

Geometric Phases, Control Theory, and Robotics

Richard M. Murray*

Department of Control and Dynamical Systems and
Department of Mechanical Engineering
California Institute of Technology
Pasadena, California 91125
murray@indra.caltech.edu

1 May 1994

Abstract

Differential geometry and nonlinear control theory provide essential tools for studying motion generation in robot systems. Two areas where progress is being made are motion planning for mobile robots on the factory floor (or on the surface of Mars), and control of highly articulated robots—such as multifingered robot hands and robot “snakes”—for medical inspection and manipulation inside the gastrointestinal tract. A common feature of these systems is the role of constraints on the behavior of the system. Typically, these constraints force the instantaneous velocities of the system to lie in a restricted set of directions, but do not actually restrict the reachable configurations of the system. A familiar example in which this geometric structure can be exploited is parallel parking of an automobile, where periodic motion in the driving speed and steering angle can be used to achieve a net sideways motion. By studying the geometric nature of velocity constraints in a more general setting, it is possible to synthesize gaits for snake-like robots, generate parking and docking maneuvers for automated vehicles, and study the effects of rolling contacts on multifingered robot hands. As in parallel parking, rectification of periodic motions in the control variables plays a central role in the techniques which are used to generation motion in this broad class of robot systems.

*Research supported in part by grants from the Powell Foundation, the National Science Foundation, and NASA.

1 Introduction

The earliest robots consisted of simple electro-mechanical devices which could be programmed to perform a limited set of tasks. They were a cross between numerically controlled milling machines and the master-slave tele-operators developed for handling radioactive material. These robots are the precursors to the automated machines used for painting, welding, and pick and place operations in today's factories. For these types of automation and manufacturing tasks, the complexity of the robot can be minimized since the workspace of the robot can be carefully controlled and the robots are not required to perform particularly dextrous manipulation of objects. However, many future applications of robotics are moving toward more autonomous operation in highly uncertain environments. The robots being developed for these applications are increasingly complex and have a high degree of interaction with their environment.

Examples of the next generation of robots range from miniature robots for medical inspection and manipulation inside the human body to mobile robots for exploration in hazardous and remote environments. These robots will require sensing, actuation, and computational capabilities which were unheard of just a few years ago. However, as we seek to design robots which can act with increasing autonomy, we move closer to endowing robots with human-like capabilities. And we begin to become limited by our own ability to understand and analyze the highly complex systems which we are trying to control.

As the complexity of robots increases, so does the importance of abstraction and theory in understanding and analyzing robot motion. One approach which has begun to yield new insights is the use of differential geometry, in the context of both geometric mechanics and nonlinear control theory. Two specific areas where progress is being made are locomotion and manipulation in robot systems.

Locomotion is defined as the act of moving from one place to another. For robots, there are several mechanisms by which this movement can occur. The use of wheels and legs are the two traditional methods, but other possibilities, such as undulatory gaits in snake-like robots, have also been proposed and implemented. Each of these mechanisms has certain advantages over the others, but all of them fundamentally involve interaction with their environment: locomotion is achieved by pushing or sliding or rolling or a combination of all of these.

Robotic manipulation involves motion of an object rather than motion

of the robot itself. The prototypical example is a multifingered hand manipulating a grasped object. Once again, the fundamental mechanisms which govern motion involve pushing and rolling and sliding. The motion of a set of fingers grasping an object is constrained in much the same way as the motion of a legged robot is constrained by the contacts between its feet and the ground. Indeed, many of the tools which are used to analyze manipulation and grasping problems are easily adapted to analyze locomotion.

The most basic problem in all locomotion and manipulation systems is to devise a method for generating and controlling motion between one configuration and another. The common feature is that motion of the robot is constrained by its interactions with the environment. For example, in wheeled mobile robots the wheels must roll in the direction in which they are pointing and are not allowed to slide sideways. In grasping, the motion of the fingers is constrained by the object being held in the grasp: motion of one finger affects the others since forces are transmitted between the fingers by the object. Even in legged robots, one usually assumes that the feet do not slip on the ground, allowing the robot to propel itself. These constraints on the motion of the system are the defining features for how locomotion and manipulation work in these systems.

Furthermore, in most locomotion and manipulation systems, the range of the actuators is small, while the desired net motion for the system may be large. A good example of this is using your fingers to screw in a light bulb: repeated grasping and twisting of the bulb is required in order to fully insert it into the socket. A large motion of the light bulb (multiple revolutions) is accomplished by repeated (i.e. periodic) small motions in your fingers.

In this paper, we consider locomotion and manipulation using the notion of geometric phases as a central theme. Intuitively, geometric phases relate the motion of one parameter describing the configuration of a system to other parameters which undergo periodic motion. A simple example of geometric phase is the motion of an automobile performing a parallel parking maneuver. By moving the car backwards and forwards and turning the steering wheel in a periodic fashion, a driver is able to achieve a net sideways motion of the car even though the car cannot move sideways directly. This net sideways motion is the geometric phase associated with this choice of the car velocity and steering wheel angle.

The role of geometric phases as a means of analyzing locomotion is a relatively new perspective. One of the earliest works is that of Shapere and Wilczek [17], who studied the motion of paramecia swimming in a highly viscous fluid. They show that periodic variations in the shape of an organism

can be used to achieve net forward motion. This is very reminiscent of the type of motion present in parallel parking and this similarity can be made precise by using geometric phases.

There has also been an increased interest in the use of geometric phases for understanding motion in other biological systems, such as snakes and insects. Here again, periodic changes in one set of variables, which describe the shape of the system, are used to obtain net motion. The phasing of the inputs plays a central role, generating different gaits for achieving different types of motion. The interpretation of locomotion in terms of geometric phases is still far from complete, but it is providing a unifying view of locomotion and manipulation which has already yielded new insights and has impact on several challenging applications.

2 Locomotion in mobile robots

Locomotion involves movement of a mechanical system by appropriate application of forces on the robot. These forces can arise in several ways, depending on the means of locomotion used. The simplest form of locomotion is to apply the forces directly, such as done in a spacecraft, where high-energy mass is ejected in the opposite direction to the desired motion. A similar technique is the use of jet engines on modern aircraft.

