Optimal Control

The optimal control problem can be described by introducing the system dynamics

$$\dot{\boldsymbol{x}} = \boldsymbol{F}(\boldsymbol{x}, \boldsymbol{u})$$

which is assumed to start in an initial state given by

$$\boldsymbol{x}(0) = \boldsymbol{x}_0$$

and has controllable parameters $\boldsymbol{u}$

$$\boldsymbol{u} \in U$$

The objective function consists of a function of the final state $\psi[\boldsymbol{x}(T)]$ and a cost function (or *loss function*) ℓ that is integrated over time.

$$J = \psi[\boldsymbol{x}(T)] + \int_0^T \ell(\boldsymbol{u}, \boldsymbol{x}) dt$$

The simplest optimization problem is depicted in Figure 1:

$$\min_{\boldsymbol{x}} \tilde{F}(\boldsymbol{x})$$

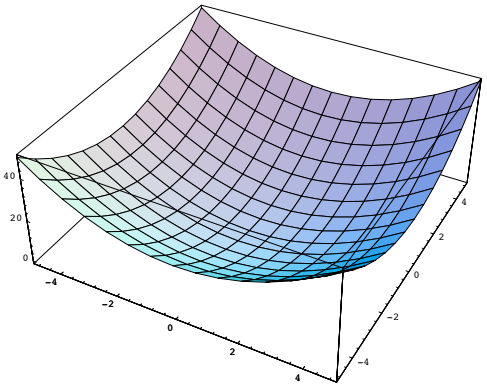

Figure 1: A simple case for optimization: a function of two variables has a single minimum at the point $\mathbf{x}^*$.

Constraints

The simplest kind of constraint is an equality constraint, for example, $g(\boldsymbol{x}) = 0$. Formally this addition is stated as

$$\min_{\boldsymbol{x}} \tilde{F}(\boldsymbol{x}) \text{ subject to } G(\boldsymbol{x}) = 0$$

This complication can be handled by the *method of Lagrange*. This method reduces the constrained problem to a new, unconstrained minimization problem with additional variables. The additional variables are known as *Lagrange multipliers*. To handle this problem, add $g(\boldsymbol{x})$ to $\tilde{f}(\boldsymbol{x})$ using a Lagrange multiplier λ:

$$F(\boldsymbol{x}, \lambda) = \tilde{F}(\boldsymbol{x}) + \lambda G(\boldsymbol{x})$$

The Lagrange multiplier is an extra scalar variable, so the number of degrees of freedom of the problem has increased, but the advantage is that now simple, unconstrained minimization techniques can be applied to the composite function. The problem becomes

$$\min_{\boldsymbol{x},\lambda} F(\boldsymbol{x}, \lambda)$$

Most often the equality constraint is the dynamic equations of the system itself, $\dot{\boldsymbol{x}} = \boldsymbol{F}(\boldsymbol{x}, \boldsymbol{u})$, and the problem is to find an optimal trajectory $\boldsymbol{x}(t)$. With this extension the problem assumes the generality of the optimal control problem.

There are two straightforward ways to solve the optimal control problem: (1) the method of Lagrange multipliers and (2) dynamic programming. We have already outlined the idea behind the Lagrange multipliers approach. The second way, dynamic programming, solves the constrained problem directly. The key is the *principle of optimality* This remarkably elegant observation leads to a simple way of solving the optimization problem directly that works especially well when all the variables are discrete.

> From any point on an optimal state space trajectory, the remaining trajectory is optimal for the corresponding problem initiated at that point.

Minimization Algorithms

The techniques of this chapter are all directed toward finding a *local minimum*.

This can be done by starting from a point in the state space and making changes in the state vector that improve the objective function. There are two principal ways to do so.

One is to try for an algebraic solution. At the minimum $\frac{df}{dx} = 0$. We can calculate this derivative and attempt to solve for x. For example, if $f(x)$ is the following quadratic function

$$f(x) = 3x^2 + x - 4$$

taking the derivative

$$\frac{df}{dx} = 6x + 1 = 0$$

allows the direct solution of the extremum as

$$x = -\frac{1}{6}$$

This is a minimum because

$$\frac{d^2f}{dx^2} = 6 > 0$$

The Hessian

When $\boldsymbol{x}$ is a vector, the *Hessian* matrix A_n can be used to test for a maximum or minimum.

The Hessian can also be related to convexity. The function $f(\boldsymbol{x})$ being convex is equivalent to $\boldsymbol{x}^T A\boldsymbol{x} \geq 0$ for all $\boldsymbol{x} \neq \mathbf{0}$.

Where

$$A_n = \begin{bmatrix} f_{x_1^2} & f_{x_1x_2} & f_{x_1x_3} & \cdots \\ f_{x_2x_1} & f_{x_2^2} & \cdots & \\ \vdots & & & \\ f_{x_nx_1} & \cdots & & f_{x_n^2} \end{bmatrix}$$

the condition for a minimum generalizes to a condition on the determinant of $A_k, k = 1, \ldots, n$, that is,

$$|A_k| > 0$$

for all $k = 1, \ldots, n$. Similarly, the condition for a maximum generalizes to

$$(-1)^{k+1}|A_k| < 0$$

for all $k = 1, \ldots, n$.

Gradient Minimization

The second way of minimizing a function $f(x)$, which is of most practical importance in modeling biological systems, is to solve the problem iteratively using the gradient. Given a starting point and the derivative at that point, we can produce a new point that is closer to the desired minimum using

$$x^{new} = x^{old} - \eta \frac{df(x)}{dx}$$

The parameter η controls the size of the change in state vector and is very important. If it is too small, then the convergence to the optimum may be slow. If it is too large, then the algorithm may not converge at all.

