Optimization-Based Control

Richard M. Murray
Control and Dynamical Systems
California Institute of Technology

DRAFT v1.2b, 15 January 2008

Preface

These notes serve as a supplement to *Feedback Systems* by Åström and Murray and expand on some of the topics introduced there. Our focus is on the use of optimization-based methods for control, including optimal control theory, receding horizon control and Kalman filtering. Each chapter is intended to be a standalone reference for advanced topics that are introduced in *Feedback Systems*.

Contents

Chapter 1. Trajectory Generation and Tracking **1**

1.1 Two Degree of Freedom Design 1
1.2 Trajectory Tracking and Gain Scheduling 3
1.3 Trajectory Generation and Differential Flatness 7
1.4 Further Reading 13

Chapter 2. Optimal Control **17**

2.1 Review: Optimization 17
2.2 Optimal Control of Systems 20
2.3 Examples 23
2.4 Linear Quadratic Regulators 25
2.5 Choosing LQR weights 26
2.6 Further Reading 27

Chapter 3. Receding Horizon Control **33**

Bibliography **34**

Chapter 1

Trajectory Generation and Tracking

This set of notes expands on Section 7.5 of *Feedback Systems* by Åström and Murray (ÅM08), which introduces the use of feedforward compensation in control system design. We begin with a review of the two degree of freedom design approach and then focus on the problem of generating feasible trajectories for a (nonlinear) control system. We make use of the concept of differential flatness as a tool for generating feasible trajectories.

Prerequisites. Readers should be familiar with modeling of input/output control systems using differential equations, linearization of a system around an equilibrium point and state space control of linear systems, including reachability and eigenvalue assignment. Although this material supplements concepts introduced in Chapter 7 of ÅM08, no knowledge of observers is required.

1.1 Two Degree of Freedom Design

A large class of control problems consist of planning and following a trajectory in the presence of noise and uncertainty. Examples include autonomous vehicles manuevering in city streets, mobile robots performing tasks on factor floors (or other planets), manufacturing systems that regulate the flow of parts and materials through a plant or factory, and supply chain management systems that balance orders and inventories across an enterprise. All of these systems are highly nonlinear and demand accurate performance.

To control such systems, we make use of the notion of *two degree of freedom* controller design. This is a standard technique in linear control theory that separates a controller into a feedforward compensator and a feedback compensator. The feedforward compensator generates the nominal input required to track a given reference trajectory. The feedback compensator corrects for errors between the desired and actual trajectories. This is shown schematically in Figure 1.1.

In a nonlinear setting, two degree of freedom controller design decouples the trajectory generation and asymptotic tracking problems. Given a desired output trajectory, we first construct a state space trajectory x_d and a nominal input u_d that satisfy the equations of motion. The error system can then be written as a time-varying control system in terms of the error, $e = x - x_d$. Under the assumption that that tracking error remains

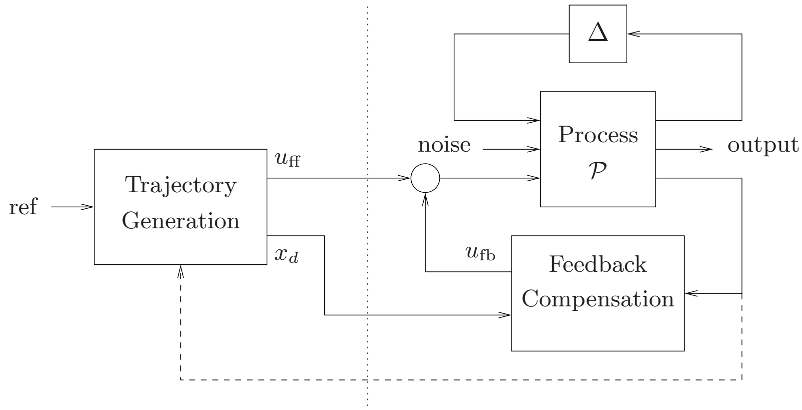


Figure 1.1: Two degree of freedom controller design for a process P with uncertainty Δ. The controller consists of a trajectory generator and feedback controller. The trajectory generation subsystem computes a feedforward command u_d along with the desired state x_d. The state feedback controller uses the measured (or estimated) state and desired state to compute a corrective input u_{fb}. Uncertainty is represented by the block Δ, representing unmodeled dynamics, as well as disturbances and noise.

small, we can linearize this time-varying system about $e = 0$ and stabilize the $e = 0$ state. (Note: in ÅM08 the notation u_{ff} was used for the desired (feedforward) input. We use u_d here to match the desired state x_d.)

More formally, we assume that our process dynamics can be described by a nonlinear differential equation of the form

$$\begin{aligned} \dot{x} &= f(x, u) \qquad && x \in \mathbb{R}^n, u \in \mathbb{R}^p, \\ y &= h(x, u) \qquad && y \in \mathbb{R}^q, \end{aligned} \tag{1.1}$$

where x is the system state, u is a vector of inputs and f is a smooth function describing the dynamics of the process. The smooth function h describes the output y that we wish to control. We are particularly interested in the class of control problems in which we wish to track a time-varying reference trajectory $r(t)$, called the *trajectory tracking* problem. In particular, we wish to find a control law $u = \alpha(x, r(\cdot))$ such that

$$\lim_{t\to\infty} \big(y(t) - r(t)\big) = 0.$$

We use the notation $r(\cdot)$ to indicate that the control law can depend not only on the reference signal $r(t)$ but also derivatives of the reference signal.

A *feasible trajectory* for the system (1.1) is a pair $(x_d(t), u_d(t))$ that satisfies the differential equation and generates the desired trajectory:

$$\dot{x}_d = f(x_d, u_d) \qquad r(t) = h(x_d, u_d).$$

The problem of finding a feasible trajectory for a system is called the *trajectory generation* problem, with x_d representing the desired state for the (nominal) system and u_d representing the desired input or the *feedforward* control. If we can find a feasible trajectory for the system, we can search for controllers of the form $u = \alpha(x, x_d, u_d)$ that track the desired reference trajectory.

In many applications, it is possible to attach a cost function to trajectories that describe how well they balance trajectory tracking with other factors, such as the magnitude of the inputs required. In such applications, it is natural to ask that we find the optimal controller. We can again use the two degree of freedom paradigm with an optimal control computation for generating the feasible trajectory. This subject is examined in more detail in Chapter 2. In addition, we can take the extra step of updating the generated trajectory based on the current state of the system. This additional feedback path is denoted by a dashed line in Figure 1.1 and allows the use of so-called *receding-horizon-control* techniques: a (optimal) feasible trajectory is computed from the current position to the desired position over a finite time T horizon, used for a short period of time $\delta < T$, and then recomputed based on the new position. Receding horizon control is described in more detail in Chapter 3.

A key advantage of optimization-based approaches is that they allow the potential for customization of the controller based on changes in *mission*, *condition* and *environment*. Because the controller is solving the optimization problem online, updates can be made to the cost function, to change the desired operation of the system; to the model, to reflect changes in parameter values or damage to sensors and actuators; and to the constraints, to reflect new regions of the state space that must be avoided due to external influences. Thus, many of the challenges of designing controllers that are robust to a large set of possible uncertainties become embedded in the online optimization.

1.2 Trajectory Tracking and Gain Scheduling

We begin by considering the problem of tracking a feasible trajectory. Assume that a trajectory generator is able to generate a trajectory (x_d, u_d) that satisfies the dynamics (1.1) and satisfies $r(t) = h(x_d(t), u_d(t))$. To design the controller, we construct the *error system*. Let $e = x - x_d$ and $v = u - u_d$ and compute the dynamics for the error:

$$\begin{aligned}\dot{e} &= \dot{x} - \dot{x}_d = f(x, u) - f(x_d, u_d) \\ &= f(e + x_d, v + u_d) - f(x_d) =: F(e, v, x_d(t), u_d(t)).\end{aligned}$$

In general, this system is time-varying.

For trajectory tracking, we can assume that e is small (if our controller is doing a good job), and so we can linearize around $e = 0$:

$$\frac{de}{dt} \approx A(t)e + B(t)v, \qquad A(t) = \frac{\partial F}{\partial e}\bigg|_{(x_d(t),u_d(t))}, \quad B(t) = \frac{\partial F}{\partial v}\bigg|_{(x_d(t),u_d(t)}.$$

It is often the case that $A(t)$ and $B(t)$ depend only on x_d, in which case it is convenient to write $A(t) = A(x_d)$ and $B(t) = B(x_d)$.

We start by reviewing the case where $A(t)$ and $B(t)$ are constant, in which case our error dynamics become

$$\dot{e} = Ae + Bv.$$

This occurs, for example, if the original nonlinear system is linear. We can then search for a control system of the form

$$v = -Ke + k_r r.$$

In the case where r is constant, we can apply the results of Chapter 6 of ÅM08 and solve the problem by finding a gain matrix K that gives the desired close loop dynamics (e.g., by eigenvalue assignment) and choosing k_r to give the desired output value at equilibrium. The equilibrium point is given by

$$x_e = -(A - BK)^{-1}Bk_r r \quad \Longrightarrow \quad y_e = -C(A - BK)^{-1}Bk_r r$$

and if we wish the output to be $y = r$ it follows that

$$k_r = -1/\big(C(A - BK)^{-1}B\big).$$

It can be shown that this formulation is equivalent to a two degree of freedom design where x_d and u_d are chosen to give the desired reference output (Exercise 1.1).

Returning to the full nonlinear system, assume now that x_d and u_d are either constant or slowly varying (with respect to the performance criterion). This allows us to consider just the (constant) linearized system given by $(A(x_d), B(x_d))$. If we design a state feedback controller $K(x_d)$ for each x_d, then we can regulate the system using the feedback

$$v = K(x_d)e.$$

Substituting back the definitions of e and v, our controller becomes

$$u = -K(x_d)(x - x_d) + u_d.$$

Note that the controller $u = \alpha(x, x_d, u_d)$ depends on (x_d, u_d), which themselves depend on the desired reference trajectory. This form of controller is called a *gain scheduled* linear controller with *feedforward* u_d.