For ground-based systems, a much more common means of locomotion is the use of forces of constraint between a robot and its environment. For example, a wheeled mobile robot exerts forces by applying a torque to its drive wheels. These wheels are touching the ground and, in the presence of sufficient friction, are constrained so as not to slip along the ground. This constraint is enforced by the application of internal forces, which cause a net force on the robot that propels it forward. If no constraints existed between the robot and the ground, then the robot would just spin its wheels. Similarly, for legged and snake robots, the parts of the robots in contact with the environment are used to exert net forces on the robot. In fact, for a large class of robotic systems we can view constraints as the basis for locomotion.

A second common feature in robot locomotion is the notion of base (or internal) variables versus fiber (or group) variables. Base variables describe the geometry and shape of the robot, while fiber variables describe its configuration relative to its environment. For example, in a snake robot the fiber variables might be the position and orientation of a coordinate frame fixed to the robot's body, while the base variables would be the angles which

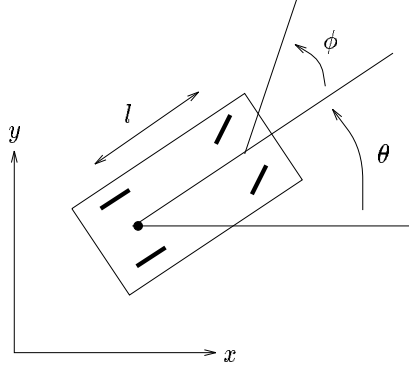


Figure 1: Kinematic model of an automobile. The configuration of the car is determined by the Cartesian location of the back wheels, the angle the car makes with the horizontal and the steering wheel angle relative to the car body. The two inputs are the velocity of the rear wheels and the steering velocity.

describe the overall shape of the robot. These base and fiber variables are coupled by the constraints acting on the robot. Hence, by making changes in the base variables, it is possible to effect changes in the fiber variables.

In this paper, we concentrate on a particular type of constraint on the configuration variables of the robot, known as a *Pfaffian* constraint. Consider a mechanical system with configuration space $Q = \mathbb{R}^n$ and configuration $q \in Q$. A Pfaffian constraint restricts the motion of the system according to the equation

$$\omega(q)\dot{q} = 0,$$

where $\omega(q)$ is a row vector which gives the direction in which motion is not allowed. Pfaffian constraints arise naturally in wheeled mobile robots: they model the ability of a wheel to roll along the ground and spin about its vertical axis, but not slide sideways. Pfaffian constraints typically do not provide a complete model of the interaction with the environment, since frictional forces are present in both rolling and spinning, but they do capture the basic behavior of the system.

As an example, consider a simple kinematic model of an automobile, as shown in Figure 1. The constraints are derived by assuming that the front and rear wheels can roll and spin (about the center of the axle) but not slide. Let $q = (x, y, \theta, \phi) \in \mathbb{R}^4$ denote the configuration of the car, parameterized by the xy location of center of the rear axle, the angle of the car body with

respect to the horizontal, θ , and the steering angle with respect to the car body, ϕ . To simplify the derivation, we model the front and rear pairs of wheels as single wheels at the midpoints of the axles. The constraints for the front and rear wheels are formed by setting the sideways velocity of the wheels to be zero. A simple calculation shows that the Pfaffian constraints are given by

$$\begin{aligned}\omega_1(q)\dot{q} &= \sin\theta\dot{x} - \cos\theta\dot{y} = 0 \\ \omega_2(q)\dot{q} &= \sin(\theta + \phi)\dot{x} - \cos(\theta + \phi)\dot{y} - l\cos\phi\dot{\theta} = 0.\end{aligned}$$

For simplicity we take $l = 1$ in the sequel.

To study the motion of a system subject to a set of Pfaffian constraints $\{\omega_1, \dots, \omega_k\}$, it is convenient to convert the problem to a control problem. Roughly speaking, we would like to shift our viewpoint from describing the directions in which the system cannot move to describing those in which it can. Formally, we choose a basis for the right null space of the constraints, denoted by $g_i(x) \in \mathbb{R}^n$, $i = 1, \dots, n - k$. The locomotion problem can be restated as finding an input function, $u(t) \in \mathbb{R}^{n-k}$, such that the control system

$$\dot{x} = g_1(x)u_1 + \dots + g_{n-k}(x)u_{n-k}$$

achieves a desired motion. The g_i 's can be regarded as vector fields on \mathbb{R}^n which describe the allowable motion of the system. This type of control system is called a *driftless* control system since setting the inputs to zero stops the motion of the system.

For the kinematic car, this conversion yields the control system

$$\frac{d}{dt} \begin{bmatrix} x \\ y \\ \theta \\ \phi \end{bmatrix} = \begin{bmatrix} \cos\theta \\ \sin\theta \\ \tan\phi \\ 0 \end{bmatrix} u_1 + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} u_2.$$

For this choice of vector fields, u_1 corresponds to the forward velocity of the rear wheels of the car and u_2 corresponds to the velocity of the steering wheel.

The first question one must consider when analyzing a control system is whether the system is *controllable*. That is, given an initial state x_i and a final state x_f , does there exist a choice of inputs u which will steer the system from one state to the other. A geometric interpretation of this question can be formulated by studying the properties of the vector fields which define the

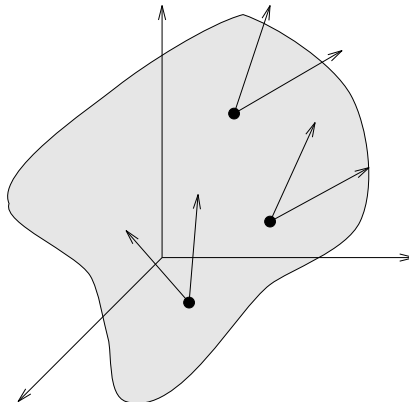


Figure 2: A set of vector fields which are tangent to a hypersurface in the configuration space. The system described by this set of vector fields is not controllable since motion is restricted to a hypersurface.

control system. A sets of vector fields which is *not* controllable is shown in Figure 2. For these vector fields, there exists a surface whose tangent space contains the span of the vector fields. Hence, any motion of the system is necessarily restricted to this surface and it is not possible to move to an arbitrary point in the configuration space (only other points on the same surface).