Ex 1: Minimizing a Quadratic As an elementary example, let's reconsider the problem of finding the minimum of

$$f(x) = 3x^2 + x - 4$$

Pick a starting point of $x^{old} = 1$ and a learning rate of $\eta = 0.05$.

$$\frac{df}{dx} = 6x + 1$$

$$\begin{aligned} x^{new} &= x^{old} - \eta \left(\frac{df}{dx}\right)_{x_{old}} \\ &= 1 - 0.05(7) = 0.65 \end{aligned}$$

Repeating this calculation again shows that the estimate is approaching the minimum of $x = -0.16667$.

$$\begin{aligned} x^{new} &= x^{old} - \eta \left(\frac{df}{dx}\right)_{x_{old}} \\ &= 0.65 - 0.05(4.9) = 0.405 \end{aligned}$$

Ex 2: A Single-Layered Network with Linear Activation Consider a single-layered network with two inputs and one output, as shown in Figure 2. The network uses a linear summation activation function thus the output is determined by

$$y = \sum_k w_k x_k$$

The problem is to "train" the network to produce an output y for each of a series of input patterns $\boldsymbol{x}^p, p = 1, \ldots, n$ such that the error between the desired output for each pattern, $y^p, p = 1, \ldots, n$ and the actual output y is minimized.

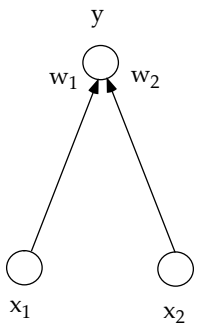

Figure 2: A simple case for optimization: a function of one variable minimizes error. Earlier a similar network was used to realize the computation of eigenvectors. There the weights for the network could be precomputed. Here they are determined by gradient descent.

The variables in this problem are the weights in the network. They are to be adjusted to minimize the error. Define an error function

$$E(\boldsymbol{w}) = \frac{1}{2} \sum_p (y^p - y)^2$$

which is the same as

$$E(\boldsymbol{w}) = \frac{1}{2} \sum_p \left(y^p - \sum_k w_k x_k^p \right)^2$$

Ex 2 Cont.

One way to minimize the cost function $E(\boldsymbol{w})$ is to use gradient search. Starting from an initial guess use the derivative $\frac{\partial E(\boldsymbol{w})}{\partial w_k}$ to successively improve the guess.

$$w_k^{new} = w_k^{old} - \alpha \frac{\partial E(\boldsymbol{w})}{\partial w_k}$$

The parameter η controls the size of the change at each step. The changes are discontinued when the improvements become very small. One way of measuring the improvement is $\sum_k \|w_k^{new} - w_k^{old}\| < \varepsilon$. For an appropriately chosen η this procedure will converge to a *local minimum* of $E(\boldsymbol{w})$.

Since E is a function of other variables besides $\boldsymbol{w}$, the *partial derivative* is used in the preceding equation:

$$\frac{\partial E(\boldsymbol{w})}{\partial w_k} = -\sum_p (y^p - y) x_k$$

This is known as the *Widrow-Hoff learning rule*.

The Method of Lagrange Multipliers

Minimization is often complicated by the addition of constraints. The simplest kind of constraint is an equality constraint, for example, $G(x) = 0$. Formally this addition is stated as

$$\min_x \tilde{F}(x) \text{ subject to } G(x) = 0$$

The method of Lagrange reduces the constrained problem to a new, unconstrained minimization problem with additional variables. The additional variables are known as Lagrange multipliers. To handle this problem, append $G(x)$ to the function $\tilde{F}(x)$ using a Lagrange multiplier λ:

$$F(x, \lambda) = \tilde{F}(x) + \lambda G(x)$$

The Lagrange multiplier is an extra scalar variable, so the number of degrees of freedom of the problem has increased, but the plus side is that now simple, unconstrained minimization techniques can be applied to the composite function. The problem becomes

$$\min_{x,\lambda} F(x, \lambda)$$

Ex 3: The Closest Point to a Circle Find the closest point from the point $(2, 2)$ to the circle $x^2 + y^2 = 1$. That is,

$$\min_{x,y}(x-2)^2 + (y-2)^2 \text{ subject to } x^2 + y^2 - 1 = 0$$

Solution. Append the constraint using a Lagrange multiplier

$$\min_{x,\lambda} F(x, \lambda) = (x-2)^2 + (y-2)^2 + \lambda(x^2 + y^2 - 1)$$

Now differentiate with respect to variables x, y, and λ, and set the partial derivatives equal to zero for a local extremum.

$$F_x = 2(x-2) + 2\lambda x = 0$$

$$F_y = 2(y-2) + 2\lambda y = 0$$

$$F_\lambda = x^2 + y^2 - 1 = 0$$

Solve for x and y in the first two equations in terms of λ and substitute these solutions into the third,

$$\left(\frac{2}{1+\lambda}\right)^2 + \left(\frac{2}{1+\lambda}\right)^2 = 1$$

$$\lambda = 2\sqrt{2} - 1$$

So that finally,

$$(x, y) = \left(\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}\right)$$

Interpreting Lagrange Multipliers Solving the unconstrained problem with the Lagrange multipliers is equivalent to solving the constrained problem. To see this, consider the problem

$$\min_{x,y} F(x,y) \text{ subject to } G(x,y) = 0$$

and suppose for a moment that the constraint equation could be solved so that

$$y = h(x)$$

In that case y can be eliminated from $F(x,y)$, reducing the problem to

$$\min_{x} F[x, h(x)]$$

which can be differentiated using the chain rule to obtain

$$F_x + F_y \frac{dy}{dx} = 0 \tag{1}$$

Now consider moving along the curve $G(x,y) = 0$. Let s be a parameter that varies with arc length so that $\frac{dG}{ds} = 0$, or alternatively

$$G_x \frac{dx}{ds} + G_y \frac{dy}{ds} = 0$$

Solving for $\frac{dy}{dx}$ and substituting into Equation 1,

$$F_x G_y = F_y G_x \tag{2}$$

So this equation must hold for the constrained problem.