More generally, the term gain scheduling is used to describe any controller that depends on a set of measured parameters in the system. So, for example, we might write

$$u = -K(x, \mu) \cdot (x - x_d) + u_d,$$

Figure 1.2: Gain scheduling. A general gain scheduling design involves finding a gain K at each desired operating point. This can be thought of as a gain surface, as shown on the left (for the case of a scalar gain). An approximation to this gain can be obtained by computing the gains at a fixed number of operating points and then interpolated between those gains. This gives an approximation of the continuous gain surface, as shown on the right.

where $K(x, \mu)$ depends on the *current* system state (or some portion of it) and an external parameter μ. The dependence on the current state x (as opposed to the desired state x_d) allows us to modify the closed loop dynamics differently depending on our location in the state space. This is particularly useful when the dynamics of the process vary depending on some subset of the states (such as the altitude for an aircraft or the internal temperature for a chemical reaction). The dependence on μ can be used to capture the dependence on the reference trajectory, or they can reflect changes in the environment or performance specifications that are not modeled in the state of the controller.

One limitation of gain scheduling as we have described it is that a separate set of gains must be designed for each operating condition x_d. In practice, gain scheduled controllers are often implemented by designing controllers at a fixed number of operating points and then interpolating the gains between these points, as illustrated in Figure 1.2. Suppose that we have a set of operating points $x_{d,j}$, $j = 1, \dots, N$. Then we can write our controller as

$$u = u_d - K(x)e \qquad K(x) = \sum(\alpha_j(x)K_j),$$

where K_j is a set of gains designed around the operating point $x_{d,j}$ and $\alpha_j(x)$ is a weighting factor. For example, we might choose the weights $\alpha_j(x)$ such that we take the gains corresponding to the nearest two operating points and weight them according to the Euclidean distance of the current state from that operating point; if the distance is small then we use a weight very near to 1 and if the distance is far then we use a weight very near to 0.

While the intuition behind gain scheduled controllers is fairly clear, some caution in required in using them. In particular, a gain scheduled controller is not gauranteed to be stable even if $K(x, \mu)$ locally stabilizes the system

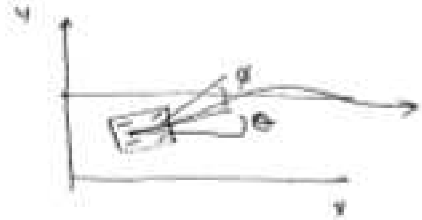


Figure 1.3: Vehicle steering using gain scheduling.

around a given equilibrium point. Gain scheduling can be proven to work in the case when the gain varies sufficiently slow (Exercise 1.3).

Example 1.1 Steering control with velocity scheduling
Consider the problem of controlling the motion of a automobile so that it follows a given trajectory on the ground, as shown in Figure 1.3. We use the model derived in ÅM08, choosing the reference point to be the center of the rear wheels. This gives dynamics of the form

$$\begin{aligned} \dot{x} &= \cos\theta\, v \\ \dot{y} &= \sin\theta\, v \\ \dot{\theta} &= \frac{v}{l}\tan\phi, \end{aligned} \tag{1.2}$$

where (x, y, θ) is the position and orientation of the vehicle, v is the velocity and ϕ is the steering angle, both considered to be inputs, and l is the wheelbase.

A simple feasible trajectory for the system is to follow a straight line in the x direction at lateral position y_r and fixed velocity v_r. This corresponds to a desired state $x_d = (v_r t, y_r, 0)$ and nominal input $u_d = (v_r, 0)$. Note that (x_d, u_d) is not an equilibrium point for the system, but it does satisfy the equations of motion.

Linearizing the system about the desired trajectory, we obtain

$$A_d = \left.\frac{\partial f}{\partial x}\right|_{(x_d,u_d)} = \left.\begin{bmatrix} 0 & 0 & -\sin\theta \\ 0 & 0 & \cos\theta \\ 0 & 0 & 0 \end{bmatrix}\right|_{(x_d,u_d)} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix},$$

$$B_d = \left.\frac{\partial f}{\partial u}\right|_{(x_d,u_d)} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & v_r/l \end{bmatrix}.$$

We form the error dynamics by setting $e = x - x_d$ and $w = u - u_d$:

$$\dot{e}_x = w_1, \qquad \dot{e}_y = e_\theta, \qquad \dot{e}_\theta = \frac{v_r}{l} w_2.$$

We see that the first state is decoupled from the second two states and hence we can design a controller by treating these two subsystems separately.

Suppose that we wish to place the closed loop eigenvalues of the longitudinal dynamics (e_x) at λ_1 and place the closed loop eigenvalues of the lateral dynamics (e_y, e_θ) at the roots of the polynomial equation $s^2 + a_1 s + a_2 = 0$. This can accomplished by setting

$$w_1 = -\lambda_1 e_x$$
$$w_2 = \frac{l}{v_r}(a_1 e_y + a_2 e_\theta).$$

Note that the gains depend on the velocity v_r (or equivalently on the nominal input u_d), giving us a gain scheduled controller.

In the original inputs and state coordinates, the controller has the form

$$\begin{bmatrix} v \\ \phi \end{bmatrix} = -\underbrace{\begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \dfrac{a_1 l}{v_r} & \dfrac{a_2 l}{v_r} \end{bmatrix}}_{K_d} \underbrace{\begin{bmatrix} x - v_r t \\ y - y_r \\ \theta \end{bmatrix}}_{e} + \underbrace{\begin{bmatrix} v_r \\ 0 \end{bmatrix}}_{u_d}.$$

The form of the controller shows that at low speeds the gains in the steering angle will be high, meaning that we must turn the wheel harder to achieve the same effect. As the speed increases, the gains become smaller. This matches the usual experience that at high speed a very small amount of actuation is required to control the lateral position of a car. Note that the gains go to infinity when the vehicle is stopped ($v_r = 0$), corresponding to the fact that the system is not reachable at this point. ∇

1.3 Trajectory Generation and Differential Flatness

We now return to the problem of generating a trajectory for a nonlinear system. Consider first the case of finding a trajectory $x_d(t)$ that steers the system from an initial condition x_0 to a final condition x_f. We seek a feasible solution $(x_d(t), u_d(t)$ that satisfies the dynamics of the process:

$$\dot{x}_d = f(x_d, u_d), \qquad x_d(0) = x_0,\, x_d(T) = x_f. \tag{1.3}$$

In addition, we may wish to satisfy additional constraints on the dynamics:

- Input saturation: $|u(t)| < M$;
- State constraints: $g(x) \leq 0$
- Tracking: $h(x) = r(t)$
- Optimization:

$$\min \int_0^T L(x,u)dt + V(x(T), u(T))$$

Formally, this problem corresponds to a two-point boundary value problem and can be quite difficult to solve in general.

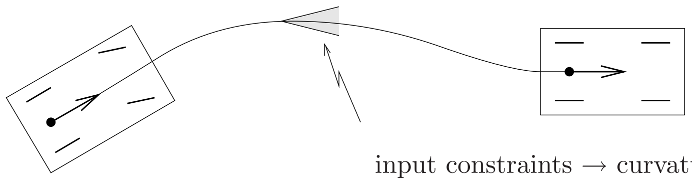


Figure 1.4: Simple model for an automobile. We wish to find a trajectory from an initial state to a final state that satisfies the dynamics of the system and constraints on the curvature (imposed by the limited travel of the front wheels).

As an example of the type of problem we would like to study, consider the problem of steering a car from an initial condition to a final condition, as show in Figure 1.4. To solve this problem, we must find a solution to the differential equations (1.2) that satisfies the endpoint conditions. Given the nonlinear nature of the dynamics, it seems unlikely that one could find explicit solutions that satisfy the dynamics except in very special cases (such as driving in a straight line).

A closer inspection of this system shows that it is possible to understand the trajectories of the system by exploiting the particular structure of the dynamics. Suppose that we are given a trajectory for the rear wheels of the system, $x(t)$ and $y(t)$. From equation (1.2), we see that we can use this solution to solve for the angle of the car by writing

$$\frac{\dot y}{\dot x} = \frac{\sin\theta}{\cos\theta} \qquad \Longrightarrow \qquad \theta = \tan^{-1}(\dot y/\dot x).$$

Furthermore, given θ we can solve for the velocity using the equation

$$\dot x = v\cos\theta \qquad \Longrightarrow \qquad v = \dot x/\cos\theta,$$

assuming $\cos\theta \neq 0$ (if it is, use $v = \dot y/\sin\theta$). And given θ, we can solve for ϕ using the relationship

$$\dot\theta = \frac{v}{l}\tan\phi \qquad \Longrightarrow \qquad \phi = \tan^{-1}(\frac{l\dot\theta}{v}).$$

Hence all of the state variables and the inputs can be determined by the trajectory of the rear wheels and its derivatives. This property of a system is known as *differential-flatness*.

Definition 1.1 (Differential flateness)**.** A nonlinear system (1.1) is *differentially flat* if there exists a function α such that

$$z = \alpha(x, u, \dot u \ldots, u^{(p)})$$

and we can write the solutions of the nonlinear system as functions of z and an finite number of derivatives

$$x = \beta(z, \dot z, \ldots, z^{(q)}),$$
$$u = \gamma(z, \dot z, \ldots, z^{(q)}).$$

For a differentially flat system, all of the feasible trajectories for the system can be written as functions of a flat output $z(\cdot)$ and its derivatives. The number of flat outputs is always equal to the number of system inputs. The kinematic car is differentially flat with the position of the rear wheels as the flat output. Differentially flat systems were originally studied by Fliess et al. [FLMR92].