To test whether a set of vector fields are tangent to some surface, we make use of a special type of motion called a *Lie bracket motion*. Roughly, the idea is to choose two vector fields, say g_1 and g_2 , and construct an infinitesimal motion by first flowing along g_1 for ϵ seconds, then flowing along g_2 for ϵ seconds, and then flowing backwards along g_1 for ϵ seconds and backwards along g_2 for ϵ seconds. This motion is illustrated in Figure 3. A simple Taylor series argument shows that the net motion given by this strategy is

$$q(4\epsilon) = q_0 + \epsilon^2 \left(\frac{\partial g_2}{\partial q} g_1(q_0) - \frac{\partial g_1}{\partial q} g_2(q_0) \right) + O(\epsilon^3)$$

(see Murray et al. [11], pp. 323–324 for a derivation). Motivated by this calculation, we define the *Lie bracket* of two vector fields g_1 and g_2 as

$$[g_1, g_2] = \frac{\partial g_2}{\partial q} g_1 - \frac{\partial g_1}{\partial q} g_2. \quad (1)$$

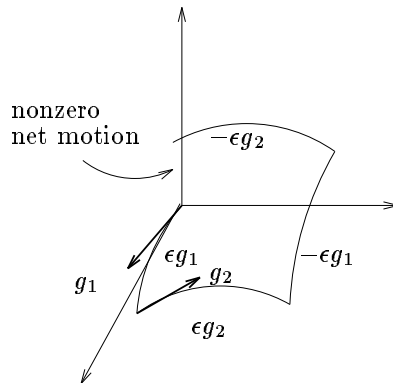


Figure 3: A Lie bracket motion.

The Lie bracket of g_1 and g_2 describes the infinitesimal motion due to cycling between the inputs corresponding to g_1 and g_2 .

The Lie bracket between two vector fields gives a potentially new direction in which we can move. In particular, if $[g_1, g_2]$ is not in the span of g_1 and g_2 , then it is not possible for g_1 and g_2 to be locally tangent to a two dimensional surface, since we could move off of such a surface by executing a Lie bracket motion. Furthermore, from a controllability point of view we can treat $g_3 = [g_1, g_2]$ as a new direction in which we are free to move, and we can look at higher order bracket motions involving g_3 . A fundamental result, proven in the 1940s by the German mathematician C.-L. Chow [4], is the central result in controllability for control systems of this type. Let $\bar{\Delta}_q$ be the set of all directions which can be achieved by the input vector fields or repeated Lie brackets. That is,

$$\bar{\Delta}_q = \text{span}\{g_i(q), [g_i, g_j](q), [g_i, [g_j, g_k]](q), \text{etc}\}.$$

Theorem 1 (Chow) *A driftless control system is controllable in a neighborhood of $q \in \mathbb{R}^n$ if $\bar{\Delta}_q = \mathbb{R}^n$.*

We can verify that the kinematic car satisfies Chow's theorem by direct calculation. The input vector fields are given by

$$g_1 = \begin{bmatrix} \cos \theta \\ \sin \theta \\ \tan \phi \\ 0 \end{bmatrix} \quad g_2 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}.$$

We call g_1 the *drive* vector field, corresponding to the motion commanded by the gas pedal, and g_2 the *steer* vector field, corresponding the motion of the steering wheel.¹ The Lie bracket between the drive and steer vector fields turns out to be

$$g_3 = [g_1, g_2] = \begin{bmatrix} 0 \\ 0 \\ -\sec^2 \phi \\ 0 \end{bmatrix}.$$

We call g_3 the *wriggle* vector field; it gives infinitesimal rotation of the car about the center of the rear wheels. Finally, we compute the Lie bracket between drive and wriggle, which yields

$$g_4 = [g_1, [g_1, g_2]] = \begin{bmatrix} -\sin \theta \sec^2 \phi \\ \cos \theta \sec^2 \phi \\ 0 \\ 0 \end{bmatrix}.$$

This vector field is called the *slide* vector field since it corresponds to motion perpendicular to the direction in which the car is pointing.

The four vector fields $g_1, g_2, g_3,$ and g_4 span \mathbb{R}^4 as long as $\phi \neq \pm\pi/2$ (at which point g_1 is not defined). By Chow's theorem this means that we can steer between any two configurations by an appropriate choice of input.

What Chow's theorem does not tell us is how to *synthesize* an input which causes the car to move to a given location. Of course, humans are very good at synthesizing trajectories for automobiles, but for more complicated situations (such as backing a truck with two or three trailers into a loading dock), the solution to the locomotion problem is not so intuitive.

One method for synthesizing trajectories is to use the Lie bracket motions described earlier. The problem is that these motions are only piecewise smooth in the inputs (a problem here since we are commanding velocities) and they only generate infinitesimal motions. A partial solution to these issues was explored in [12], where we suggested the use of sinusoids to steer control systems of this form. The basic idea was to use sinusoids at integrally related frequencies to generate motion in the Lie bracket directions. In essence, one replaces the squares of a Lie bracket motion by circles. By varying the relative frequencies of the inputs, motion corresponding to different combinations of brackets between the inputs can be obtained. These

¹These names for the vector fields are due to Nelson [13] who considered this system as part of an example in a book on differential geometry.