Lagrange Multipliers Cont. Now consider the unconstrained problem using the Lagrange multiplier,

$$F'(x, y, \lambda) = F(x, y) + \lambda G(x, y)$$

Differentiating with respect to x and y yields

$$F_x + \lambda G_x = 0$$

and

$$F_y + \lambda G_y = 0$$

Eliminating λ gives the desired result,

$$F_x G_y = F_y G_x$$

Thus the equation that defines the extrema in the unconstrained problem using Lagrange multipliers is the same as Equation 2, which was obtained by solving for the extrema in the constrained problem.

Geometric Interpretation

Equation 2 also has a nice interpretation in terms of geometry. Rewrite it as

$$\frac{F_x}{F_y} = \frac{G_x}{G_y}$$

What this says is that at the extremum, the gradient must be in the same direction as the gradient of level curves of $G = constant$. If this were not true, then one could improve F by sliding along $G = 0$ in the appropriate direction. This relationship is shown in Figure 3.

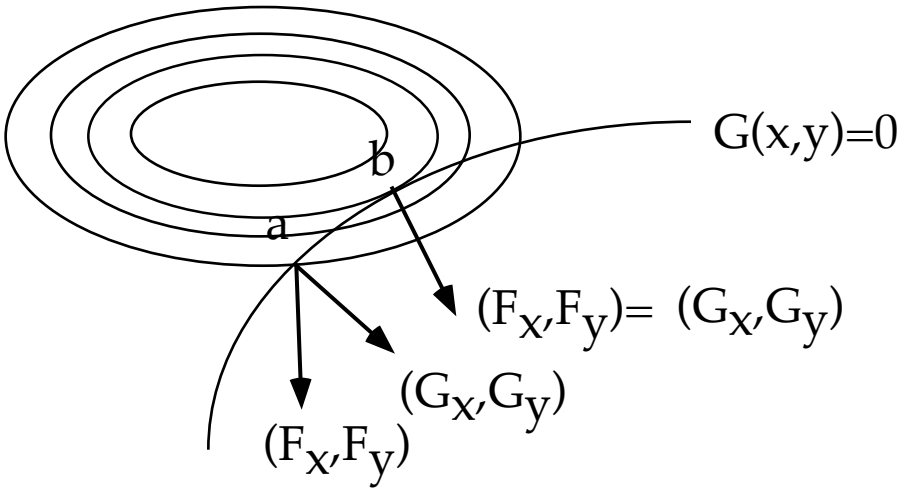

Figure 3: Interpretation of the constraint obtained by using Lagrange multipliers. At the local optimum point (b), $(F_x, F_y) = (G_x, G_y)$. At any other point, such as a, one can improve F by sliding along $G = 0$ in the appropriate direction.

Optimal Control

All physical systems will have a dynamics with associated parameters. So a ubiquitous problem is to pick those parameters to maximize some objective function.

Formally the dynamics can be described by

$$\dot{\boldsymbol{x}} = \boldsymbol{F}(\boldsymbol{x}, \boldsymbol{u})$$

which starts at an initial state given by

$$\boldsymbol{x}(0) = \boldsymbol{x}_0$$

and has controllable parameters $\boldsymbol{u}$

$$\boldsymbol{u} \in U$$

The objective function consists of a function of the final state $\psi[\boldsymbol{x}(T)]$ and a cost function (or *loss function*) ℓ that is integrated over time.

$$J = \psi[\boldsymbol{x}(T)] + \int_0^T \ell(\boldsymbol{u}, \boldsymbol{x}) dt$$

We will derive two principal ways of solving this problem that are at the core of the learning algorithms to come later.

Dynamic Programming

The second main method of solving the optimal control problem is a direct method that works best for discrete systems. The first step is to convert the formulation of the problem to a discrete $\boldsymbol{x}(k)$ and $\boldsymbol{u}(k), k = 0, \ldots, N-1$.

The dynamics are now expressed by a difference equation,

$$\boldsymbol{x}(k+1) = f[\boldsymbol{x}(k), \boldsymbol{u}(k)]$$

The initial condition is:

$$\boldsymbol{x}(0) = \boldsymbol{x}_0$$

The allowable control is also discrete:

$$\boldsymbol{u}(k) \in U, k = 0, \ldots, N$$

Finally, the integral in the objective function is replaced by a sum:

$$J = \psi[\boldsymbol{x}(T)] + \sum_{0}^{N-1} \ell[\boldsymbol{u}(k), \boldsymbol{x}(k)]$$

To solve this equation directly recall that the principle of optimality states that from any point on an optimal trajectory, the remaining trajectory is optimal for the corresponding problem initiated at that point. A way of using this principle is to start at one time step just before the end, that is, at $k = N-1$, and calculate the best control for each possible state $\boldsymbol{x}(k)$. Once this calculation has been done, back up one step to $k = N-2$. The principle states that the steps from $k = N-1$ should be part of the solution so that they can be utilized in building the optimal solution from step $N-2$. Along the way, partial results of what is the best thing to do must be kept for every point in the state space that is examined. Let us let $V(\boldsymbol{x}, k)$ keep track of these partial results.

Now it's easy to see that at the end,

$$V(\boldsymbol{x}, N) = \psi[\boldsymbol{x}(N)]$$

One step back,

$$V(\boldsymbol{x}, N-1) = \max_{\boldsymbol{u}\in U}\{\ell[\boldsymbol{u}(N-1), \boldsymbol{x}(N-1)] + \psi[\boldsymbol{x}(N)]\}$$

And in general, for $k < N-1$,

$$V(\boldsymbol{x}, k-1) = \max_{\boldsymbol{u}\in U}\{\ell[\boldsymbol{u}(k-1), \boldsymbol{x}(k-1)] + V[\boldsymbol{x}(k), k]\}, k = 0, \ldots, N$$

These equations are elegant in their simplicity but suffer from the "curse of dimensionality," in that as the state space increases in size, the amount of memory needed for V increases as M^d, where M is the discretization used for each state variable and d is the dimension of the space. The extent of the calculations can be appreciated by consulting Figure 4. The first pass in dynamic programming is the calculation of the objective function or value $V(\boldsymbol{x}, k)$ for each point in the state space and for each discrete time, from $k = 1, \ldots, T$. The top part of this figure shows the calculations proceeding backward in time. From the penultimate state you can try out all the possible controls and pick the one that best maximizes $V[\boldsymbol{x}(k-1), k-1]$. Next the calculations are repeated for the state space at $k-2$ and so on, back to the initial time $k = 0$. Along the way the values of $\boldsymbol{u}$ that maximize V are saved. Next, at time $k = 0$, the recovery of the best control starts with the recovery of the best control for the particular initial condition $\boldsymbol{x}(0)$. That $\boldsymbol{u}$ generates the state $\boldsymbol{x}(1)$, and the best control from that state is recovered. The recovery of the trajectory proceeds until the final time is reached, as shown in the lower part of the figure.

