A4})$$

The mixed inertia matrices \mathbf{D}_{ij} are also functions of the barycentric vectors [10,11]:

$$\mathbf{D}_{ij} = \begin{cases} -M\{ \mathbf{l}_j^{*T} \mathbf{r}_i^* \} - \mathbf{l}_j^* \mathbf{r}_i^{*T} & i < j \\ \mathbf{I}_i + \sum_{k=0}^N m_k \{ \mathbf{v}_{ik}^2 - \mathbf{v}_{ik} \mathbf{v}_{ik}^T \} & i=j \\ -M\{ \mathbf{l}_j^* \mathbf{l}_i^* \} - \mathbf{r}_j^* \mathbf{l}_i^{*T} & i > j \end{cases} \quad (\text{A5})$$

where \mathbf{v}_{ik} , \mathbf{r}_i^* , and \mathbf{l}_i^* are defined by Equation (A1). These mixed inertia matrices are transformed to a spacecraft's frame according to the following formula:

$$\mathbf{D}_{ij} = \mathbf{T}_0 {}^0\mathbf{D}_{ij} {}^0\mathbf{T}_0^T \quad i, j = 1, \dots, N \quad (\text{A6})$$

For simplicity, the following definitions are used:

$${}^0\mathbf{D}_j \equiv \sum_{i=0}^N {}^0\mathbf{D}_{ij} \quad j = 0, \dots, N \quad {}^0\mathbf{D} \equiv \sum_{j=0}^N {}^0\mathbf{D}_j \quad (\text{A7a})$$

$${}^0\mathbf{D}_q \equiv \sum_{j=1}^N {}^0\mathbf{D}_j {}^0\mathbf{F}_j \quad {}^0\mathbf{D}_{qq} \equiv \sum_{j=1}^N \sum_{i=1}^N {}^0\mathbf{F}_i^T {}^0\mathbf{D}_{ij} {}^0\mathbf{F}_j \quad (\text{A7b})$$

Appendix B

For planar systems, the inertia matrices ${}^0\mathbf{D}_{ij}$ in Equations (A7) reduce to scalars ${}^0d_{ij}$ which are written as:

$${}^0d_{00} = I_0 + \frac{m_0(m_1+m_2)}{M} r_0^2$$

$${}^0d_{10} = \frac{m_0 r_0}{M} \{ l_1(m_1+m_2) + r_1 m_2 \} \cos(q_1) = {}^0d_{01}$$

$$\begin{aligned}
 {}^0d_{20} &= \frac{m_0 m_2}{M} r_0 l_2 \cos(q_1 + q_2) = {}^0d_{02} \\
 {}^0d_{11} &= I_1 + \frac{m_0 m_1}{M} l_1^2 + \frac{m_1 m_2}{M} r_1^2 + \frac{m_0 m_2}{M} (l_1 + r_1)^2 \\
 {}^0d_{21} &= \left\{ \frac{m_1 m_2}{M} r_1 l_2 + \frac{m_0 m_2}{M} l_2 (l_1 + r_1) \right\} \cos(q_2) = {}^0d_{12} \\
 {}^0d_{22} &= I_2 + \frac{m_2(m_0 + m_1)}{M} l_2^2
 \end{aligned} \tag{B1}$$

The mixed inertia sum defined by Equations (A7) become:

$$\begin{aligned}
 {}^0D_j &\equiv D_j = \sum_{i=0}^2 {}^0d_{ij} \quad (j=0,1,2) & {}^0D &\equiv D = D_0 + D_1 + D_2 \\
 {}^0D_q &= [D_1 + D_2 \quad D_2] & {}^0D_{qq} &= \begin{bmatrix} {}^0d_{11} + 2{}^0d_{12} + {}^0d_{22} & {}^0d_{12} + {}^0d_{22} \\ {}^0d_{12} + {}^0d_{22} & {}^0d_{22} \end{bmatrix}
 \end{aligned} \tag{B2}$$

INDEX

A

abnormal extremals, 364
accessibility, 19
angular momentum, 356, 380, 429
approximation technique, 109, 112, 121
asymptotic behavior, 58, 68
averaging, 56, 68, 109

B

Baillieul, 6
basis algorithm, 389, 401
Bloch, 201, 202, 381
Brockett 1, 27, 33, 55, 73, 202,
236, 381
bundles, 351

C

Campbell-Haussdorf-Baker
formula, 176
Canny, 2, 237, 271
canonical form, 5, 34, 36
canonical trajectories, 276
Caplygin dynamical system, 201
chained system, 39, 45
Chow's theorem, 24
collision constraint, 54, 56
connection, 351, 352
controllability, 1, 5, 40, 46, 75,
85, 153, 168, 204, 360, 429
controllability algorithm, 162
curvature, 360

D

Dai, 1, 381
degree on nonholonomy, 25, 75, 158
distribution, 24, 204, 353

E

Elliptic functions, 1, 12
Euler-Lagrangian equation, 10, 16,
28, 31
exact differentials, 4, 27
extended inputs, 58, 61, 64, 115
extended system, 57, 237

F

factorization, 73, 117
falling cat, 379
feedback algorithm, 57, 60, 143
feedback stabilization, 371
feedback transformation, 44, 48, 246
Fernandes, 53, 379
filtration, 25, 157, 158
Frobenius Theorem, 155
front wheel drive cart, 32

G

gauge theory, 343
grid based algorithm, 304, 308
Gurvits, 53, 55, 379

H

holonomic path, 57, 58, 184

I

involutive closure, 24

J

Jacobs, 153, 185, 271

K

Krishnaprasad, 213

L

Lafferriere, 2, 44, 55, 153, 156, 212,
235, 381
Lagrangian function, 27
Laumond, 149
Li, 2, 29 53, 55, 202, 237, 379, 381

Lie algebra, 35, 117, 236

Lie algebra rank condition, 112,
236, 390

Lie bracket, 3, 24, 40, 51, 109, 155

Liu, 109, 184

M

McClamroch, 201, 381
mobile robots, 186, 271
Montana, 2
Montgomery, 27, 55, 213, 343
Murray, 2, 23, 55, 160, 212, 237, 343

N

- nilpotent systems, 36, 167, 246
- nonholonomic constraint, 1, 23, 53, 149, 201, 343, 379, 423

O

- obstacle avoidance, 48, 175, 179, 283
- open loop, 48
- optimal control, 16, 363
- optimization, 9

P

- Papadopoulos, 423
- parallel transport, 358
- Phillip Hall basis, 6, 35, 140, 159, 160, 184, 240
- plate-ball problem, 7

R

- regular system, 25
- Reyhanoglu, 201, 381
- Ritz approximation, 389

S

- Sastry, 2, 23, 55, 153, 212, 236, 381
- shape space, 345
- sinusoids, 23, 26, 30, 178
- space manipulators, 423
- sub-Riemannian metric, 27, 153, 180, 181
- Sussmann, 2, 5, 44, 55, 109, 156, 184, 212, 235, 381
- symmetry, 14

T

- time-periodic feedback, 53, 58, 86
- topological aspects, 149

U

- universal joint, 387, 404

V

- Vershik, 25