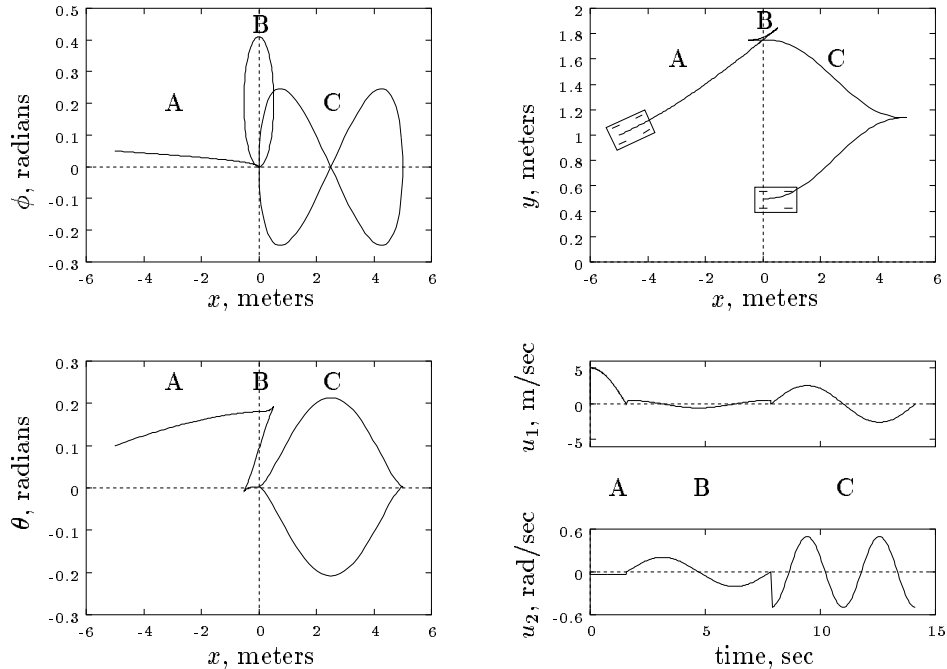


Figure 4: Motion of a kinematic car. The trajectory shown is a sample path which moves the car from $(x, y, \theta, \phi) = (-5, 1, 0.05, 1)$ to $(0, 0.5, 0, 0)$. The first three figures show the states versus x , the bottom right graphs show the inputs as functions of time. (Adapted from [12])

calculation were motivated by results of Brockett [1], who showed that under certain conditions these types of inputs are actually optimal.

An example of this type of motion is shown in Figure 4. The input, shown in the lower right, consists of a sequence of sinusoidal input segments with different frequencies. The first part of the path, labeled A, drives x and ϕ to their desired values using a constant input. These are the states controlled directly by the inputs, so no periodic motion is needed.

The second portion, labeled B, uses a sine and cosine to drive θ while bringing the other two states back to their desired values. Thus, choosing $u_1 = a \sin t$ and $u_2 = b \cos t$ gives motion in the wriggle direction, $[g_1, g_2]$. By choosing a and b properly, we can control the net change in orientation. However, a careful examination of the motion reveals that some motion also occurs in the y direction. This is due to the higher order terms which appear in the expression for a Lie bracket motion. However, the input directions, x

and ϕ return to their original values.

The last step, labeled C, uses the inputs $u_1 = a \sin t$ and $u_2 = b \sin 2t$ to steer y to the desired value and returns the other states back to their correct values. This choice of inputs moves the car back and forth once while rotating the steering wheel twice in the proper phase. It generates motion in the bracket direction corresponding to *slide*, $g_4 = [g_1, [g_1, g_2]]$. Notice that the Lie bracket expression contains two copies of g_1 and one copy of g_2 , while the inputs move twice in u_2 versus once in u_1 .

The Lissajous figures obtained from the phase portraits of the different variables are quite instructive. Consider the part 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 interesting implication here is that the Lie bracket motions correspond to rectification of harmonic periodic motions of the driving vector fields, and the harmonic relations are determined by the order of the Lie bracket corresponding to the desired direction of motion.

It is instructive to reinterpret this example in terms of geometric phases. To do this, we rewrite the equations of motion for the system as

$$\begin{aligned} \dot{r}_1 &= u_1 & \dot{s}_1 &= \cos s_3 \dot{r}_1 \\ \dot{r}_2 &= u_2 & \dot{s}_2 &= \sin s_3 \dot{r}_1 \\ & & \dot{s}_3 &= \tan r_2 \dot{r}_1. \end{aligned}$$

Note that the right-hand set of equations have the form of a set of Pfaffian constraints. One can directly verify that these equations describe the motion of the system with identification of (s_1, s_2, s_3) with (x, y, θ) and u_1 and u_2 with the driving and steering and velocity. The variable r_1 represents the (signed) distance traveled by the car and r_2 the angle of the steering wheel.

The decomposition of the problem into a set of independent variables, $r \in \mathbb{R}^2$, and dependent variables, $s \in \mathbb{R}^3$ is an example of a fiber bundle decomposition of the system. We call $r \in \mathbb{R}^2$ the base variables and we call $s \in \mathbb{R}^3$ the fiber variables. Looking back at Figure 4, we see that the motion of the fiber variables in segments *B* and *C* is obtained by using closed loops in the base variables. The amount of motion in the fiber variables due to a trajectory in the base variables is the geometric phase associated with the path in the base space. Parallel parking corresponds to a phase shift in the y direction, while making a U-turn corresponds to a phase shift in the θ direction (sometimes combined with y).

The trajectories shown in Figure 4 show how geometric phases can be used to understand car parking, but they are not very good examples of parallel parking maneuvers. In fact, it is possible to get much better trajectories for this system by using some obvious tricks, such as using the sum of a set of sinusoids instead of applying simple periodic inputs in a piecewise fashion. One can even solve for the optimal trajectories in simple examples such as this one. Reeds and Schepp [15] showed that optimal trajectory between any two configurations can be obtained by following a path consisting of up to five segments consisting of either straight, hard left, or hard right driving in either forward or reverse. Furthermore they were able to show that no more than two backups are required and that only 48 different input patterns are needed to construct minimum length paths (this number has since been reduced to 46 by Sussman and Tang [18]).

3 Grasping and manipulation

A somewhat more complicated example of geometric phases occurs in the area of dextrous manipulation using multifingered robot hands. Here the basic issues involve the use of phases for repositioning the fingers of a hand without removing the fingers from the object. In fact, one is usually more interested in making sure that geometric phase is *not* generated, so that periodic motions of the object cause the fingers to return to their original positions. We start by discussing the repositioning problem and then make some brief comments about generating cyclic motions.