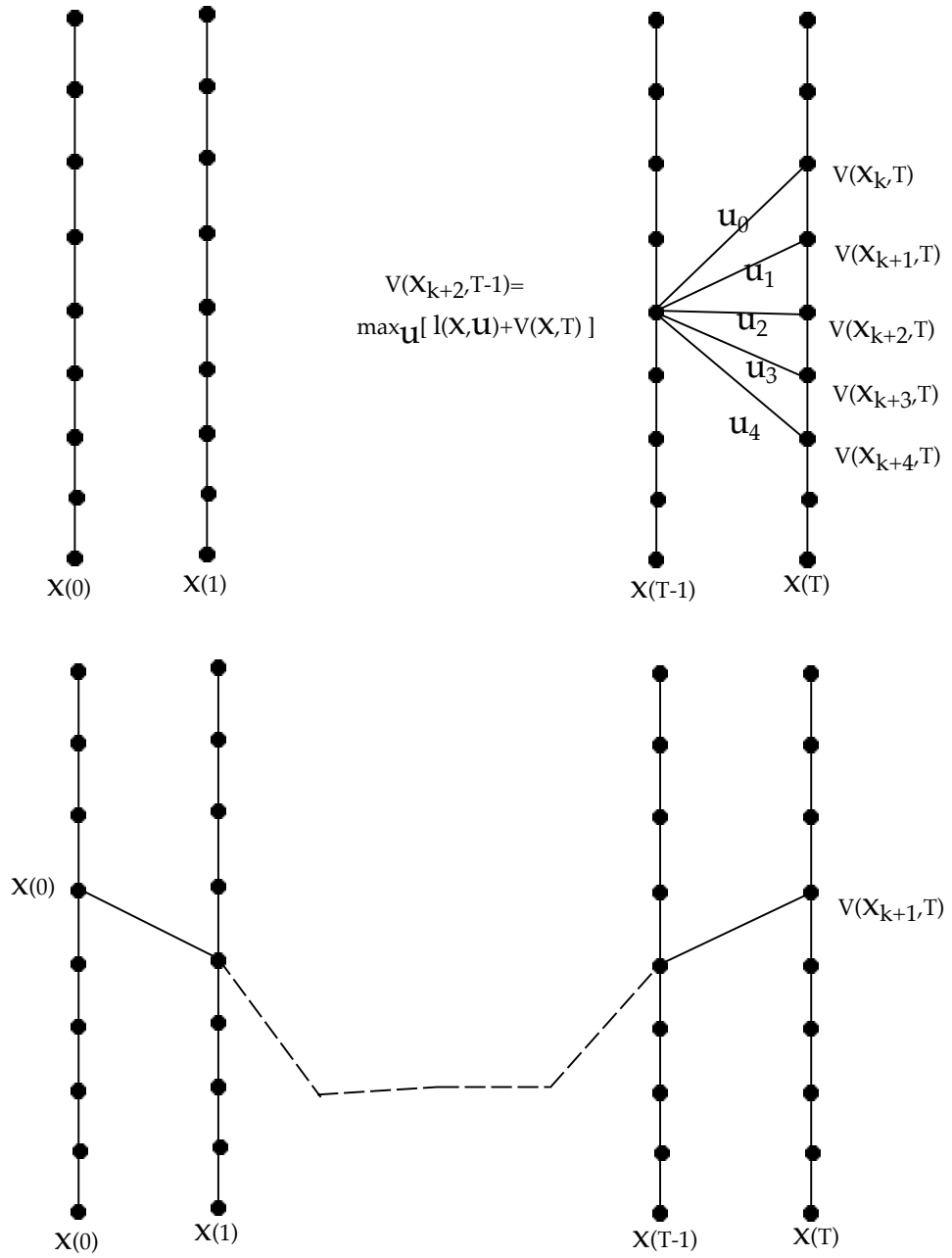

Figure 4: The basic computations in dynamic programming can be seen as a stage-wise process that starts at the terminal time and proceeds back to the initial time. (*upper*) The computation of the value for each node in the state space. During this calculation the control that produces the best value is also saved. (*lower*) At the end of the calculation, the remembered best controls can be used to recover the state space trajectory.

Ex 5: The Cart Revisited Consider a discrete version of the cart problem where the state equations are given by

$$x(k+1) = x(k) + v(k)$$

$$v(k+1) = v(k) + u(k)$$

and the cost function is

$$J = x(T) - \sum_{0}^{T-1} \frac{u(k)^2}{2}$$

with $T = 9$ and $N = 9$.

The results are shown in Figure 5a–c. Note that the different form of the dynamic equations results in a solution that is slightly different from the one found analytically using the Hamiltonian. However, the qualitative features of the solution are the same: Accelerate the cart at the beginning and then back off at the end.