Differentially flat systems are useful in situations where explicit trajectory generation is required. Since the behavior of flat system is determined by the flat outputs, we can plan trajectories in output space, and then map these to appropriate inputs. Suppose we wish to generate a feasible trajectory for the the nonlinear system

$$\dot{x} = f(x, u), \qquad x(0) = x_0,\, x(T) = x_f.$$

If the system is differentially flat then

$$\begin{aligned} x(0) &= \beta\big(z(0), \dot{z}(0), \dots, z^{(q)}(0)\big) = x_0 \\ x(T) &= \gamma\big(z(T), \dot{z}(T), \dots, z^{(q)}(T)\big) = x_f \end{aligned} \tag{1.4}$$

and any trajectory for z that satisfies these boundary conditions will be a feasible trajectory for the system.

In particular, given initial and final conditions on z and its derivatives that satisfy equation (1.4), any curve $z(\cdot)$ satisfing those conditions will correspond to a feasible trajectory of the system. We can parameterize the flat output trajectory using a set of smooth basis functions $\psi^i(t)$:

$$z(t) = \sum_{i=1}^{N} \alpha_i \psi_i(t), \qquad \alpha_i \in \mathbb{R}.$$

We seek a set of coefficients α_i, $i = 1, \dots, N$ such that $z(t)$ satisfies the boundary conditions (1.4). The derivatives of the flat output can be computed in terms of the derivatives of the basis functions:

$$\begin{aligned} \dot{z}(t) &= \sum_{i=1}^{N} \alpha_i \dot{\psi}_i(t) \\ &\vdots \\ \dot{z}^{(q)}(t) &= \sum_{i=1}^{N} \alpha_i \psi_i^{(q)}(t). \end{aligned}$$

We can thus write the conditions on the flat outputs and their derivatives as

$$\begin{bmatrix} \psi_1(0) & \psi_2(0) & \dots & \psi_N(0) \\ \dot\psi_1(0) & \dot\psi_2(0) & \dots & \dot\psi_N(0) \\ \vdots & \vdots & & \vdots \\ \psi_1^{(q)}(0) & \psi_2^{(q)}(0) & \dots & \psi_N^{(q)}(0) \\ \psi_1(T) & \psi_2(T) & \dots & \psi_N(T) \\ \dot\psi_1(T) & \dot\psi_2(T) & \dots & \dot\psi_N(T) \\ \vdots & \vdots & & \vdots \\ \psi_1^{(q)}(T) & \psi_2^{(q)}(T) & \dots & \psi_N^{(q)}(T) \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_N \end{bmatrix} = \begin{bmatrix} z(0) \\ \dot z(0) \\ \vdots \\ z^{(q)}(0) \\ z(T) \\ \dot z(T) \\ \vdots \\ z^{(q)}(T) \end{bmatrix}$$

This equation is a *linear* equation of the form $M\alpha = \bar z$. Assuming that M has a sufficient number of columns and that it is full column rank, we can solve for a (possibly non-unique) α that solves the trajectory generation problem.

Example 1.2 Nonholonomic integrator
A simple nonlinear system called a *nonholonomic integrator* [**?**] is given by the differential equations

$$\begin{aligned} \dot x_1 &= u_1 \\ \dot x_2 &= u_2 \\ \dot x_3 &= x_2 u_1 \end{aligned}$$

This system is differentially flat with flat output $z = (x_1, x_3)$. The relationship betwen the flat variables and the states is given by

$$\begin{aligned} x_1 &= z_1 \\ x_2 &= \dot x_3/\dot x_1 = \dot z_2/\dot z_1 \\ x_3 &= z_2. \end{aligned}$$

Using simple polynomials as our basis functions,

$$\begin{aligned} \psi_{1,1}(t) = 1 \quad \psi_{1,2}(t) = t\psi_{1,3}(t) &= t^2\psi_{1,4}(t) &= t^3 \\ \psi_{2,1}(t) = 1 \quad \psi_{2,2}(t) = t\psi_{2,3}(t) &= t^2\psi_{2,4}(t) &= t^3, \end{aligned}$$

the equations for the feasible (flat) trajectory become

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & T & T^2 & T^3 & 0 & 0 & 0 & 0 \\ 0 & 1 & 2T & 3T^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & T & T^2 & T^3 \\ 0 & 0 & 0 & 0 & 0 & 1 & 2T & 3T^2 \end{bmatrix} \begin{bmatrix} \alpha_{11} \\ \alpha_{12} \\ \alpha_{13} \\ \alpha_{14} \\ \alpha_{21} \\ \alpha_{22} \\ \alpha_{23} \\ \alpha_{24} \end{bmatrix} = \begin{bmatrix} x_{1,0} \\ 1 \\ x_{3,0} \\ x_{2,0} \\ x_{1,f} \\ 1 \\ x_{3,f} \\ x_{2,f} \end{bmatrix}.$$

This is a set of 8 linear equations in 8 variables. It can be shown that the matrix M is full rank and the system can be solved numerically. ∇

Note that no ODEs need to be integrated in order to compute the feasible trajectories for a differentially flat system (unlike optimal control methods that we will consider in the next chapter, which involve parameterizing the *input* and then solving the ODEs). This is the defining feature of differentially flat systems. The practical implication is that nominal trajectories and inputs which satisfy the equations of motion for a differentially flat system can be computed in a computationally efficient way (solution of algebraic equations). Since the flat output functions are completely free, the only constraints that must be satisfied are the initial and final conditions on the endpoints, their tangents, and higher order derivatives. Any other constraints on the system, such as bounds on the inputs, can be transformed into the flat output space and (typically) become limits on the curvature or higher order derivative properties of the curve.

If there is a performance index for the system, this index can be transformed and becomes a functional depending on the flat outputs and their derivatives up to some order. By approximating the performance index we can achieve paths for the system that are suboptimal but still feasible. This approach is often much more appealing than the traditional method of approximating the system (for example by its linearization) and then using the exact performance index, which yields optimal paths but for the wrong system.

In light of the techniques that are available for differentially flat systems, the characterization of flat systems becomes particularly important. Unfortunately, general conditions for flatness are not known, but all (dynamic) feedback linearizable systems are differentially flat, as are all driftless systems that can be converted into chained form (see [vNRM94] for details). Another large class of differentially flat systems are those in "pure feedback form":

$$\begin{aligned}
\dot{x}_1 &= f_1(x_1, x_2) \\
\dot{x}_2 &= f_2(x_1, x_2, x_3) \\
&\vdots \\
\dot{x}_n &= f_n(x_1, \dots, x_n, u).
\end{aligned}$$

Under certain regularity conditions these systems are differentially flat with output $y = x_1$. These systems have been used for so-called "integrator backstepping" approaches to nonlinear control by Kokotovic et al. [KKM91]. Figure 1.5 shows some additional systems that are differentially flat.

Example 1.3 Planar ducted fan

Consider the dynamics of a planar, vectored thrust flight control system as shown in Figure 1.6. This system consists of a rigid body with body fixed forces and is a simplified model for the Caltech ducted fan [**?**]. Let (x, y, θ)

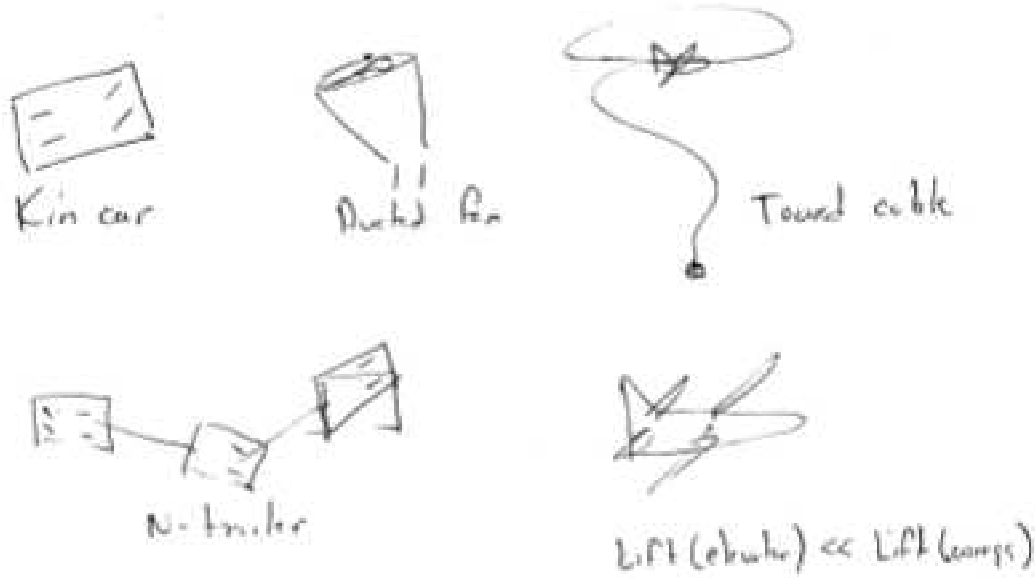


Figure 1.5: Examples of flat systems.

denote the position and orientation of the center of mass of the fan. We assume that the forces acting on the fan consist of a force f_1 perpendicular to the axis of the fan acting at a distance r from the center of mass, and a force f_2 parallel to the axis of the fan. Let m be the mass of the fan, J the moment of inertia, and g the gravitational constant. We ignore aerodynamic forces for the purpose of this example.