Consider the grasping control problem with rolling contacts, such as the system shown in Figure 5. Assuming that the fingers roll without slipping on the surface of the object (a very useful assumption to enforce, since control of sliding is very tricky), the constraints on the system can be described by a set of equations of the form

$$J_h(\theta, x)\dot{\theta} = G^T(\theta, x)\dot{x}, \quad (2)$$

where θ is the vector of finger joint angles and x specifies the position and orientation of the grasped object. In the robotics literature, J_h is called the *hand Jacobian* and G is the *grasp map* (see [11] for a detailed discussion).

A detailed calculation of the hand Jacobian and grasp map is quite involved and makes use of a large amount of specialized machinery. However, the basic idea behind the grasp constraint is quite straightforward: the left hand side of equation (2) is the vector of fingertip contact velocities for the

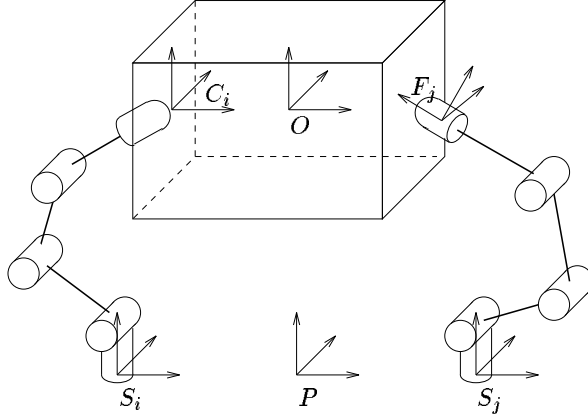


Figure 5: Multi-fingered hand grasping an object [11].

robot hand, expressed in an appropriate frame of reference. The right hand side of equation (2) is the vector of velocities for the contact points on the object, expressed in the same frame of reference. The condition that the fingers roll without slipping is obtained by equating these sets of velocities.

If the fingers of a grasp have sufficient dexterity, they can follow any motion of the object without slipping. A grasp of this type is called a *manipulable grasp*. For manipulable grasps, any object velocity \dot{x} can be accommodated by some finger velocity vector $\dot{\theta}$. However, the vector $\dot{\theta}$ may not be unique in the case that the null space of J_h (that is, the set of vectors that J_h maps to the zero vector) is nontrivial. This situation corresponds to the existence of internal motions of the fingers that do not affect the motion of the object. If we let u_1 be an input which controls the velocity of the object and let u_2 parameterize the internal motions, then equation (2) can be written as

$$\begin{aligned}\dot{x} &= u_1 \\ \dot{\theta} &= J_h^+ G^T u_1 + K u_2,\end{aligned}\tag{3}$$

where $J_h^+ = J_h^T (J_h J_h^T)^{-1}$ is the right pseudo-inverse of J_h and K is a matrix whose columns span the null space of J_h .

Equation (3) describes the grasp kinematics as a control system. The input u_1 describes the motion of the object, whose position is given by x . The effect of u_1 on θ describes how the fingers must move in order to maintain contact with the object. If the fingers have any extra degrees of freedom, u_2 can be used to control the internal motions which affect the shape of

the fingers but leave the object position unchanged. The dynamic finger repositioning problem is to steer the system from an initial configuration (θ_0, x_0) to a desired final configuration (θ_f, x_f) . The explicit location of the fingertip on the object at the initial and final configurations can be found by solving the forward kinematics of the system.

The general case of finding $u_1(t)$ and $u_2(t)$ such that the object and the fingers move from an initial to final position (while maintaining contact) can be very difficult. A special case is when the object position is kept fixed by setting u_1 to be zero. If there are sufficient degrees of freedom available for the fingers to still roll on the object, then the internal motions parameterized by u_2 can be used to move the fingers individually around on the object. With the object position held fixed, each of the fingers can be controlled individually without regard to the motion of the others, simplifying the problem somewhat.

An example of such a path moving a spherical fingertip down the side of a planar object is shown in Figure 6. In this figure, we consider the motion of a finger with a spherical tip on a rectangular object. The plots show trajectories which move a finger down the side of the object. The location of the contact on the finger is unchanged as shown in the right graph which plots the finger contact configurations (u_f, v_f) , while the location of the contact on the face of the object (u_o, v_o) undergoes a displacement in the v_o direction.

In addition to describing how the fingers can be repositioned on the object without releasing contact, geometric phases can also be used to understand when control laws keep the fingers from drifting in the case that this is not desired. Imagine, for example, performing a complicated manipulation of an object. Depending on the geometry of the object and fingers, it is possible that the object might return to its starting configuration while the fingers would have shifted to a new location. In many cases we are interested in choosing control laws to insure that this *doesn't* happen. Thus we want to control the fingers in such a way that there is no geometric phase associated with any closed loop motions of the controls.

A controller which always returns the fingers to their original configuration when the object does is said to be *cyclic*. The same type of problem can occur in resolving motion in redundant robots and has been studied a great deal in that context. In terms of the point of view described in this paper, the work of Shamir and Yomdin [16] gives necessary and sufficient conditions in terms of Lie brackets which guarantee that a controller is cyclic. The same basic ideas can be used in the grasping case.

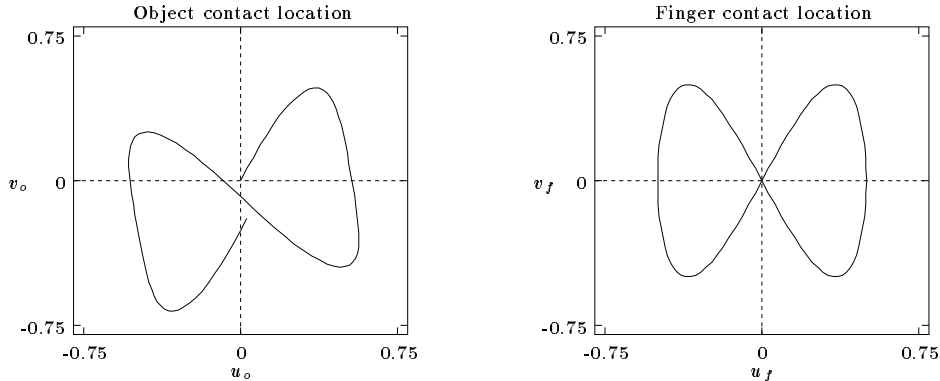


Figure 6: Steering applied to the multifingered hand shown in Figure 5. The left plot shows the location of the contact point on the object and the right plot shows the corresponding contact point on the finger. The object contact point moves down and slightly to the right, so the object is shifted slightly in the grasp after executing the steering maneuver.