Remember that what we have been calling the state vector, $\mathbf{x}$, here for the cart example is actually composed of position and velocity, so that

$$\mathbf{x}(k) = (x(k), v(k))$$

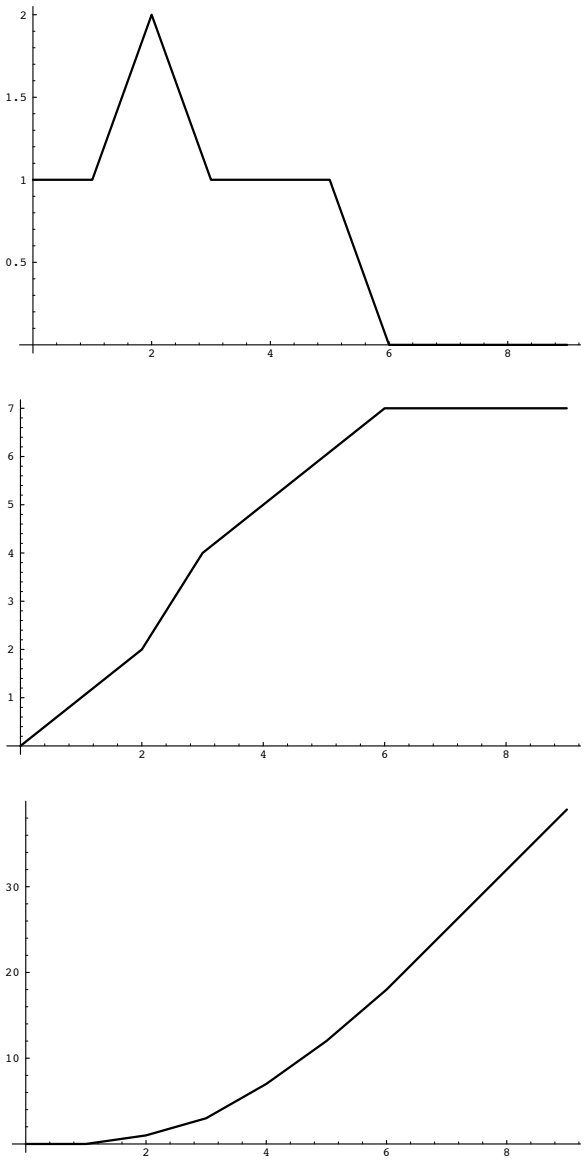

Figure 5: The optimal solution found by dynamic programming: (a) acceleration profile; (b) velocity profile; (c) distance profile.

The Euler-Lagrange Method

The goal of this section is to determine the conditions for the control to maximize the objective function J.

$$\max_{\boldsymbol{u}} J \text{ subject to the constraint } \dot{\boldsymbol{x}} = \boldsymbol{F}(\boldsymbol{x}, \boldsymbol{u})$$

The strategy will be to assume that $\boldsymbol{u}$ maximizes J and then use this assumption to derive other conditions for a maximum. These arguments depend on making a perturbation in $\boldsymbol{u}$ and seeing what happens. Since $\boldsymbol{u}$ affects $\boldsymbol{x}$, the calculations become a little involved, but the argument is just a matter of careful bookkeeping. The main trick is to add additional terms to J that sum to zero. Let's start by appending the dynamic equation to J as before, but this time using continuous Lagrange multipliers $\boldsymbol{\lambda}(t)$:

$$\bar{J} = J - \int_0^T \boldsymbol{\lambda}^T [\dot{\boldsymbol{x}} - \boldsymbol{F}(\boldsymbol{x}, \boldsymbol{u})] dt$$

Anticipating what is about to happen, we define the *Hamiltonian* $H(\boldsymbol{\lambda}, \boldsymbol{x}, \boldsymbol{u})$ as

$$H(\boldsymbol{\lambda}, \boldsymbol{x}, \boldsymbol{u}) \equiv \boldsymbol{\lambda}^T [\boldsymbol{F}(\boldsymbol{x}, \boldsymbol{u})] + \ell(\boldsymbol{x}, \boldsymbol{u})$$

so that the expression for $\bar{J}$ becomes

$$\bar{J} = \psi[\boldsymbol{x}(T)] + \int_0^T [H(\boldsymbol{\lambda}, \boldsymbol{x}, \boldsymbol{u}) - \boldsymbol{\lambda}^T \dot{\boldsymbol{x}}] dt$$

Now let's examine the effects of a small change in $\boldsymbol{u}$, as shown in Figure 6 on $\bar{J}$, just keeping track of the change $\delta\bar{J}$:

$$\delta\bar{J} = \psi[\boldsymbol{x}(T)+\delta\boldsymbol{x}(T)]-\psi[\boldsymbol{x}(T)]+\int_0^T [H(\boldsymbol{\lambda},\boldsymbol{x}+\delta\boldsymbol{x},\boldsymbol{v})-H(\boldsymbol{\lambda},\boldsymbol{x},\boldsymbol{u})-\boldsymbol{\lambda}^T\delta\dot{\boldsymbol{x}}]dt$$

