Chapter 7

Differential Models

7.1 Motivation

In the models and methods studied so far, it has been assumed that a path can easily between obtained between any two configurations if there are no collisions. For example, the randomized roadmap approach assumed that two nearby configurations could be connected by a "straight line" in the configuration space. The constraints on the path are global in the sense that the restrictions are on the set of allowable configurations.

For the next few chapters, local constraints will be introduced. One of the simplest examples is a car-like robot. Imagine a trying to automate the motions of a typical automobile that has a limited steering angle. Consider the difficulty of moving a car sideways, while the rear wheels are always pointing forward. It would certainly make parallel parking easy if it was possible to simply turn all four wheels toward the curb. The orientation limits of the wheels, however, prohibit this motion. At any configuration, there are constraints on the velocity of the car. In other words, it is permitted only to move along certain directions to ensure that the wheels roll.

Although the motion is constrained in this way, most of us are experienced with making very complex driving maneuvers to parallel park a car. We would generally like to have algorithms that can maneuver a car-like robot and a variety of other nonholonomic systems while avoiding collisions. This will be the subject of nonholonomic planning.

7.2 Control System Representation

The most important concept in this section is the state transition equation. This equation indicates how the state will change over time, given a current state and current input. The input is selected by the user, and could correspond, for example, to the steering angle of a car. Suppose f is a vector-valued function, $f : X \times U \rightarrow R^n$, in which X is an n-dimensional state space, and U is an m dimensional *input space*. Let $\dot{x}$ denote a velocity vector,

$$\dot{x} = \left[\frac{dx_1}{dt} \;\; \frac{dx_2}{dt} \;\; \cdots \;\; \frac{dx_n}{dt} \right].$$

A *state transition equation* is defined as

$$\dot{x} = f(x, u). \tag{7.1}$$

For a given state, $x \in X$ and a given input $u \in U$, the state transition equation yields a velocity.

By integration, the state transition equation can be used to determine the state after some fixed amount of time, Δt has passed. For example, if we know the $x(t)$ and inputs $u(t')$ over the interval $t' \in [t, t + \Delta t]$, then the state, $x(t + \Delta t)$ can be determined as

$$x(t + \Delta t) = x(t) + \int_t^{t+\Delta t} f(x(t'), u(t'))dt'$$

The integral above cannot be evaluated directly because $x(t')$ appears in the integrand, but is known for time $t' > t$.

Several numerical techniques exist for numerically approximating the solution. Using the fact that

$$f(x,u) = \dot{x} = \frac{dx}{dt} \approx \frac{\Delta x}{\Delta t} = \frac{x(t+\Delta t) - x(t)}{\Delta t},$$

one can solve for $x(t+\Delta t)$ to yield the classic Euler integration method,

$$x(t+\Delta t) \approx x(t) + \Delta t \; f(x(t), u(t)).$$

For many applications, too much numerical error introduced by Euler integration. Runge-Kutta integration provides an improvement that is based on higher-order Taylor series expansion of the solution. One useful form of Runga-Kutta integration is the fourth-order approximation,

$$x(t+\Delta t) \approx x(t) + \frac{\Delta t}{6}(w_1 + 2w_2 + 2w_3 + w_4),$$

in which

$$w_1 = f(x(t), u(t)),$$

$$w_2 = f(x(t) + \frac{\Delta t}{2} \; w_1, u(t)),$$

$$w_3 = f(x(t) + \frac{\Delta t}{2} \; w_2, u(t)),$$

and

$$w_4 = f(x(t) + \Delta t \; w_3, u(t)).$$

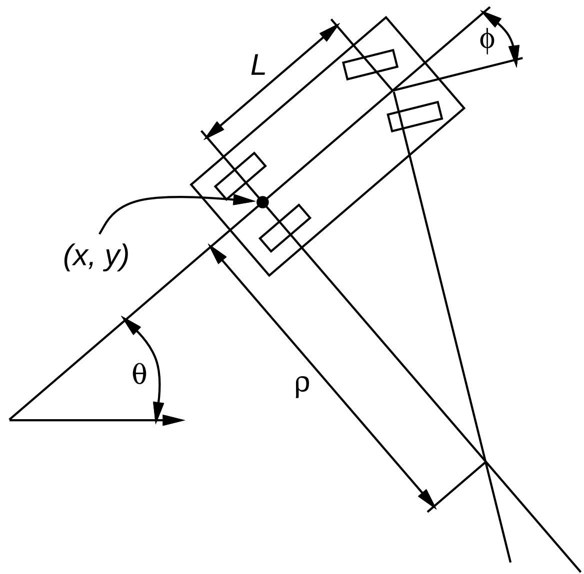
For some problems, a state transition equation might not be available; however, it is still possible to compute any future state, given a current state and an input. This might occur, for example, in a complex software system that simulates the dynamics of a automobile, or a collection of parts that bounce around on a table. In this situation, we simply define the existence of an *incremental simulator*, which serves as a "black box" that produces a future state, given any current state and input. Euler and Runge-Kutta integration may be viewed as techniques that convert a state transition equation into an incremental simulator.