The dynamics for the system are

$$\begin{aligned} m\ddot{x} &= f_1 \cos\theta - f_2 \sin\theta, \\ m\ddot{y} &= f_1 \sin\theta + f_2 \cos\theta - mg, \\ J\ddot{\theta} &= r f_1. \end{aligned} \tag{1.5}$$

Martin et al. [MDP94] showed that this system is differentially flat and that one set of flat outputs is given by

$$\begin{aligned} z_1 &= x - (J/mr)\sin\theta, \\ z_2 &= y + (J/mr)\cos\theta. \end{aligned} \tag{1.6}$$

Using the system dynamics, it can be shown that

$$\ddot{z}_1 \cos\theta + (\ddot{z}_2 + g)\sin\theta = 0 \tag{1.7}$$

and thus given $z_1(t)$ and $z_2(t)$ we can find $\theta(t)$ except for an ambiguity of π and away from the singularity $\ddot{z}_1 = \ddot{z}_2 + g = 0$. The remaining states and the forces $f_1(t)$ and $f_2(t)$ can then be obtained from the dynamic equations, all in terms of z_1, z_2, and their higher order derivatives. ∇

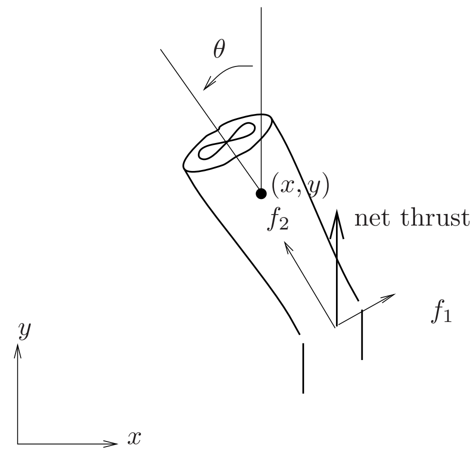


Figure 1.6: Planar ducted fan engine. Thrust is vectored by moving the flaps at the end of the duct.

1.4 Further Reading

The two degree of freedom controller structure introduced in this chapter is described in a bit more detail in ÅM08 [ÅM08] (in the context of output feedback control) and a description of some of the origins of this structure are provided in the "Further Reading" section of Chapter 8. Gain scheduling is a classical technique that is often omitted from introductory control texts, but a good desciption can be found in the survey article by Rugh [Rug90] and the work of Shamma [Sha90]. Differential flatness was originally developed by Fliess, Levin, Martin and Rouchon [FLMR92]. See [Mur97] for a description of the role of flatness in control of mechanical systems and [vNM98] for more information on flatness applied to flight control systems.

Exercises

1.1 (Feasible trajectory for constant reference) Consider a linear input/output system of the form

$$\dot{A}x + Bu, \qquad y = Cx \tag{1.8}$$

in which we wish to track a constant reference r. A feasible (steady state) trajectory for the system is given by solving the equation

$$\begin{bmatrix} 0 \\ r \end{bmatrix} = \begin{bmatrix} A & B \\ C & 0 \end{bmatrix} \begin{bmatrix} x_d \\ u_{\text{ff}} \end{bmatrix}$$

for x_d and u_{ff}.

(a) Show that these equations always has a solution as long as the linear system (1.8) is reachable.

(b) In Section 6.2 of ÅM08 we showed that the reference tracking problem could be solved using a control law of the form $u = -Kx + k_r r$. Show that this is equivalent to a two degree of freedom control design using x_d and u_{ff} and give a formula for k_r in terms of x_d and u_{ff}. Show that this formula matches that given in ÅM08.

1.2 A simplified model of the steering control problem is derived in Åström and Murray, Example 5.12. The model has the form

$$\dot{z} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} z + \begin{bmatrix} \gamma \\ 1 \end{bmatrix} u$$
$$y = z_1$$

where $z \in \mathbb{R}^2$ is the (normalized) state of the system and γ is a parameter related to the speed of the vehicle. Suppose that we wish to track a piecewise constant reference trajectory

$$r = \texttt{square}(2\pi t/20),$$

where `square` is the square wave function in MATLAB. Suppose further that the speed of the vehicle varies according to the formula

$$\gamma = 2 + 2\sin(2\pi t/50).$$

Design and implement a gain-scheduled controller for this system by first designing a state space controller that places the closed loop poles of the system at the roots of $s^2 + 2\zeta\omega_0 s + \omega_0^2$, where $\zeta = 0.7$ and $\omega_0 = 1$. You should design controllers for three different parameter values: $\gamma = 0, 2, 4$. Then use linear interpolation to compute the gain for values of γ between these fixed values. Compare the performance of the gain scheduled controller to a simple controller that assumes $\gamma = 2$ for the purpose of the control design (but leaving γ time-varying in your simulation).

Note: a MATLAB file with the vehicle dynamics is available on the course web page. You can use this if you like to get the reference trajectory and parameter variation.

1.3 (Stability of gain scheduled controllers for slowly varying systems) Consider a nonlinear control system with gain scheduled feedback

$$\dot{e} = f(e, v) \qquad v = k(\mu)e,$$

where $\mu(t) \in \mathbb{R}$ is an externally specified parameter (eg, the desired trajectory) and $k(\mu)$ is chosen such that the linearization of the closed loop system around the origin is stable for each fixed μ.

Show that if $|\dot{\mu}|$ is sufficiently small then the equilibrium point is locally asymptotically stable for the full nonlinear, time-varying system. (Hint: find a Lyapunov function of the form $V = x^T P(\mu)x$ based on the linearization of the system dynamics for fixed μ and then show this is a Lyapunov function for the full system.)

1.4 (Flatness of systems in reachable canonical form) Consider a single input system in reachable canonical form [ÅM08, Sec. 6.1]:

$$\frac{dx}{dt} = \begin{bmatrix} -a_1 & -a_2 & -a_3 & \dots & -a_n \\ 1 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \\ \vdots & & \ddots & \ddots & \vdots \\ 0 & & & 1 & 0 \end{bmatrix} x + \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} u, \tag{1.9}$$
$$y = \begin{bmatrix} b_1 & b_2 & b_3 & \dots & b_n \end{bmatrix} x + du.$$

Suppose that we wish to find an input u that moves the system from x_0 to x_f. This system is differentially flat with flat output given by $z = x_n$ and hence we can parameterize the solutions by a curve of the form

$$x_n(t) = \sum_{k=0}^{N} \alpha_k t^k, \tag{1.10}$$

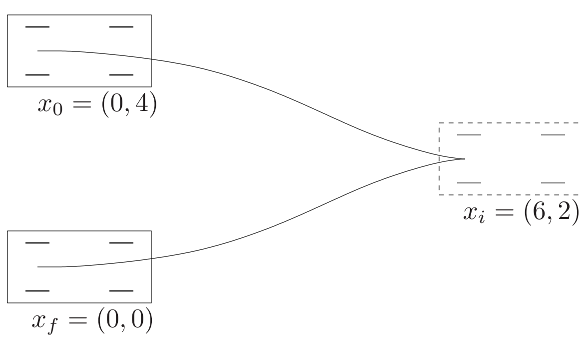
where N is a sufficiently large integer.

(a) Compute the state space trajectory $x(t)$ and input $u(t)$ corresponding to equation (1.10) and satisfying the differential equation (1.9). Your answer should be an equation similar to equation (1.10) for each state x_i and the input u.

(b) Find an explicit input that steers a double integrator system between any two equilibrium points $x_0 \in \mathbb{R}^2$ and $x_f \in \mathbb{R}^2$.

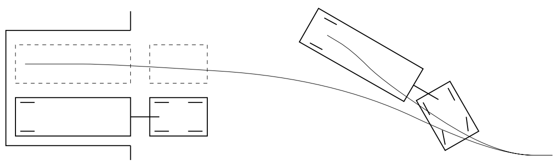
(c) Show that all reachable systems are differentially flat and give a formula for the flat output.

1.5 Consider the lateral control problem for an autonomous ground vehicle as described in Example 1.1 and Section 1.3. Using the fact that the system is differentially flat, find an explicit trajectory that solves the following parallel parking manuever:



Your solution should consist of two segments: a curve from x_0 to x_i with $v > 0$ and a curve from x_i to x_f with $v < 0$. For the trajectory that you determine, plot the trajectory in the plane (x versus y) and also the inputs v and ϕ as a function of time.

1.6 Consider first the problem of controlling a truck with trailer, as shown in the figure below:



$$\begin{aligned}
\dot{x} &= \cos\theta\, u_1 \\
\dot{y} &= \sin\theta\, u_1 \\
\dot{\phi} &= u_2 \\
\dot{\theta} &= \frac{1}{l}\tan\phi\, u_1 \\
\dot{\theta}_1 &= \frac{1}{d}\cos(\theta-\theta_1)\sin(\theta-\theta_1)u_1,
\end{aligned}$$

The dynamics are given above, where (x, y, θ) is the position and orientation of the truck, ϕ is the angle of the steering wheels, θ_1 is the angle of the trailer, and l and d are the length of the truck and trailer. We want to generate a trajectory for the truck to move it from a given initial position to the loading dock. We ignore the role of obstacles and concentrate on generation of feasible trajectories.

(a) Show that the system is differentially flat using the center of the rear wheels of the trailer as the flat output.

(b) Generate a trajectory for the system that steerings the vehicle from an initial condition with the truck and trailer perpendicular to the loading dock into the loading dock.

(c) Write a simulation of the system stabilizes the desired trajectory and demonstrate your two-degree of freedom control system maneuving from several different initial conditions into the parking space, with either disturbances or modeling errors included in the simulation.

Chapter 2
Optimal Control

This set of notes expands on Chapter 6 of *Feedback Systems* by Åström and Murray (ÅM08), which introduces the concepts of reachability and state feedback. We also expand on topics in Section 7.5 of ÅM08 in the area of feedforward compensation. Beginning with a review of optimization, we introduce the notion of Lagrange multipliers and provide a summary of the Pontryagin's maximum principle. Using these tools we derive the linear quadratic regulator for linear systems and describe its usage.

Prerequisites. Readers should be familiar with modeling of input/output control systems using differential equations, linearization of a system around an equilibrium point and state space control of linear systems, including reachability and eigenvalue assignment.