Using the kinematics described in equation (3), the input u_1 describes the motion of the object. To make the overall controller cyclic, we must choose u_2 so that any cyclic motion of u_1 gives a cyclic motion in θ . Thus, we wish to choose a feedback law $u_2 = \alpha(x, \theta)$ such the geometric phase associated with u_1 is always zero.

While dynamic finger repositioning provides a direct connection between geometric phases and manipulation, the basic ideas behind phases are present in other types of manipulation as well. Consider the problem of inserting a light bulb into a threaded hole using your fingers. One way to do this would be to grab the bulb and then walk around in circles to insert it. This obviously requires a large workspace and is completely inappropriate for manipulation inside a cluttered environment (like the inside of a refrigerator).

A much more natural way to insert the bulb is to rotate your fingers, release the bulb, and then move your fingers back to perform another rotation. If we treat "twisting" and "grasp/release" as inputs, then this type of motion corresponds exactly to a Lie bracket motion where the bracket direction corresponds to the motion of the bulb into the socket. The description of this problem does not quite fit into the differential framework described above without some modification of the underlying mathematics, but the basic notion of a Lie bracket motion is still present.

Another example along these lines is using a (miniature) multi-fingered hand for sewing stitches in tissue. To understand how such a hand should be designed and how such a task might be accomplished, we can use the tools from geometric phases to guide our insights and uncover the fundamental principles which govern the behavior of the system. Since these systems tend to be highly complex, it is essential to understand the essential geometry of the system and its role in satisfying the overall task. The implications of some of these ideas in areas such as medicine is discussed in Section 4.2.

4 Applications

We now discuss two specific applications of the techniques outlined above to existing and future robotic systems.

4.1 Exploration of Mars

In 1996, NASA is scheduled to launch a spacecraft to Mars containing the Microrover Flight Experiment (MFEX) [14]. A major part of the mission consists of landing a semi-autonomous rover on the surface of Mars and using the rover to analyze soil and rock samples on the Martian surface. Due in part to funding cuts in the program, the size of the rover has been reduced from an initial mass of 800–1000 kg to a “mini” or “micro” rover in the 5–50 kg range.

The starting point for the MFEX rover design is the Rocky IV microrover pictured in Figure 7. Rocky IV has a mass of 7.2 kg and is approximately 60 cm in length. It uses a novel “rocker-bogie” design for the wheels which allow it to climb over obstacles which are as much as 50% larger than the wheel diameter. Rocky IV is equipped with a color video camera and numerous proximity sensors. The flight version of the rover will also be equipped with an alpha proton X-ray spectrometer (AXPS), to be used to analyze the composition of Martian soil and rocks.

The three primary goals of the MFEX rover mission are to complete a set of technology experiments in at least one soil type, complete an AXPS measurement on at least one rock with a video image of that rock, and take at least one full image of the lander. When these goals are met, the rover will complete additional experiments on different soil and rock types and attempt to take two more pictures of the lander, giving a complete view from all sides.

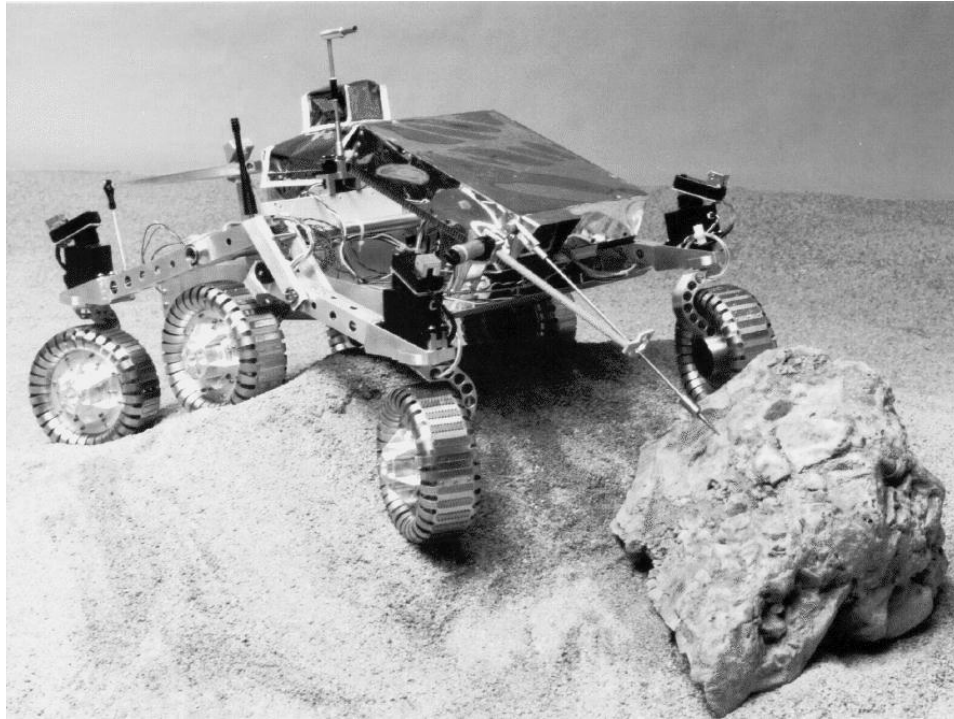


Figure 7: Rocky IV Microrover. (Figure courtesy the Jet Propulsion Laboratory, Pasadena, CA)

The basic mode of operation for the Mars rover mission involves humans providing overall planning and guidance while the rover itself will be responsible for low-level navigation and control. This level of autonomy in the control of the rover is necessitated by two factors: communication delays to Mars and communications bandwidth. The round trip travel time for a signal to Mars can range from approximately 10 minutes to as long as 40 minutes. This makes direct teleoperation of the rover impossible. Furthermore, the communications bandwidth of the lander is limited to approximately 4 megabits/day, or the equivalent of 10 bytes every 1.7 seconds, and can only be sustained while Earth is visible from the lander sight.