7.3 Kinematics for Wheeled Systems

Several interesting state transition equations can be defined to model the motions of objects that move by rolling wheels. For all of these examples, the state space, X, is equivalent to the configuration space, $\mathcal{C}$.

7.3.1 A Simple Car

A simple example is the car-like robot. It is assumed that the car can translate and rotate, resulting in $\mathcal{C} = \mathbb{R}^2 \times S^1$. Assume that the state space is defined as $X = \mathcal{C}$. For convenience, let each state be denoted by (x, y, θ). Let s and ϕ denote two scalar inputs, which represent the speed of the car and the steering angle, respectively. The picture below indicates several parameters associated with the car.



The distance between the front and rear axles is represented as L. The steering angle is denoted by ϕ. The configuration is given by (x, y, θ). When the steering angle is ϕ, the car will roll in a circular motion, in which the radius of the circle is ρ. Note that ρ can be determined from the intersection of the two axes as shown (the angle between these axes is ϕ).

The task is to represent the motion of the car as a set of equations of the form

$$\dot{x} = f_1(x, y, \theta, s, \phi)$$

$$\dot{y} = f_2(x, y, \theta, s, \phi)$$

$$\dot{\theta} = f_3(x, y, \theta, s, \phi).$$

In a small time interval, the car must move in the direction that the rear wheels are pointing. This implies that $\frac{dy}{dx} = \tan\theta$. Since $\frac{dy}{dx} = \frac{\dot{y}}{\dot{x}}$ and $\tan\theta = \frac{\sin\theta}{\cos\theta}$, this motion constraint can be written as

$$-\dot{x}\sin\theta + \dot{y}\cos\theta = 0. \tag{7.2}$$

The equation above is satisfied if $\dot{x} = \cos\theta$ and $\dot{y} = \sin\theta$. Furthermore, any scalar multiple of this solution is also a solution, which corresponds directly to the speed of the car. Thus, the first two scalar components of the state transition equation are $\dot{x} = s\cos\theta$ and $\dot{y} = s\sin\theta$.

The next task is to derive the equation for $\dot{\theta}$. Let p denote the distance traveled by the car. Then $\dot{p} = s$, which is the speed. As shown in the figure above, ρ represents the radius of a circle that will be traversed by the center of the rear axle, when the steering angle is fixed. Note that $dp = \rho d\theta$. From simple trigonometry, $\rho = \frac{L}{\tan\phi}$, which implies

$$d\theta = \frac{\tan\phi}{L} dp.$$

Dividing by dt and using the fact that $\dot{p} = s$ yields

$$\dot{\theta} = \frac{s}{L} \tan\phi.$$

Thus, the state transition equation for the car-like robot is

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} s\cos\theta \\ s\sin\theta \\ \frac{s}{L}\tan\phi \end{pmatrix}$$

Most vehicles with steering have a limited steering angle, ϕ_{max} such that $0 < \phi_{max} < \frac{\pi}{2}$.

The speed of the car is usually bounded. If there are only two possible speeds (forward or reverse), $s \in \{-1, 1\}$, then the model is referred to as the *Reeds-Shepp* car. If the only possible speed is $s = 1$, then the model is referred to as the *Dubins* car.

7.3.2 A Continuous-Steering Car

In the previous model, the steering angle, ϕ, was an input, which implies that one can instantaneously move the front wheels. In many applications, this assumption is unrealistic. In the path traced out in the plane by the center of the rear axle of the car, there is a curvature discontinuity will occur when the steering angle is changed discontinuously. To make a car model that only generates smooth paths, the steering angle can be added as a state variable. The input is the angular velocity, ω, of the steering angle.