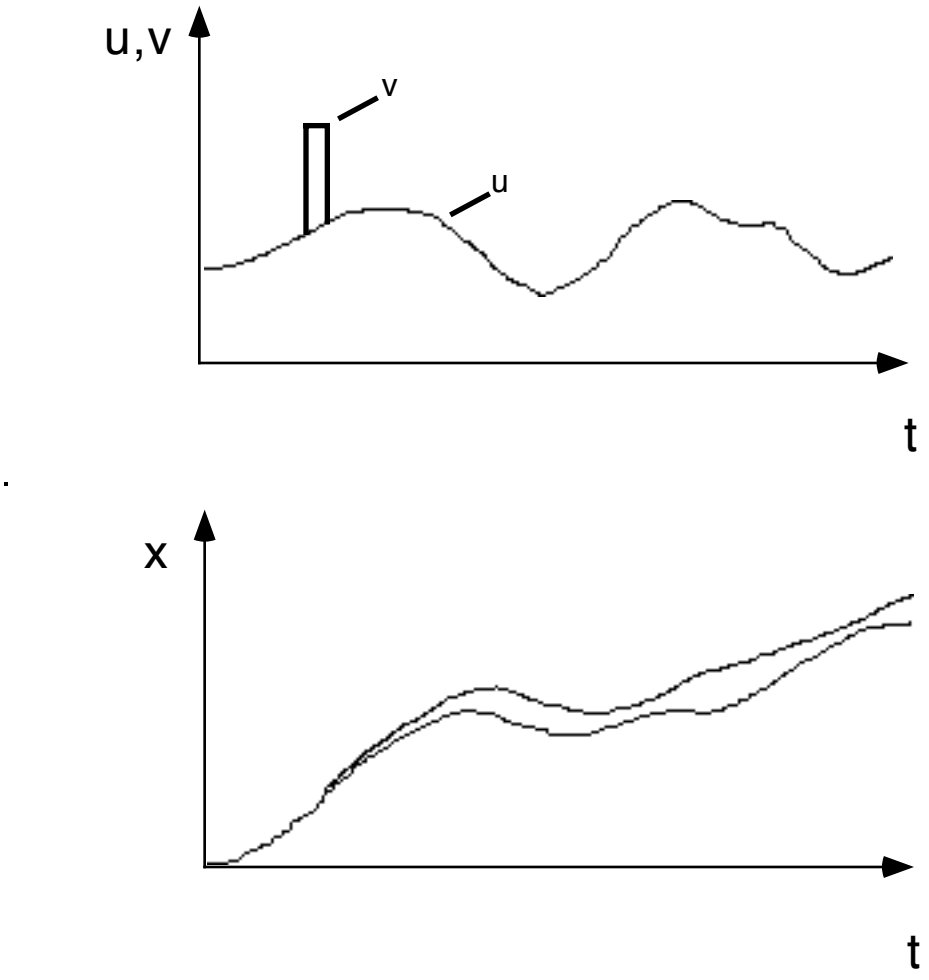

Figure 6: The maximum condition for $u(t)$ can be developed by studying a small perturbation about a trajectory that is assumed optimal.

Using the expression for integration by parts for $\int \boldsymbol{\lambda}^T \delta\dot{\boldsymbol{x}} dt$:

$$\int_0^T \boldsymbol{\lambda}^T \delta\dot{\boldsymbol{x}}\, dt = \boldsymbol{\lambda}^T(T)\delta\boldsymbol{x}(T) - \boldsymbol{\lambda}^T(0)\delta\boldsymbol{x}(0) - \int_0^T \dot{\boldsymbol{\lambda}}^T \delta\boldsymbol{x}\, dt$$

Now substitute this into the expression for $\delta\bar{J}$,

$$\begin{aligned}\delta\bar{J} \;=\; & \psi[\boldsymbol{x}(T) + \delta\boldsymbol{x}(T)] - \psi[\boldsymbol{x}(T)] - \boldsymbol{\lambda}(T)^T\delta\boldsymbol{x}(T) \\ & + \int_0^T [H(\boldsymbol{\lambda}, \boldsymbol{x} + \delta\boldsymbol{x}, \boldsymbol{v}) - H(\boldsymbol{\lambda}, \boldsymbol{x}, \boldsymbol{u}) + \dot{\boldsymbol{\lambda}}^T \delta\boldsymbol{x}]dt\end{aligned}$$

Now concentrate just on the first two terms in the integral:

$$\int_0^T [H(\boldsymbol{\lambda}, \boldsymbol{x} + \delta\boldsymbol{x}, \boldsymbol{v}) - H(\boldsymbol{\lambda}, \boldsymbol{x}, \boldsymbol{u})]dt$$

First add and subtract $H(\boldsymbol{\lambda}, \boldsymbol{x}, \boldsymbol{v})$:

$$= \int_0^T [H(\boldsymbol{\lambda}, \boldsymbol{x} + \delta\boldsymbol{x}, \boldsymbol{v}) - H(\boldsymbol{\lambda}, \boldsymbol{x}, \boldsymbol{v}) + H(\boldsymbol{\lambda}, \boldsymbol{x}, \boldsymbol{v}) - H(\boldsymbol{\lambda}, \boldsymbol{x}, \boldsymbol{u})]dt$$

Next expand the first term inside the integral in a Taylor series and neglect terms above first order,

$$\cong \int_0^T (H_{\boldsymbol{x}}(\boldsymbol{\lambda}, \boldsymbol{x}, \boldsymbol{v})^T \delta\boldsymbol{x} + H(\boldsymbol{\lambda}, \boldsymbol{x}, \boldsymbol{v}) - H(\boldsymbol{\lambda}, \boldsymbol{x}, \boldsymbol{u}))dt$$

where $H_{\boldsymbol{x}}$ is the partial $\frac{\partial H}{\partial \boldsymbol{x}}$, which is

$$\begin{pmatrix} H_{x_1} \\ \vdots \\ H_{x_n} \end{pmatrix}$$

Now add and subtract $H_{\boldsymbol{x}}(\boldsymbol{\lambda}, \boldsymbol{x}, \boldsymbol{u})^T \delta \boldsymbol{x}$:

$$= \int_0^T \{H_{\boldsymbol{x}}(\boldsymbol{\lambda}, \boldsymbol{x}, \boldsymbol{u})^T \delta \boldsymbol{x} + [H_{\boldsymbol{x}}(\boldsymbol{\lambda}, \boldsymbol{x}, \boldsymbol{v}) - H_{\boldsymbol{x}}(\boldsymbol{\lambda}, \boldsymbol{x}, \boldsymbol{u})]^T \delta \boldsymbol{x} + H(\boldsymbol{\lambda}, \boldsymbol{x}, \boldsymbol{v}) - H(\boldsymbol{\lambda}, \boldsymbol{x}, \boldsymbol{u})\} dt$$

The term $[H_{\boldsymbol{x}}(\boldsymbol{\lambda}, \boldsymbol{x}, \boldsymbol{v}) - H_{\boldsymbol{x}}(\boldsymbol{\lambda}, \boldsymbol{x}, \boldsymbol{u})]^T \delta \boldsymbol{x}$ can be neglected because it is the product of two small terms, $[H_{\boldsymbol{x}}(\boldsymbol{\lambda}, \boldsymbol{x}, \boldsymbol{v}) - H_{\boldsymbol{x}}(\boldsymbol{\lambda}, \boldsymbol{x}, \boldsymbol{u})]$ and $\delta \boldsymbol{x}$, and thus is a second-order term. Thus

$$\cong \int_0^T (H_{\boldsymbol{x}}(\boldsymbol{\lambda}, \boldsymbol{x}, \boldsymbol{u}) \delta \boldsymbol{x} + H(\boldsymbol{\lambda}, \boldsymbol{x}, \boldsymbol{v}) - H(\boldsymbol{\lambda}, \boldsymbol{x}, \boldsymbol{u})) dt$$