2.1 Review: Optimization

Consider first the problem of finding the maximum of a smooth function $F : \mathbb{R}^n \to \mathbb{R}$. That is, we wish to find a point $x^* \in \mathbb{R}^n$ such that $F(x^*) \geq F(x)$ for all $x \in \mathbb{R}^n$. A necessary condition for x^* to be a maximum is that the gradient of the function be zero at x^*,

$$\frac{\partial F}{\partial x}(x^*) = 0.$$

Figure 2.1 gives a graphical interpretation of this condition. Note that these are *not* sufficient conditions; the points x_1 and x_2 and x^* in the figure all

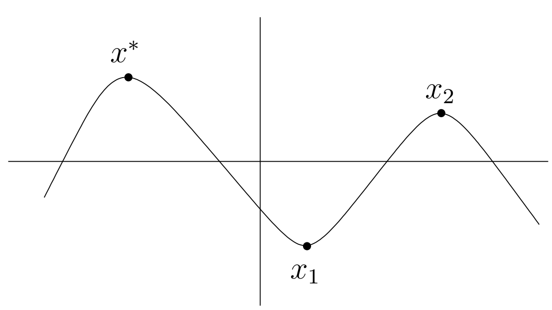
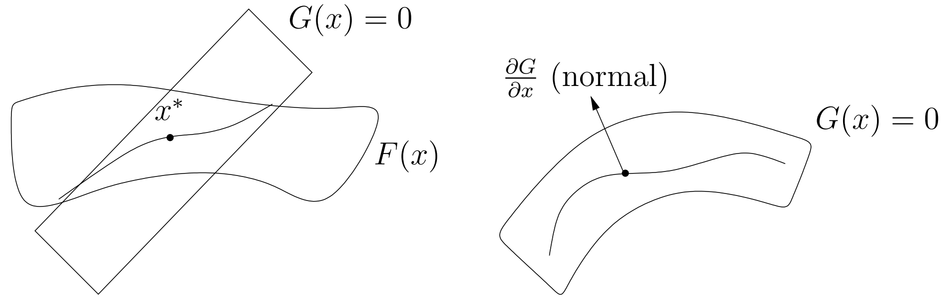


Figure 2.1: Optimization of functions. The maximum of a function occurs at a point where the gradient is zero.



(a) Constrained optimization
(b) Constraint normal vectors

Figure 2.2: Optimization with constraints. (a) We seek a point x^* that maximizes $F(x)$ while lying on the surface $G(x) = 0$. (b) We can parameterize the constrained directions by computing the gradient of the constraint G.

satisfy the necessary condition but only one is the (global) maximum.

The situation is more complicated if constraints are present. Let $G_i : \mathbb{R}^n \to \mathbb{R}$, $i = 1, \dots, k$ be a set of smooth functions with $G_i(x) = 0$ representing the constraints. Suppose that we wish to find $x^* \in \mathbb{R}^n$ such that $G_i(x^*) = 0$ and $F(x^*) \geq F(x)$ for all $x \in \{x \in \mathbb{R}^n : G_i(x) = 0, i = 1, \dots, k\}$. This situation can be visualized as constraining the point to a surface (defined by the constraints) and searching for the maximum of the cost function along this surface, as illustrated in Figure 2.2.

A necessary condition for being at a maximum is that there are no directions tangent to the constraints that also increase the cost. The normal directions to the surface are spanned by $\partial G_i / \partial x$, as shown in Figure **??**. A necessary condition is that the gradient of F is spanned by vectors that are normal to the constraints, so that the only directions that increase the cost violate the constraints. We thus require that there exist scalars λ_i, $i = 1, \dots, k$ such that

$$\frac{\partial F}{\partial x}(x^*) + \sum_{i=1}^{k} \lambda_i \frac{\partial G_i}{\partial x}(x^*) = 0.$$

If we let $G = \begin{bmatrix} G_1 & G_2 & \dots & G_k \end{bmatrix}^T$, then we can write this condition as

$$\frac{\partial F}{\partial x} + \lambda^T \frac{\partial G}{\partial x} = 0$$

the term $\frac{\partial F}{\partial x}$ is the usual (gradient) optimality condition while the term $\frac{\partial G}{\partial x}$ is used to "cancel" the gradient in the directions normal to the constraint.

An alternative condition can be derived by modifying the cost function to incorporate the constraints. Defining $\widetilde{F} = F + \sum \lambda_i G_i$, the necessary condition becomes

$$\frac{\partial \widetilde{F}}{\partial x}(x^*) = 0.$$

The scalars λ_i are called *Lagrange multipliers*. Minimize $\widetilde{F}$ is equivalent to the optimization given by

$$\min_x \left(F(x) + \lambda^T G(x)\right).$$

The variables λ can be regarded as free variables, which implies that need to choose x such that $G(x) = 0$. Otherwise, we could choose λ to generate a large cost.

Example 2.1 Two free variables with a constraint
Consider the cost function given by

$$F(x) = F_0 - (x_1 - a)^2 - (x_2 - b)^2,$$

which has an unconstrained maximum at $x = (a, b)$. Suppose that we add a constraint $G(x) = 0$ given by

$$G(x) = x_1 - x_2.$$

With this constrain, we seek to optimize F subject to $x_1 = x_2$. Although in this case we could easily do this by simple substitution, we instead carry out the more general procedure using Lagrange multipliers.

The augmented cost function is given by

$$\tilde{F}(x) = F_0 - (x_1 - a)^2 - (x_2 - b)^2 + \lambda(x_1 - x_2),$$

where λ is the Lagrange multiplier for the constraint. Taking the derivative of F, we have

$$\frac{\partial F}{\partial x} = \begin{bmatrix} -2x_1 + 2a + \lambda & -2x_2 + 2b - \lambda \end{bmatrix}.$$

Setting each of these equations equal to zero, we have that at the maximum

$$x_1^* = a + \lambda/2, \qquad x_2^* = b - \lambda/2.$$

The remaining equation that we need is the constraint, which requires that $x_1^* = x_2^*$. Using these three equations, we see that $\lambda^* = b - a$ and we have

$$x_1^* = \frac{a+b}{2}, \qquad x_2^* = \frac{a+b}{2}.$$

To verify the geometric view described above, note that the gradients of F and G are given by

$$\frac{\partial F}{\partial x} = \begin{bmatrix} -2x_1 + 2a & -2x_2 + 2b \end{bmatrix}, \qquad \frac{\partial G}{\partial x} = \begin{bmatrix} 1 & -1 \end{bmatrix}.$$

At the optimal value of the (constrained) optimization, we have

$$\frac{\partial F}{\partial x} = \begin{bmatrix} a - b & b - a \end{bmatrix}, \qquad \frac{\partial G}{\partial x} = \begin{bmatrix} 1 & -1 \end{bmatrix}.$$

Although the derivative of F is not zero, it is pointed in a direction that is normal to the constraint, and hence we cannot decrease the cost while staying on the constraint surface. ∇

We have focused on finding the maximum of a function. We can switch back and forth between max and min by simply negating the cost function:

$$\max_x F(x) = \min_x \bigl(-F(x)\bigr)$$

We see that the conditions that we have derived are independent of the sign of F since they only depend on the gradient begin zero in approximate directions. Thus finding x^* that satisfies the conditions corresponds to finding an *extremum* for the function.

Very good software is available for solving optimization problems numerically of this sort. The NPSOL and SNOPT libraries are available in FORTRAN (and C). In MATLAB, the fmin function can be used to solve a constrained optimization problem.

2.2 Optimal Control of Systems

Consider now the *optimal control problem*:

$$\min_u \underbrace{\int_0^T L(x,u)\,dt}_{\text{integrated cost}} + \underbrace{V\bigl(x(T),u(T)\bigr)}_{\text{final cost}}$$

subject to the constraint

$$\dot{x} = f(x,u) \qquad x \in \mathbb{R}^n,\ u \in \mathbb{R}^m.$$

Abstractly, this is a constrained optimization problem where we seek a *feasible trajectory* $(x(t), u(t))$ that minimizes the cost function

$$J(x,u) = \int_0^T L(x,u)\,dt + V\bigl(x(T),u(T)\bigr).$$

More formally, this problem is equivalent to the "standard" problem of minimizing a cost function $J(x,u)$ where $(x,u) \in L_2[0,T]$ (the set of square integral functions) and $h(z) = \dot{x}(t) - f(x(t),u(t)) = 0$ models the dynamics.

There are many variations and special cases of the optimal control problem. We mention a few here:

Infinite Horizon. if we let $T = \infty$ and set $V = 0$, then we seek to optimize a cost function over all time. This is called the *infinite horizon* optimal control problem, versus the *finite horizon* problem with $T < \infty$.

Linear Quadratic. If the dynamical system is linear and the cost function is quadratic, we obtain the *linear quadratic* optimal control problem:

$$\dot{x} = Ax + Bu \qquad J = \int_0^T \left(x^T Q x + u^T R u\right)\, dt + x^T(T) P_1 x(T).$$

In this formulation, $Q \geq 0$ penalizes state error (assumes $x_d = 0$), $R > 0$ penalizes the input (*must* be positive definite) and $P_1 > 0$ penalizes terminal state.

Terminal Constraints. It is often convenient to ask that the final value of the trajectory, denoted x_f, be specified. We can do this by requiring that $x(T) = x_f$ or by using a more general form of constraint:

$$\psi_i(x(T)) = 0, \qquad i = 1, \dots, q.$$

The fully constrained case is obtained by setting $q = n$ and defining $\psi_i(x(T)) = x_i(T) - x_{i,f}$.

Time Optimal. If we constrain the terminal condition to $x(T) = x_f$, let the terminal time T be free (so that we can optimize over it) and choose $L(x, u) = 1$, we can find the *time-optimal* trajectory between an initial and final condition. This problem is usually only well-posed if we additionally constrain the inputs u to be bounded.