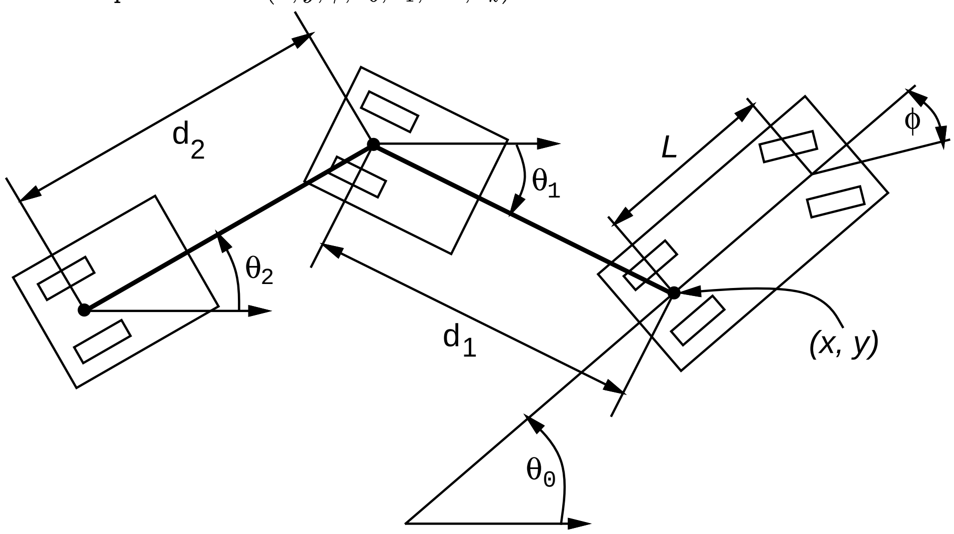
The result is a four-dimensional state space, in which each state is represented as (x, y, ϕ, θ). This yields the following state transition equation:

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} s\cos\theta \\ s\sin\theta \\ \omega \\ \frac{s}{L}\tan\phi, \end{pmatrix}$$

in which there are two inputs, s and ω.

7.3.3 A Car Pulling Trailers

The continuous-steering car can be extended to allow one or more single-axle trailers to be pulled. For k trailers, the state is represented as $(x, y, \phi, \theta_0, \theta_1, \ldots, \theta_k)$.



The state transition equation is

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \\ \dot{\theta}_0 \\ \vdots \\ \dot{\theta}_i \\ \vdots \end{pmatrix} = \begin{pmatrix} s\cos\theta \\ s\sin\theta \\ \omega \\ \frac{s}{L}\tan\phi \\ \vdots \\ \frac{s}{d_i}\left(\prod_{j=1}^{i-1}\cos(\theta_{j-1}-\theta_j)\right)\sin(\theta_{i-1}-\theta_i) \\ \vdots \end{pmatrix},$$

in which θ_0 is the orientation of the car, θ_i is the orientation of the i^{th} trailer, and d_i is the distance from the i^{th} trailer wheel axle to the hitch point.

7.3.4 A Differential Drive

The differential drive consists of a single axle, which connects two independently-controlled wheels. Each wheel is driven by its own motor, and it free to rotate without affecting the other wheel. Each state is represented as (x, y, θ). The state transition equation is

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} \frac{r}{2}(u_l + u_r)\cos\theta \\ \frac{r}{2}(u_l + u_r)\sin\theta \\ \frac{r}{\ell}(u_r - u_l) \end{pmatrix}, \tag{7.3}$$

in which r is the wheel radius, ℓ is the axle length, u_r is the angular velocity of the right wheel, and u_l is the angular velocity of the left wheel. If $u_l = u_r = 1$, the differential drive rolls forward. If $u_l = u_r = -1$, the differential drive rolls in the opposite direction. If $u_l = -u_r$, the differential drive performs a rotation.

7.4 Rigid-Body Dynamics

Full rigid-body dynamics were not covered. However, an example of a point-mass spacecraft was addressed.

For problems that involve dynamics, constraints will exist on accelerations, in addition to velocities and configurations. Accelerations may appear problematic because they represent second-order derivatives, which cannot appear in the state transition equation (7.1). To overcome this problem a state space will be defined that allows the equations of motion to be converted into the form $\dot{x} = f(x, u)$. Usually, the dimension of this state space is twice the dimension of the configuration space.