Finally, substitute this expression back into the original equation for δJ, yielding

$$\begin{aligned} \delta \bar{J} \cong & \{\psi_{\boldsymbol{x}}[\boldsymbol{x}(T)] - \boldsymbol{\lambda}^T(T)\} \delta \boldsymbol{x}(T) \\ & + \int_0^T [H_{\boldsymbol{x}}(\boldsymbol{\lambda}, \boldsymbol{x}, \boldsymbol{u}) + \dot{\boldsymbol{\lambda}}^T] \delta \boldsymbol{x} \, dt \\ & + \int_0^T [H(\boldsymbol{\lambda}, \boldsymbol{x}, \boldsymbol{v}) - H(\boldsymbol{\lambda}, \boldsymbol{x}, \boldsymbol{u})] dt \end{aligned}$$

Since we have the freedom to pick $\boldsymbol{\lambda}$, just to make matters simpler, pick it so that the first integral vanishes:

$$-\dot{\boldsymbol{\lambda}}^T = H_{\boldsymbol{x}}(\boldsymbol{\lambda}, \boldsymbol{x}, \boldsymbol{u})$$

$$\boldsymbol{\lambda}^T(T) = \psi_{\boldsymbol{x}}[\boldsymbol{x}(T)]$$

Now all $\delta\bar{J}$ has left is

$$\delta\bar{J} = \int_0^T [H(\boldsymbol{\lambda}, \boldsymbol{x}, \boldsymbol{v}) - H(\boldsymbol{\lambda}, \boldsymbol{x}, \boldsymbol{u})]dt$$

From this equation it follows that the optimal control $\boldsymbol{u}^*$ must be such that

$$H(\boldsymbol{\lambda}, \boldsymbol{x}, \boldsymbol{u}^*) \geq H(\boldsymbol{\lambda}, \boldsymbol{x}, \boldsymbol{u}),\ \boldsymbol{u} \in U \tag{3}$$

To see this point, suppose that it were not true, that is, that for some interval of time there was a $\boldsymbol{v} \in U$ such that

$$H(\boldsymbol{\lambda}, \boldsymbol{x}, \boldsymbol{v}) \geq H(\boldsymbol{\lambda}, \boldsymbol{x}, \boldsymbol{u}^*)$$

This assumption would mean that you could adjust the integral so that the perturbation $\delta\bar{J}$ is positive, contradicting the original assumption that $\bar{J}$ is maximimized by $\boldsymbol{u}^*$. Therefore Equation 3 must hold.

Conditions for Optimality

,

In addition to the dynamic equations

$$\dot{\boldsymbol{x}} = f(\boldsymbol{x}, \boldsymbol{u})$$

and associated initial condition

$$\boldsymbol{x}(0) = \boldsymbol{x}_0$$

the Lagrange multipliers also must obey a constraint equation

$$-\dot{\boldsymbol{\lambda}}^T = H_{\boldsymbol{x}}$$

that has a *final condition*

$$\boldsymbol{\lambda}^T(T) = \psi_{\boldsymbol{x}}[\boldsymbol{x}(T)]$$

The equation for $\boldsymbol{\lambda}$ is known as the *adjoint equation*. In addition, for all t, the optimal control u is such that

$$H[\boldsymbol{\lambda}(t), \boldsymbol{x}(t), \boldsymbol{v}] \leq H[\boldsymbol{\lambda}(t), \boldsymbol{x}(t), \boldsymbol{u}(t)]$$

where H is the Hamiltonian

$$H = \boldsymbol{\lambda}^T f(\boldsymbol{x}, \boldsymbol{u}) + \ell(\boldsymbol{x}, \boldsymbol{u})$$

Ex 4: Accelerating a Cart Consider the one-dimensional problem of accelerating a cart on a track, as shown in Figure 7. The problem is to pick a control law $u(t)$ that will get the cart as far as possible down the track at time T, but at the same time avoid costly accelerations.

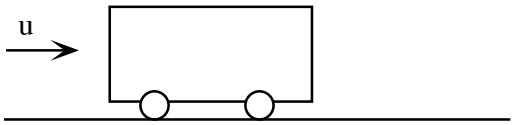

Figure 7: A cart on a one-dimensional track. The position on the track is given by $x(t)$. The cart is controlled by an acceleration $u(t)$.

The dynamic equation is

$$\ddot{x} = -\dot{x} + u(t)$$

with initial conditions

$$x(0) = 0$$
$$\dot{x}(0) = 0$$

The cost functional

$$J = x(T) - \frac{1}{2}\int_0^T u^2(t)dt$$

captures the desire to maximize the distance traveled in time T and at the same time penalize excessive accelerations.

Using the transformation of Section 5.2.1, the state variables x_1 and x_2 are defined by

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \end{pmatrix} = \begin{pmatrix} x_2 \\ -x_2 + u \end{pmatrix}$$
$$x_1(0) = x_2(0) = 0$$
$$J = x_1(T) - \frac{1}{2}\int_0^T u^2 dt$$

The Hamiltonian is given by

$$H = \lambda_1 x_2 - \lambda_2 x_2 + \lambda_2 u - \frac{1}{2}u^2$$

Differentiating this equation allows the determination of the adjoint system as

$$-\dot{\lambda}_1 = \frac{\partial H}{\partial x_1} = 0$$

$$-\dot{\lambda}_2 = \frac{\partial H}{\partial x_2} = \lambda_1 - \lambda_2$$

and its final condition can be determined from

$$\psi = x_1(T)$$

$$\boldsymbol{\lambda}(T) = \psi_{\boldsymbol{x}}(T) = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

The simple form of the adjoint equations allows their direct solution. For λ_1,

$$\lambda_1 = \text{ const } = 1$$

For λ_2,we could use Laplace transform methods, but they have not been discussed, so lets make the incredible lucky guess:

$$\lambda_2 = 1 - e^{t-T}$$

For a maximum differentiate H with respect to u,

$$\frac{\partial H}{\partial u} = 0 \Rightarrow \lambda_2 - u = 0$$

$$u = \lambda_2 = 1 - e^{t-T}$$

So the optimal control is to start off with an acceleration of magnitude $1-e^{-T}$ and then decrease it to 0 using the schedule presented here.