Because of these communications limitations, the Mars rover will be given commands once per day describing a sequence of actions to be carried out. On board computation will be used to provide low-level trajectory tracking and obstacle avoidance capabilities. Due to power constraints

on the rover and the need to use flight qualified hardware, the amount of computational power on board the rover is quite low. Current earth-based experiments employ an 8-bit microprocessor capable of performing only 1 million operations per second and containing less than 40K of memory. Behavioral control is being explored as a means of implementing the controller and has shown good reliability in autonomous navigation and manipulation tasks in both indoor and outdoor rough-terrain environments [6].

Path planning for the Mars rover involves finding paths which satisfy the basic kinematics of the rover and also avoid obstacles. To a rough degree of approximation, the constraints on the rover can be described by a set of Pfaffian constraints and hence the geometric machinery described above can be used to understand the vehicle motion and plan maneuvers. Examples of path planners for mobile robots in the presence of obstacles can be found in [7] and [8].

Future Mars missions are expected to include a sample and return scenario, in which a soil sample from Mars will be returned to the lander for further analysis. Since the lander can house much larger and heavier measurement apparatus, returning a soil sample to the lander allows much more detailed experiments to be run. In returning to the lander, the rover must accurately position itself to dock with the lander. This must be done in the presence of unknown terrain and with a reasonably high degree of precision.

One approach to this problem is the use of controllers which use real-time feedback to guide the rover to the docking bay. Stabilization of constrained systems of this type turns out to be a challenging theoretical problem which has received a large amount of attention in the controls community over the past several years (see Murray and M'Closkey [10] for a recent list of papers and experimental results in active stabilization). These results have relied heavily on the geometric point of view which has evolved over the past few years and will no doubt continue to make use of these tools.

4.2 Medical robotics: minimally invasive surgery

One of the exciting applications of robotic manipulation and locomotion is in medicine, particularly minimally invasive surgery. Over the past 10 years, the use of minimally invasive surgical techniques has increased dramatically in the US and abroad. These techniques offer several advantages over conventional surgery, including reduced hospital stays after an operation and decreased risk of infection and other complications.

A typical minimally invasive surgical operation is a laparoscopic chole-

cystectomy (gall bladder removal). In this procedure, a doctor removes the gall bladder through several small incisions in the patient's abdomen. One of the incisions is used to insert a tube which inflates the abdominal cavity with gas, while the other incisions are used to insert medical instruments. The gall bladder is removed by cutting it with a pair of scissors and extracting it through one of the incisions.

Gall bladder removal is one of the most common surgeries performed in the United States. Ten to fifteen years ago, the percentage of such surgeries which were done using minimally invasive techniques was minimal. However with new techniques and new medical instruments the use of minimally invasive techniques has surged, and this is by far the most common method currently in use for gall bladder removal.

Another minimally invasive surgical technique is the use of an endoscope for inspection and removal of tissue or polyps from the gastrointestinal tract. One example is an endoscopic polypectomy. In this procedure, a flexible endoscope is maneuvered near a polyp on the interior of the colon. Using a video display connected to a camera at the end of the endoscopic (via fiber optics), the surgeon is able to snare the polyp and cut it off with electrocautery while drawing the snare closed.

While the use of minimally invasive techniques has progressed rapidly, it is currently limited by several factors. Among them is the limited dexterity of the tools used by the surgeons and the large portions of the body which cannot be reached with an endoscope or laparoscope. One of the applications of the work described here is towards extending the abilities of surgeons in these directions.

Researchers at UC Berkeley, Harvard, and other institutions are working to develop a *teleoperative surgical workstation* which would allow surgeons more dexterity and ease of use than current minimally invasive technology [5]. They are focusing on a number of different problems, including the design, fabrication, and control of miniature robotic hands and the use of instrumented data gloves to allow the surgeon to control the hands in a natural way. A photograph of one of the hands which has been fabricated is shown in Figure 8.

The design and control of these complicated machines requires a thorough understanding of the basic mechanisms which are present in manipulation tasks. A fairly simple miniature hand such as the one shown in Figure 8 might have up to 9 degrees of freedom which, when combined with the dynamics of the object, can give a phase space with as many as 24 states (configurations plus velocities). The dynamics for such a system cannot be

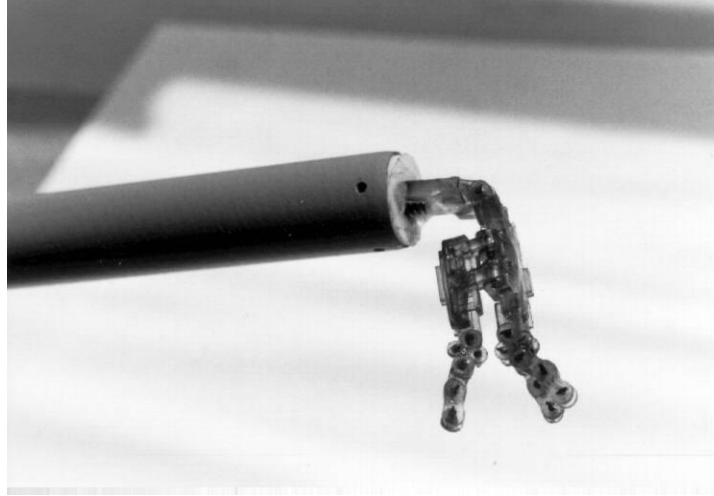


Figure 8: Three-fingered laparoscopic manipulator designed for tendon actuation (tendons omitted for clarity). The manipulator is shown in a 10 mm diameter laparoscopic trocar. (Figure courtesy of Micheal Cohn, UC Berkeley)

easily studied without understanding the basic structure which the dynamics inherit from the specific manipulation problem under consideration.