A very general set of conditions are available for the optimal control problem that captures most of these special cases in a unifying framework. Consider a nonlinear system

$$\begin{aligned} \dot{x} &= f(x, u) & x &= \mathbb{R}^n \\ x(0) &\text{ given} & u &\in \Omega \subset \mathbb{R}^p \end{aligned}$$

where $f(x, u) = (f_1(x, u), \dots f_n(x, u)) : \mathbb{R}^n \times \mathbb{R}^p \to \mathbb{R}^n$. We wish to minimize a cost function J with terminal constraints:

$$J = \int_0^T L(x, u)\, dt + V(x(T)), \qquad \psi(x(T)) = 0.$$

The function $\psi : \mathbb{R}^n \to \mathbb{R}^q$ gives a set of q terminal constraints. Analogous to the case of optimizing a function subject to constraints, we construct the *Hamiltonian*:

$$H = L + \lambda^T f = L + \sum \lambda_i f_i.$$

A set of necessary conditions for a solution to be optimal was derived by Pontryagin [PBGM62].

Theorem 2.1 (Maximum Principle)**.** *If* (x^*, u^*) *is optimal, then there exists* $\lambda^*(t) \in \mathbb{R}^n$ *and* $\nu^* \in \mathbb{R}^q$ *such that*

$$\dot{x}_i = \frac{\partial H}{\partial \lambda_i} \qquad -\dot{\lambda}_i = \frac{\partial H}{\partial x_i} \qquad \begin{aligned} &x(0) \text{ given, } \psi(x(T)) = 0 \\ &\lambda(T) = \frac{\partial V}{\partial x}(x(T)) + \nu^T \frac{\partial \psi}{\partial x} \end{aligned}$$

and

$$H(x^*(t), u^*(t), \lambda^*(t)) \leq H(x^*(t), u, \lambda^*(t)) \quad \textit{for all} \quad u \in \Omega$$

The form of the optimal solution is given by the solution of a differential equation with boundary conditions. If $u = \operatorname{argmin} H(x, u, \lambda)$ exists, we can use this to choose the control law u and solve for the resulting feasible trajectory that minimizes the cost. The boundary conditions are given by the n initial states $x(0)$, the q terminal constraints on the state $\psi(x(T)) = 0$ and the $n - q$ final values for the Lagrange multipliers

$$\lambda(T) = \frac{\partial V}{\partial x}(x(T)) + \nu^T \frac{\partial \psi}{\partial x}.$$

In this last equation, ν is a free variable and so there are n equations in $n+q$ free variables, leaving $n - q$ constraints on $\lambda(T)$. In total, we thus have $2n$ boundary values.

The maximum principle is a very general (and elegant) theorem. It allows the dynamics to be nonlinear and the input to be constrained to lie in a set Ω, allowing the possibility of bounded inputs. If $\Omega = \mathbb{R}^m$ (unconstrained input) and H is differentiable, then a necessary condition for the optimal input is

$$\frac{\partial H}{\partial u} = 0.$$

We note that even though we are *minimizing* the cost, this is still usually called the maximum principle (artifact of history).

Sketch of proof. We follow the proof given by Lewis and Syrmos [LS95], omitting some of the details required for a fully rigorous proof. We use the method of Lagrange multipliers, augmenting our cost function by the dynamical constraints and the terminal constraints:

$$\begin{aligned}\tilde{J}(x(\cdot), u(\cdot)) &= J(x, u) + \int_0^T \lambda^T(t)\big(\dot{x}(t) - f(x, u)\big)\, dt + \nu^T \psi(x(T), u(T)) \\ &= \int_0^T \big(L(x, u) + \lambda^T(t)\big(\dot{x}(t) - f(x, u)\big)\, dt \\ &\qquad\qquad + V(x(T), u(T)) + \nu^T \psi(x(T), u(T)).\end{aligned}$$

Note that λ is a function of time, with each $\lambda(t)$ corresponding to the instantaneous constraint imposed by the dynamics. The integral over the interval $[0, T]$ plays the role of the sum of the finite constraints in the regular optimization.

Making use of the definition of the Hamiltonian, the augmented cost becomes

$$\tilde{J}(x(\cdot), u(\cdot)) = \int_0^T \big(H(x, u) - \lambda^T(t)\dot{x}\big)\, dt + V(x(T), u(T)) + \nu^T \psi(x(T), u(T)).$$

We can now "linearize" the cost function around the optimal solution $x(t) = x^*(t) + \delta x(t)$, $u(t) = u^*(t) + \delta u(t)$. Using Leibnitz's rule, we have □

2.3 Examples

To illustrate the use of the maximum principle, we consider a number of analytical examples. Additional examples are given in the exercises.

Example 2.2 Scalar linear system
Consider the optimal control problem for the system

$$\dot{x} = ax + bu, \tag{2.1}$$

where $x = \mathbb{R}$ is a scalar state, $u \in \mathbb{R}$ is the input, the initial state $x(t_0)$ is given, and $a, b \in \mathbb{R}$ are positive constants. We wish to find a trajectory $(x(t), u(t))$ that minimizes the cost function

$$J = \tfrac{1}{2}\int_{t_0}^{t_f} u^2(t)\,dt + \tfrac{1}{2}cx^2(t_f),$$

where the terminal time t_f is given and $c > 0$ is a constant. This cost function balances the final value of the state with the input required to get to that position.

To solve the problem, we define the various elements used in the maximum principle. Our integrated and terminal costs are given by

$$L = \tfrac{1}{2}u^2(t) \qquad V = \tfrac{1}{2}cx^2(t_f).$$

We write the Hamiltonian of this system and derive the following expressions:

$$H = L + \lambda f = \tfrac{1}{2}u^2 + \lambda(ax + bu)$$

$$\dot{\lambda} = -\frac{\partial H}{\partial x} = -a\lambda, \qquad \lambda(t_f) = \frac{\partial V}{\partial x} = cx(t_f).$$

This is a final value problem for a linear differential equation and the solution can be shown to be

$$\lambda(t) = cx(t_f)e^{a(t_f - t)}$$

The optimal control is given by

$$\frac{\partial H}{\partial u} = u + b\lambda = 0 \quad \Rightarrow \quad u^*(t) = -b\lambda(t) = -bcx(t_f)e^{a(t_f - t)}.$$

Substituting this control into the dynamics given by equation (2.1) yields a first-order ODE in x:

$$\dot{x} = ax - b^2cx(t_f)e^{a(t_f - t)}.$$

This can be solved explicitly as

$$x^*(t) = x(t_o)e^{a(t - t_o)} + \frac{b^2c}{2a}x^*(t_f)\left[e^{a(t_f - t)} - e^{a(t + t_f - 2t_o)}\right].$$

Setting $t = t_f$ and solving for $x(t_f)$ gives

$$x^*(t_f) = \frac{2a\, e^{a(t_f - t_o)} x(t_o)}{2a - b^2 c \left(1 - e^{2a(t_f - t_o)}\right)}$$

and finally we can write

$$\begin{aligned} u^*(t) &= -\frac{2abc\, e^{a(2t_f - t_o - t)} x(t_o)}{2a - b^2 c \left(1 - e^{2a(t_f - t_o)}\right)} \\ x^*(t) &= x(t_o) e^{a(t - t_o)} + \frac{b^2 c\, e^{a(t_f - t_o)} x(t_o)}{2a - b^2 c \left(1 - e^{2a(t_f - t_o)}\right)} \left[e^{a(t_f - t)} - e^{a(t + t_f - 2t_o)} \right]. \end{aligned}$$

We can use the form of this expression to explore how our cost function affects the optimal trajectory. For example, we can ask what happens to the terminal state $x^*(t_f)$ and $c \to \infty$. Setting $t = t_f$ in equation (2.2) and taking the limit we find that

$$\lim_{c \to \infty} x^*(t_f) = 0.$$

∇

Example 2.3 Bang-bang control

The time-optimal control program for a linear system has a particularly simple solution. Consider a linear system with bounded input

$$\dot{x} = Ax + Bu, \qquad |u| \leq 1$$

and suppose we wish to minimize the time required to move from an initial state x_0 to a final state x_f. Without loss of generality we can take $x_f = 0$. We choose the cost functions and terminal constraints to satisfy

$$J = \int_0^T 1\, dt, \qquad \psi(x(T)) = x(T)$$

To find the optimal control, we form the Hamiltonian

$$H = 1 + \lambda^T (Ax + Bu) = 1 + (\lambda^T A)x + (\lambda^T B)u.$$

Now apply the conditions in the maximum principle:

$$\begin{aligned} \dot{x} &= \frac{\partial H}{\partial \lambda} = Ax + Bu \\ -\dot{\lambda} &= \frac{\partial H}{\partial x} = A^T \lambda \\ u &= \text{arg min } H = -\text{sgn}(\lambda^T B) \end{aligned}$$

The optimal solution always satisfies this equation (necessary condition) with $x(0) = x_0$ and $x(T) = 0$. It follows that the input is always $u = \pm 1 \implies$ "bang-bang". ∇

2.4 Linear Quadratic Regulators

The finite horizon, linear quadratic regulator (LQR) is given by

$$\begin{aligned}
\dot{x} &= Ax + Bu \qquad x \in \mathbb{R}^n, u \in \mathbb{R}^n, x_0 \text{ given} \\
\tilde{J} &= \frac{1}{2}\int_0^T \left(x^T Q_x x + u^T Q_u u\right)\, dt + \frac{1}{2}x^T(T)P_1 x(T)
\end{aligned}$$

where $Q_x \geq 0$, $Q_u > 0$, $P_1 \geq 0$ are symmetric, positive (semi-) definite matrices. Note the factor of $\frac{1}{2}$ is left out, but we included it here to simplify the derivation. Gives same answer (with $\frac{1}{2}x$ cost).