Linear System with Noise Up until now the state vector has been noiseless, but in any realistic case the state vector will be corrupted with noise. An especially elegant way of dealing with noise occurs when the system is linear. Suppose that the dynamic equation is given by

$$\dot{\boldsymbol{x}} = W\boldsymbol{x} + \boldsymbol{\mu}$$

where

$$\boldsymbol{x}(0) = \boldsymbol{x}_0$$

and where $\boldsymbol{\mu}$ is a noise vector with parameters

$$E[\boldsymbol{\mu}] = 0 \text{ and } E[\boldsymbol{\mu}\boldsymbol{\mu}^T] = Q$$

In other words, the state vector evolves from an initial condition and has noise added to it. The problem is to come up with an optimal estimate of the state that includes $\boldsymbol{x}$ and $\boldsymbol{\mu}$. How should we make this estimate? The state itself cannot be measured directly. Instead it is accessible through a measurement equation

$$\boldsymbol{z} = \boldsymbol{x} + \boldsymbol{\nu}$$

where $\boldsymbol{\nu}$ is a noise vector with parameters

$$E[\boldsymbol{\nu}] = 0 \text{ and } E[\boldsymbol{\nu}\boldsymbol{\nu}^T] = R$$

Thus one can only *estimate* the state. Lets call this estimate $\hat{\boldsymbol{x}}(t)$. The covariance of the error in the estimate is then

$$P(t) = E[(\boldsymbol{x}(t) - \hat{\boldsymbol{x}}(t))^T(\boldsymbol{x} - \hat{\boldsymbol{x}}(t))]$$

and we will asume $P(0)$ is known.

Given an initial estimate $\hat{\boldsymbol{x}}(0)$ we would like it to be close to the initial state, so $[\hat{\boldsymbol{x}}(0) - \boldsymbol{x}(0)]^T[\hat{\boldsymbol{x}}(0) - \boldsymbol{x}(0)]$ is a good measure. Similarly for the noise $\boldsymbol{\mu}$ and measurements $\boldsymbol{z}$ we can use error metrics $\boldsymbol{\mu}^T\boldsymbol{\mu}$ and $(\boldsymbol{z} - \boldsymbol{x})^T(\boldsymbol{z} - \boldsymbol{x})$, respectively. The only additional wrinkle is to weight these measurements by the inverse of the covariance matrix. This step has the effect of making the

term cost less when its component is noisy. Thus the objective function is given by

$$J = [\hat{\boldsymbol{x}}(0) - \boldsymbol{x}(0)]^T P_0^{-1} [\hat{\boldsymbol{x}}(0) - \boldsymbol{x}(0)] + \int_0^T [\boldsymbol{\mu}^T Q^{-1} \boldsymbol{\mu} + (\boldsymbol{z} - \boldsymbol{x})^T R^{-1} (\boldsymbol{z} - \boldsymbol{x})] dt$$

The solution to this problem can be obtained from the optimality conditions. Where the Hamiltonian H is given by

$$H = \boldsymbol{\lambda}^T (W\boldsymbol{x} + \boldsymbol{\mu}) + \boldsymbol{\mu}^T Q^{-1} \boldsymbol{\mu} + (\boldsymbol{z} - \boldsymbol{x})^T R^{-1} (\boldsymbol{z} - \boldsymbol{x})$$

differentiating with respect to $\boldsymbol{\lambda}$ results in the dynamic equation

$$\dot{\boldsymbol{x}} = W\boldsymbol{x} + \boldsymbol{\mu} \tag{4}$$

Differentiating with respect to $\boldsymbol{x}$ results in

$$-\dot{\boldsymbol{\lambda}} = R^{-1} (\boldsymbol{z} - \boldsymbol{x}) + W^T \boldsymbol{\lambda} \tag{5}$$

with final condition

$$\boldsymbol{\lambda}(T) = 0$$

And finally, differentiating with respect to $\boldsymbol{\mu}$ gives

$$\boldsymbol{\mu} = Q\boldsymbol{\lambda} \tag{6}$$

To solve this problem, postulate that the solution has the form

$$\boldsymbol{x} = \hat{\boldsymbol{x}} + P\boldsymbol{\lambda}$$

with initial condition

$$\boldsymbol{x}(0) = \hat{\boldsymbol{x}}(0) - P(0)\lambda(0)$$

Differentiating this equation with respect to time and using Equations 4, 5, and 6 results in

$$-\dot{\hat{\boldsymbol{x}}} + W\hat{\boldsymbol{x}} + PR^{-1}(\boldsymbol{z} - \hat{\boldsymbol{x}}) = [-\dot{P} + WP + PW^T + Q - PR^{-1}P]\boldsymbol{\lambda}$$

Since both sides of the equation are independent, they must be independently zero, that is,

$$\dot{\hat{\boldsymbol{x}}} = W\hat{\boldsymbol{x}} + PR^{-1}(\boldsymbol{z} - \hat{\boldsymbol{x}}) \text{ with } \hat{\boldsymbol{x}}(0) = \hat{\boldsymbol{x}}_0 \tag{7}$$

and

$$\dot{P} = WP + PW^T + Q - PR^{-1}P \text{ with } P(0) = P_0 \tag{8}$$

Equation 8 is known as the Ricatti equation.

Thus the optimal estimate of the state can be obtained by first solving the Ricatti equation to get $P(t)$ and then using it to solve Equation 7.