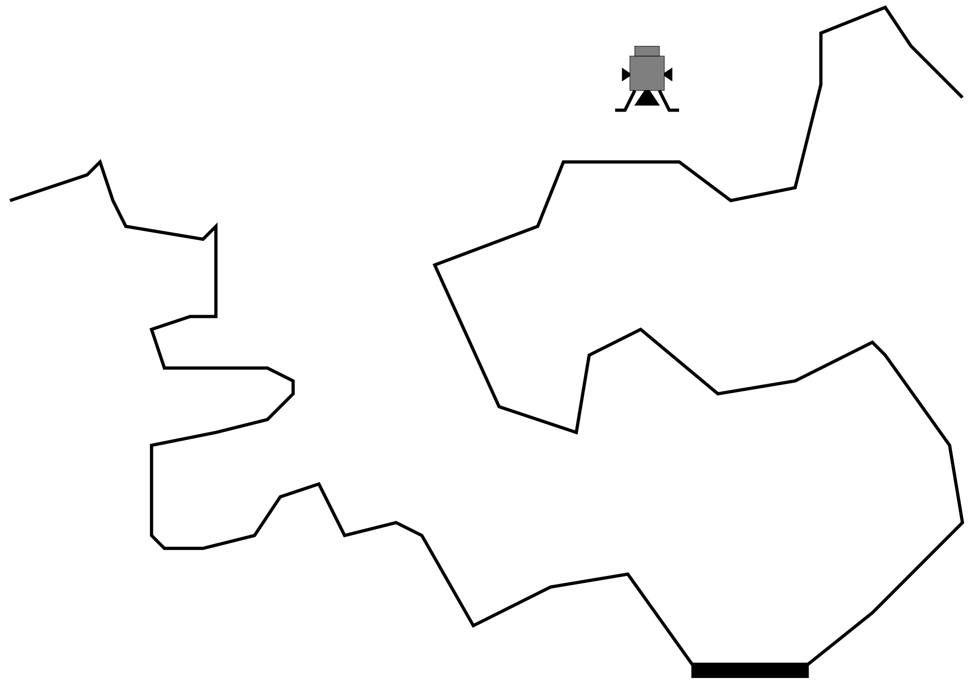
The state space For a broad class of problems, equations of motion that involve dynamics can be expressed as $\ddot{q} = g(\dot{q}, q)$, for some measurable function g. Suppose a problem is defined on an n-dimensional configuration space, $\mathcal{C}$. Define a $2n$-dimensional state vector $x = [q \;\; \dot{q}]$. In other words, x represents both configuration and velocity,

$$x = [q_1 \;\; q_2 \;\; \cdots q_n \;\; \dot{q}_1 \;\; \dot{q}_2 \;\; \cdots \;\; \dot{q}_n].$$

Let X denote the $2n$-dimensional state space, which is the set of all state vectors.

The goal is to construct a state transition equation of the form $\dot{x} = f(x, u)$. Given the definition of the state vector, note that $\dot{x}_i = x_{n+i}$ if $i \leq n$. This immediately defines half of the components of the state transition equation. The other half is defined using $\ddot{q} = g(\dot{q}, q)$. This is obtained by simply substituting each of the $\ddot{q}$, $\ddot{q}$, and q variables by their state space equivalents.

Example: Lunar lander A simple example that illustrates the concepts is given.



The lander is modeled as a point with mass, m, in a 2D world. It is not allowed to rotate, implying that $\mathcal{C} = \mathbb{R}^2$. There are three thrusters on the lander: Thruster One (right side), Thruster Two (bottom), and Thruster Three (left side). The activation of each thruster is considered as a binary switch. Let u_i denote a binary-valued action that can activate the i^{th} thruster. If $u_i = 1$, the thruster fires, if $u_i = 0$, then the thruster is dormant. Each of the two lateral thrusters provides a force F_s when activated. The upward thruster, mounted to the bottom of the lander, provides a force F_u when activated. Let g denote the acceleration of gravity.

From simple Newtonian mechanics, $\sum F = ma$, in which $\sum F$ denotes the vector sum of the forces, m denotes the mass of the lander, and a denote the acceleration, $\ddot{q}$. The q_1-component (x-direction) yields

$$m\ddot{q}_1 = u_1 F_s - u_3 F_s,$$

and the q_2-component (y-direction) yields

$$m\ddot{q}_2 = u_2 F_u - mg$$

The constraints above can be written in the form $f(q, \dot{q}, \ddot{q}) = 0$ (actually, the equations are simple enough to obtain $f(\ddot{q}) = 0$).

The lunar lander model can be transformed into a four-dimensional state space in which $x = [q_1 \;\; q_2 \;\; \dot{q}_1 \;\; \dot{q}_2]$. By replacing $\ddot{q}_1$ and $\ddot{q}_2$ with $\dot{x}_3$ and $\dot{x}_4$, respectively, the Newtonian equations of motion can be written as

$$\dot{x}_3 = \frac{F_s}{m}(u_1 - u_3)$$

$$\dot{x}_4 = \frac{u_2 F_u}{m} - g$$

Since $\dot{x}_1 = x_3$ and $\dot{x}_2 = x_4$, the state transition equation becomes

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{pmatrix} = \begin{pmatrix} x_3 \\ x_4 \\ \frac{F_s}{m}(u_1 - u_3) \\ \frac{u_2 F_u}{m} - g \end{pmatrix},$$

which is in the desired form $\dot{x} = f(x, u)$.

7.5 Multiple-Body Dynamics

Not covered.

7.6 More Examples

The nonholonomic integrator Here is a simple nonholonomic system that might be useful for experimentation. Let $X = \mathbb{R}^3$, and let the set of inputs, $U = \mathbb{R}^2$. The state transition equation for the nonholonomic integrator is

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{pmatrix} = \begin{pmatrix} u_1 \\ u_2 \\ x_1 u_2 - x_2 u_1 \end{pmatrix}.$$

Bibliography