Solve via maximum principle:

$$\begin{aligned}
H &= x^T Q_x x + u^T Q_u u + \lambda^T(Ax + Bu) \\
\dot{x} &= \left(\frac{\partial H}{\partial \lambda}\right)^T \quad = Ax + Bu \qquad x(0) = x_0 \\
-\dot{\lambda} &= \left(\frac{\partial H}{\partial x}\right)^T \quad = Q_x x + A^T\lambda \qquad \lambda(T) = P_1 x(T) \\
0 &= \frac{\partial H}{\partial u} \quad = Q_u u + \lambda^T B \quad \implies \quad u = -Q_u^{-1}B^T\lambda.
\end{aligned}$$

This gives the optimal solution. Apply by solving *two point boundary value problem* (hard).

Alternative: guess the form of the solution, $\lambda(t) = P(t)x(t)$. Then

$$\begin{aligned}
\dot{\lambda} &= \dot{P}x + P\dot{x} = \dot{P}x + P(Ax - BQ_u^{-1}B^T P)x \\
-\dot{P}x - PAx + PBQ_u^{-1}BPx &= Q_x x + A^T Px.
\end{aligned}$$

This equation is satisfied if we can find $P(t)$ such that

$$-\dot{P} = PA + A^T P - PBQ_u^{-1}B^T P + Q_x \qquad P(T) = P_1$$

Remarks:

1. This ODE is called *Riccati ODE*.
2. Can solve for $P(t)$ backwards in time and then apply

   $$u(t) = -Q_u^{-1}B^T P(t)x.$$

   This is a (time-varying) *feedback* control $\implies$ tells you how to move from *any* state to the origin.
3. Variation: set $T = \infty$ and eliminate terminal constraint:

   $$\begin{aligned}
   J &= \int_0^\infty (x^T Q_x x + u^T Q_u u)\, dt \\
   u &= -\underbrace{Q_u^{-1}B^T P}_{K} x \qquad \text{Can show } P \text{ is } \textit{constant} \\
   0 &= PA + A^T P - PBQ_u^{-1}B^T P + Q_x
   \end{aligned}$$

This equation is called the *algebraic Riccati equation*.

4. In MATLAB, `K = lqr(A, B, Q_x, Q_u)`.

5. Require $Q_u > 0$ but $Q_x \geq 0$. Let $Q_x = H^T H$ (always possible) so that $L = \int_0^\infty x^T H^T H x + u^T Q_u u \, dt = \int_0^\infty \|Hx\|^2 + u^T Q_u u \, dt$. Require that (A, H) is *observable*. Intuition: if not, dynamics may not affect cost $\implies$ ill-posed.

2.5 Choosing LQR weights

$$\dot{x} = Ax + Bu \qquad J = \int_0^\infty \overbrace{\left(x^T Q_x x + u^T Q_u u + x^T S u\right)}^{L(x,u)} dt,$$

where the S term is almost always left out.

Q: How should we choose Q_x and Q_u?

1. Simplest choice: $Q_x = I$, $Q_u = \rho I \implies L = \|x\|^2 + \rho\|u\|^2$. Vary ρ to get something that has good response.

2. Diagonal weights

$$Q_x = \begin{bmatrix} q_1 & & \\ & \ddots & \\ & & q_n \end{bmatrix} \qquad Q_u = \rho \begin{bmatrix} r_1 & & \\ & \ddots & \\ & & r_n \end{bmatrix}$$

Choose each q_i to given equal effort for same "badness". E.g., $x_1 =$ distance in meters, $x_3 =$ angle in radians:

$$1 \text{ cm error OK} \implies q_1 = \left(\frac{1}{100}\right)^2 \qquad q_1 x_1^2 = 1 \text{ when } x_1 = 1 \text{ cm}$$

$$\frac{1}{60} \text{ rad error OK} \implies q_3 = (60)^2 \qquad q_3 x_3^2 = 1 \text{ when } x_3 = \frac{1}{60} \text{ rad}$$

Similarly with r_i. Use ρ to adjust input/state balance.

3. Output weighting. Let $z = Hx$ be the output you want to keep small. Assume (A, H) observable. Use

$$Q_x = H^T H \qquad Q_u = \rho I \qquad \implies \text{trade off } \|z\|^2 \text{ vs } \rho\|u\|^2$$

4. Trial and error (on *weights*)

2.6 Further Reading

Exercises

2.1 (a) Let $G_1, G_2, \ldots, G_k$ be a set of row vectors on a $\mathbb{R}^n$. Let F be another row vector on $\mathbb{R}^n$ such that for every $x \in \mathbb{R}^n$ satisfying $G_i x = 0$, $i = 1, \ldots, k$, we have $Fx = 0$. Show that there are constants $\lambda_1, \lambda_2, \ldots, \lambda_k$ such that

$$F = \sum_{i=1}^{k} \lambda_k G_k.$$

(b) Let $x^* \in \mathbb{R}^n$ be an the extremal point (maximum or minimum) of a function f subject to the constraints $g_i(x) = 0$, $i = 1, \ldots, k$. Assuming that the gradients $\partial g_i(x^*)/\partial x$ are linearly independent, show that there are k scalers λ_k, $i = 1, \ldots, n$ such that the function

$$\tilde{f}(x) = f(x) + \sum_{i=1}^{n} \lambda_i g_i(x)$$

has an extremal point at x^*.

2.2 Consider the following control system

$$\begin{aligned} \dot{q} &= u \\ \dot{Y} &= qu^T - uq^T \end{aligned}$$

where $u \in \mathbb{R}^m$ and $Y \in reals^{m \times m}$ is a skew symmetric matrix.

(a) For the fixed end point problem, derive the form of the optimal controller minimizing the following integral

$$\frac{1}{2} \int_0^1 u^T u \, dt.$$

(b) For the boundary conditions $q(0) = q(1) = 0$, $Y(0) = 0$ and

$$Y(1) = \begin{bmatrix} 0 & -y_3 & y_2 \\ y_3 & 0 & -y_1 \\ -y_2 & y_1 & 0 \end{bmatrix}$$

for some $y \in \mathbb{R}^3$, give an explicit formula for the optimal inputs u.

(c) (Optional) Find the input u to steer the system from $(0, 0)$ to $(0, \tilde{Y}) \in \mathbb{R}^m \times \mathbb{R}^{m \times m}$ where $\tilde{Y}^T = -\tilde{Y}$.

(Hint: if you get stuck, there is a paper by Brockett on this problem.)

2.3 In this problem, you will use the maximum principle to show that the shortest path between two points is a straight line. We model the problem by constructing a control system

$$\dot{x} = u$$

where $x \in \mathbb{R}^2$ is the position in the plane and $u \in \mathbb{R}^2$ is the velocity vector along the curve. Suppose we wish to find a curve of minimal length connecting $x(0) = x_0$ and $x(1) = x_f$. To minimize the length, we minimize the integral of the velocity along the curve,

$$J = \int_0^1 \sqrt{\|\dot{x}\|}\, dt,$$

subject to to the initial and final state constraints. Use the maximum principle to show that the minimal length path is indeed a straight line at maximum velocity. (Hint: minimizing $\sqrt{\|\dot{x}\|}$ is the same as minimizing $\dot{x}^T\dot{x}$; this will simplify the algebra a bit.)

2.4 Consider the optimal control problem for the system

$$\dot{x} = -ax + bu$$

where $x = \mathbb{R}$ is a scalar state, $u \in \mathbb{R}$ is the input, the initial state $x(t_0)$ is given, and $a, b \in \mathbb{R}$ are positive constants. (Note that this system is not quite the same as the one in Example **??**.) The cost function is given by

$$J = \tfrac{1}{2} \int_{t_0}^{t_f} u^2(t)\, dt + \tfrac{1}{2} c x^2(t_f),$$

where the terminal time t_f is given and c is a constant.

(a) Solve explicitly for the optimal control $u^*(t)$ and the corresponding state $x^*(t)$ in terms of t_0, t_f, $x(t_0)$ and t and describe what happens to the terminal state $x^*(t_f)$ as $c \to \infty$.

(b) Show that the system is differentially flat with appropriate choice of output(s) and compute the state and input as a function of the flat output(s).

(c) Using the polynomial basis $\{t^k,\ k = 0, \dots, M-1\}$ with an appropriate choice of M, solve for the (non-optimal) trajectory between $x(t_0)$ and $x(t_f)$. Your answer should specify the explicit input $u_d(t)$ and state $x_d(t)$ in terms of t_0, t_f, $x(t_0)$, $x(t_f)$ and t.

(d) Let $a = 1$ and $c = 1$. Use your solution to the optimal control problem and the flatness-based trajectory generation to find a trajectory between $x(0) = 0$ and $x(1) = 1$. Plot the state and input trajectories for each solution and compare the costs of the two approaches.

(e) (Optional) Suppose that we choose more than the minimal number of basis functions for the differentially flat output. Show how to use the additional degrees of freedom to minimize the cost of the flat trajectory and demonstrate that you can obtain a cost that is closer to the optimal.

2.5 Consider the optimal control problem for the system

$$\dot{x} = -ax^3 + bu$$

where $x = \mathbb{R}$ is a scalar state, $u \in \mathbb{R}$ is the input, the initial state $x(t_0)$ is given, and $a, b \in \mathbb{R}$ are positive constants. The cost function is given by

$$J = \tfrac{1}{2} \int_{t_0}^{t_f} u^2(t)\, dt + \tfrac{1}{2} c x^2(t_f),$$

where the terminal time t_f is given and c is a constant.

(a) Derive a set of differential equations for the optimal control $u^*(t)$ and the corresponding state $x^*(t)$ in terms of t_0, t_f, $x(t_0)$ and t. Be sure to provide any initial or final conditions required for your equations to be solved.

(b) Show that the system is differentially flat with appropriate choice of output(s) and compute the state and input as a function of the flat output(s).