A second area of research in medical robotics is the development of small locomotion robots capable of navigation and inspection in the gastrointestinal tract. J. Burdick and his students at Caltech have built several prototype devices for locomotion inside of an intestine and clinical trials on pigs are in development [2]. The motion of their devices rely on the geometric phases associated with alternately inflating and uninflating a set of gas filled bags coupled with shortening and lengthening the robot along its longitudinal axis.

Other possibilities for gastrointestinal robots include the use of hyper-redundant (or “snake”) robots. Over the past 5 years, a very complete theory for these robots has been developed by Chirikjian and Burdick [3]. They have explored the use of hyper-redundant robots not only for locomotion tasks, but also for manipulation tasks in which the robot wraps itself around the object that it is manipulating. This provides a very stable grasp while still allowing the object to be manipulated within the grasp.

5 Conclusions and discussion

In this paper we have indicated some of the roles that geometric phases play in modern robotics, concentrating on applications in robotic locomotion and dextrous manipulation. As the robotic systems which we seek to control become increasingly complex, our ability to understand and program them is forced to rely more and more on the use of abstraction and advanced analysis. Further development of relevant theory, and applications of that theory to engineering problems, will help promote the use of advanced technology in many areas.

In addition to the specific applications discussed above, there are many other areas which overlap with the ideas presented here. For example, the use of geometric phases to understand biological motion is starting to become more clear. The use of central pattern generators (CPGs) to generate repetitive motion is common to many types of animals. One can view CPGs as the driving input to a set of kinematic constraints. Intuitively, the geometric phase associated with a particular gait pattern determines the direction and amount of motion of the system.

The explicit connection between CPGs and geometric phases remains to be established, but there are several clues which indicate that some new advances in theory might help. One such clue is the motion of the *snakeboard*, a commercial variant of a skateboard which is discussed by Marsden in his paper and is described in detail in [9]. The snakeboard relies on coupling between angular momentum and Pfaffian constraints to generate motion. Different gaits can be achieved in the snakeboard by using integrally-related periodic motions in the input variables of the system. As its name indicates, the snakeboard provides an important link between wheeled mobile robots and more complicated snake-like robots. By studying the geometry of the snakeboard we are able to understand one of the mechanisms by which locomotion occurs.

In order to expand this geometric point of view to other locomotion and manipulation problems, a slightly more general framework is required. In particular, while the notion of periodic motions for locomotion is fairly ubiquitous, the generation of trajectories via Pfaffian constraints is limited. For snake-like robots, a more general framework would allow different friction models and discontinuous dependence on the velocity of the snake. For legged locomotion, a completely different approach may be required since the contacts occur in piecewise fashion. These problems are the subject of current work by researchers in the US and around the world, and we can

expect to see exciting new insights and applications in the years to come.

Acknowledgements

The author would like to thank Joel Burdick, Michael Cohn, Carl Ruoff, Shankar Sastry, and Donna Shirley for providing references and pictures used in this report, and Roger Brockett, P. S. Krishnaprasad, and Jerry Marsden for many valuable and stimulating conversations on this material.

References

- [1] R. W. Brockett. Control theory and singular Riemannian geometry. In *New Directions in Applied Mathematics*, pages 11–27. Springer-Verlag, New York, 1981.
- [2] J. W. Burdick and B. Slatkin. Personal communication, 1994.
- [3] G. Chirikjian and J. W. Burdick. Kinematics of hyperredundant locomotion. *IEEE Transactions on Robotics and Automation*, 1994. (to appear).
- [4] W-L. Chow. Über systeme von linearen partiellen differentialgleichungen erster ordnung. *Math. Annalen*, 117:98–105, 1940.
- [5] M. Cohn, L. Crawford, J. Wendlandt, and S. Sastry. Millirobotics for telesurgery. Manuscript, 1994. (in preparation).
- [6] E. Gat, R. Desai, R. Ivlev, J. Lock, and D. Miller. Behavior control for robotic exploration of planetary surfaces. *IEEE Transactions on Robotics and Automation*, 1994. (to appear).
- [7] J-C. Latombe. *Robot Motion Planning*. Kluwer Academic Publishers, Boston, 1991.
- [8] J-P. Laumond, P. Jacobs, M. Taix, and R. M. Murray. A motion planner for nonholonomic mobile robots. *IEEE Transactions on Robotics and Automation*, 1994. (to appear).
- [9] A. Lewis, J. Ostrowski, R. M. Murray, and J. W. Burdick. Nonholonomic mechanics and locomotion: The snakeboard example. In *Proc. IEEE International Conference on Robotics and Automation*, 1994.

- [10] R. T. M'Closkey and R. M. Murray. Experiments in exponential stabilization of a mobile robot towing a trailer. In *Proc. American Control Conference*, 1994.
- [11] R. M. Murray, Z. Li, and S. S. Sastry. *A Mathematical Introduction to Robotic Manipulation*. CRC Press, 1994.
- [12] R. M. Murray and S. S. Sastry. Nonholonomic motion planning: Steering using sinusoids. *IEEE Transactions on Automatic Control*, 38(5):700–716, 1993.
- [13] E. Nelson. *Tensor Analysis*. Princeton University Press, Princeton, 1967.
- [14] D. Shirley Pivrotto. Finding the path to a better Mars rover. *Aerospace America*, pages 12–15, 25, September 1993.
- [15] J. A. Reeds and L. A. Shepp. Optimal paths for a car that goes both forwards and backwards. *Pacific Journal of Mathematics*, 145(2), 1990.
- [16] Tzila Shamir and Yosef Yomdin. Repeatability of redundant manipulators: Mathematical solution of the problem. *IEEE Transactions on Automatic Control*, 33(11):1004–1009, 1988.
- [17] A. Shapere and F. Wilczek. Efficiencies of self-propulsion at low Reynolds number. *Journal of Fluid Mechanics*, 198:587–599, 1989.
- [18] H. J. Sussmann and G. Tang. Shortest paths for the reeds-shepp car: A worked out example of the use of geometric techniques in nonlinear optimal control. Technical report, Rutgers Center for Systems and Control, September 1991.