(c) Using the polynomial basis $\{t^k,\, k = 0, \dots, M-1\}$ with an appropriate choice of M, solve for the (non-optimal) trajectory between $x(t_0)$ and $x(t_f)$. Your answer should specify the explicit input $u_d(t)$ and state $x_d(t)$ in terms of t_0, t_f, $x(t_0)$, $x(t_f)$ and t.

(d) Increase M by one and show how to choose the free parameter to minimize the cost function.

2.6 Consider the problem of moving a two-wheeled mobile robot (eg, a Segway) from one position and orientation to another. The dynamics for the system is given by the nonlinear differential equation

$$\begin{aligned}\dot{x} &= \cos\theta\, v \\ \dot{y} &= \sin\theta\, v \\ \dot{\theta} &= \omega\end{aligned}$$

where (x, y) is the position of the rear wheels, θ is the angle of the robot with respect to the x axis, v is the forward velocity of the robot and ω is spinning rate. We wish to choose an input (v, ω) that minimizes the time that it takes to move between two configurations (x_0, y_0, θ_0) and (x_f, y_f, θ_f), subject to input constraints $|v| \le L$ and $|\omega| \le M$.

Use the maximum principle to show that any optimal trajectory consists of segments in which the robot is traveling at maximum velocity in either the forward or reverse direction, and going either straight, hard left ($\omega = -M$) or hard right ($\omega = +M$).

Note: one of the cases is a bit tricky and can't be completely proven with the tools we have learned so far. However, you should be able to show the other cases and verify that the tricky case is possible.

2.7 Consider a linear system with input u and output y and suppose we wish to minimize the quadratic cost function

$$J = \int_0^\infty \left(y^T y + \rho u^T u \right) dt.$$

Show that if the corresponding linear system is observable, then the closed loop system obtained by using the optimal feedback $u = -Kx$ is guaranteed to be stable.

2.8 Consider the control system transfer function

$$H(s) = \frac{s+b}{s(s+a)} \qquad a, b > 0$$

with state space representation

$$\dot{x} = \begin{bmatrix} 0 & 1 \\ 0 & -a \end{bmatrix} x + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u$$
$$y = \begin{bmatrix} b & 1 \end{bmatrix} x$$

and performance criterion

$$V = \int_0^\infty (x_1^2 + u^2) dt.$$

(a) Let

$$P = \begin{bmatrix} p_{11} & p_{12} \\ p_{21} & p_{22} \end{bmatrix}$$

with $p_{12} = p_{21}$ and $P > 0$ (positive definite). Write the steady state Riccati equation as a system of four explicit equations in terms of the elements of P and the constants a and b.

(b) Find the gains for the optimal controller assuming the full state is available for feedback.

(c) Find the closed loop natural frequency and damping ratio.

2.9 The output $c(t)$ in a position-control system is governed by

$$J\ddot{c} = u,$$

where $u(t)$ is applied force.

(a) Write down a state space realization (find A and B).

(b) Use the matrix Riccati equation to find the feedback control law minimizing

$$\int_0^\infty (c^2 + q^2 u^2) dt.$$

(c) Show that the optimal control system has damping ratio $\frac{1}{\sqrt{2}}$.

(d) What is the corresponding optimal value of natural frequency?

(See AM05, Sec 4.4 if you don't remember how damping ratio (or factor) and natural frequency are defined.)

2.10 Consider the optimal control problem for the system

$$\dot{x} = ax + bu \qquad J = \tfrac{1}{2}\int_{t_0}^{t_f} u^2(t)\,dt + \tfrac{1}{2}cx^2(t_f),$$

where $x \in \mathbb{R}$ is a scalar state, $u \in \mathbb{R}$ is the input, the initial state $x(t_0)$ is given, and $a, b \in \mathbb{R}$ are positive constants. We take the terminal time t_f as given and let $c > 0$ be a constant that balances the final value of the state with the input required to get to that position. The optimal is derived in the lecture notes for week 6 and is shown to be

$$\begin{aligned} u^*(t) &= -\frac{2abc\,e^{a(2t_f-t_o-t)}x(t_o)}{2a - b^2c\left(1-e^{2a(t_f-t_o)}\right)} \\ x^*(t) &= x(t_o)e^{a(t-t_o)} + \frac{b^2c\,e^{a(t_f-t_o)}x(t_o)}{2a - b^2c\left(1-e^{2a(t_f-t_o)}\right)}\left[e^{a(t_f-t)} - e^{a(t+t_f-2t_o)}\right]. \end{aligned} \tag{2.2}$$

Now consider the infinite horizon cost

$$J = \tfrac{1}{2}\int_{t_0}^{\infty} u^2(t)\,dt$$

with $x(t)$ at $t = \infty$ constrained to be zero.

(a) Solve for $u^*(t) = -bPx^*(t)$ where P is the positive solution corresponding to the algebraic Riccati equation. Note that this gives an explicit feedback law ($u = -bPx$).

(b) Plot the state solution of the finite time optimal controller for the following parameter values

$$\begin{aligned} a &= 2 \qquad b = 0.5 \qquad x(t_0) = 4 \\ c &= 0.1,\ 10 \qquad t_f = 0.5,\ 1,\ 10 \end{aligned}$$

(This should give you a total of 6 curves.) Compare these to the infinite time optimal control solution. Which finite time solution is closest to the infinite time solution? Why?

2.11 In this problem we will explore the effect of constraints on control of the linear unstable system given by

$$\begin{aligned} \dot{x}_1 &= 0.8x_1 - 0.5x_2 + 0.5u \\ \dot{x}_2 &= x_1 + 0.5u \end{aligned}$$

subject to the constraint that $|u| \leq a$ where a is a postive constant.

(a) Ignore the constraint ($a = \infty$) and design an LQR controller to stabilize the system. Plot the response of the closed system from the initial condition given by $x = (1, 0)$.

(b) Use SIMULINK or `ode45` to simulate the the system for some finite value of a with an initial condition $x(0) = (1, 0)$. Numerically (trial and error) determine the smallest value of a for which the system goes unstable.

(c) Let $a_{\min}(\rho)$ be the smallest value of a for which the system is unstable from $x(0) = (\rho, 0)$. Plot $a_{\min}(\rho)$ for $\rho = 1, 4, 16, 64, 256$.

(d) *Optional*: Given $a > 0$, design and implement a receding horizon control law for this system. Show that this controller has larger region of attraction than the controller designed in part (b). (Hint: solve the finite horizon LQ problem analytically, using the bang-bang example as a guide to handle the input constraint.)

2.12 Consider the lateral control problem for an autonomous ground vehicle from Example 1.1. We assume that we are given a reference trajectory $r = (x_d, y_d)$ corresponding to the desired trajectory of the vehicle. For simplicity, we will assume that we wish to follow a straight line in the x direction at a constant velocity $v_d > 0$ and hence we focus on the y and θ dynamics:

$$\dot{y} = \sin\theta \, v_d$$
$$\dot{\theta} = \frac{1}{\ell} \tan\phi \, v_d.$$

We let $v_d = 10$ m/s and $\ell = 2$ m.

(a) Design an LQR controller that stabilizes the position y to the origin. Plot the step and frequency response for your controller and determine the overshoot, rise time, bandwidth and phase margin for your design. (Hint: for the frequency domain specifications, break the loop just before the process dynamics and use the resulting SISO loop transfer function.)

(b) Suppose now that $y_d(t)$ is not identically zero, but is instead given by $y_d(t) = r(t)$. Modify your control law so that you track $r(t)$ and demonstrate the performance of your controller on a "slalom course" given by a sinusoidal trajectory with magnitude 1 meter and frequency 1 Hz.

Chapter 3
Receding Horizon Control

Bibliography

[ÅM08] K. J. Åström and R. M. Murray. *Feedback Systems: An Introduction for Scientists and Engineers*. Princeton University Press, 2008. Available at `http://www.cds.caltech.edu/~murray/amwiki`.

[FLMR92] M. Fliess, J. Levine, P. Martin, and P. Rouchon. On differentially flat nonlinear systems. *Comptes Rendus des Séances de l'Académie des Sciences*, 315:619–624, 1992. Serie I.

[KKM91] I. Kanellakopoulos, P. V. Kokotovic, and A. S. Morse. Systematic design of adaptive controllers for feedback linearizable systems. *IEEE Transactions on Automatic Control*, 36(11):1241–1253, 1991.

[LS95] F. L. Lewis and V. L. Syrmos. *Optimal Control*. Wiley, second edition, 1995.

[MDP94] P. Martin, S. Devasia, and B. Paden. A different look at output tracking—Control of a VTOL aircraft. *Automatica*, 32(1):101–107, 1994.

[Mur97] R. M. Murray. Nonlinear control of mechanical systems: A Lagrangian perspective. *Annual Reviews in Control*, 21:31–45, 1997.

[PBGM62] L. S. Pontryagin, V. G. Boltyanskii, R. V. Gamkrelidze, and E. F. Mishchenko. *The Mathematical Theory of Optimal Processes*. Wiley-Interscience, 1962. (translated from Russian).

[Rug90] W. J. Rugh. Analytical framework for gain scheduling. In *Proc. American Control Conference*, pages 1688–1694, 1990.

[Sha90] J. S. Shamma. Analysis of gain scheduled control for nonlinear plants. *IEEE Transactions on Automatic Control*, 35(12):898–907, 1990.

[vNM98] M. J. van Nieuwstadt and R. M. Murray. Rapid hover to forward flight transitions for a thrust vectored aircraft. *Journal of Guidance, Control, and Dynamics*, 21(1):93–100, 1998.

[vNRM94] M. van Nieuwstadt, M. Rathinam, and R. M. Murray. Differential flatness and absolute equivalence. Technical Report CIT/CDS 94-006, California Institute of Technology, 1994. Submitted *SIAM J. Control and Optimization*. Available electronically from `http://cds.caltech.edu/cds/reports`.