2 Optimal Control: Dynamics and Direct Methods

To develop a theory of active learning, we need to be able to translate data about the world into actions. Our focus here will be methods that focus on optimality criteria for doing so. Optimality criteria are not the only possible criteria one could use to drive actions—one could, for instance, simply define a proportional response to state, such as is done in PID control of linear systems—but optimality conceptually extends to many situations and systems that other techniques cannot treat.

There are several pieces involved in specifying a problem through optimality-based principles. These include a description of the dynamics, an objective function that represents the goal and distinguishes between good and bad outcomes. The ability to take derivatives of the dynamics and the objective will play a critical role in simplifying the procedural aspects of optimizing a function.

2.1 Dynamic Models

Typically a robotic system can be described by an ordinary differential equation.

$$\dot{x} = f(x,u) \quad x(0) = x_0 \tag{1}$$

In this equation we have the state $x \in \mathbb{R}^n$, the control $u \in \mathbb{R}^m$, and assume that f is at least differentiable with respect to both x and u.

Examples include:

1. $\dot{x} = u$, the single integrator system, where $x \in \mathbb{R}$ and $u \in \mathbb{R}$.
2. $\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u$, the double integrator system $\ddot{x} = u$.
3. $\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos(\theta)u_1 \\ \sin(\theta)u_1 \\ u_2 \end{bmatrix}$, the differential drive vehicle.

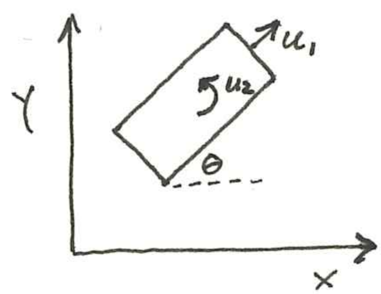

Figure 5: A vehicle capable of turning in place and moving forward

Note that sometimes variables like x will be used to denote vector or components of vectors. This will generally not lead to confusion, even when we say things like $x = \begin{bmatrix} x \\ y \end{bmatrix}$.

2.2 Objective functions

Objective functions generally come in a standardized form. We will see later that this standardized form is not always what we are looking for, but this provides a basis for thinking about optimization.

$$J(x(t),u(t)) = \int_0^T \ell(x(t),u(t))dt + m(x(T)) \tag{2}$$

Like the dynamic equations of motion, we will assume that $\ell(\cdot,\cdot)$ is at least differentiable with respect to $x(t)$ and $u(t)$.

Example of an objective function:

$$J(x(t),u(t)) = \int_0^T \frac{1}{2}(x(t)-x_d(t))^T Q(x(t)-x_d(t)) + \frac{1}{2}u(t)^T Ru(t)dt + \frac{1}{2}(x(T)-x_d(T))P_1(x(T)-x_d(T))$$

where $Q \geq 0$, $R > 0$, $P_1 \geq 0$. (That is, these are matrices and they are either positive semi-definite or positive definite, meaning that when their arguments are nonzero they either return a positive number or, in the case of semi-definiteness, can also return zero. For instance, we could have $Q = \begin{bmatrix} 1 & 0 \\ 0 & 0.1 \end{bmatrix}$, $R = [1]$, and $P_1 = \begin{bmatrix} 10^3 & 0 \\ 0 & 10 \end{bmatrix}$ for the double integrator dynamics above.

Our goal is to optimize J, only using *trajectories* of Eq.(1). This optimization over a curve $u(t)$ or the pair $(x(t), u(t))$ is an *infinite dimensional* optimization.

2.3 Taking Derivatives: Df and ∇f

There are two notions of derivative, the Frechét and Gateaux derivatives. They both assume that states and the values of objective functions are elements of *vector spaces*—sets where addition and scalar multiplication make sense. Moreover, differentiation outside of the scalar-valued case requires a norm to be well defined, as we see momentarily.

Definition 1. A mapping $f : X \to Y$ (X and Y both vector spaces) is (Frechét) differentiable at $x_0 \in X$ if there is a continuous linear mapping $Df(x_0) : X \to Y$ such that

$$\lim_{\|z\|_X \to 0} \frac{\|f(x_0 + z) - f(x_0) - Df(x_0) \cdot z\|_Y}{\|z\|_X} = 0.$$

Note that in this definition we have $x_0 + z, x_0 \in X$ and $f(\cdot)$ of these quantities are in Y.

To make sense of this notion of derivative, let's go back to what we learned in introductory calculus of one variable:

$$\frac{df}{dx}(x_0) = \lim_{h \to 0} \frac{f(x_0 + h) - f(x_0)}{h}.$$

Now, rearranging terms by moving the left hand side of the definition over to the right hand side, we get

$$0 = \lim_{h \to 0} \frac{f(x_0 + h) - f(x_0) - \frac{df}{dx}(x_0) \cdot h}{h}.$$

Why do we need the norms? Without the norms, we do not know what it means to "divide by" the denominator h. Even if we did, it would mean the definition would depend on the particular value h takes on. Instead, we require that the numerator and denominator be replaced by the norms of those elements of the vector space.

In both these definitions, $\frac{df}{dx}(x_0)$ is said to *approximate* f to first-order at x_0.

Definition 2. Another notion of derivative is

$$Df(x) \cdot h = \frac{df}{dx}(x) \cdot h = \frac{d}{d\varepsilon}|_{\varepsilon=0} f(x + \varepsilon z) \quad z \in X.$$

This is a weaker notion of derivative and is called the *Gateaux* derivative. However, this is the notion of derivative that we will use in practice because it is procedurally easy to work with. Any Frechét derivative that exists can be evaluated using the Gateaux derivative. However, a Gateaux derivative existing does not imply that the Frechét derivative exists.

EXAMPLE 1. Let's take the derivative using the Gateaux derivative. Assume that $f : \mathbb{R}^2 \to \mathbb{R}^2$

$$f\left(\begin{bmatrix} x \\ y \end{bmatrix}\right) = \begin{bmatrix} x^2 \\ x + y^2 \end{bmatrix}.$$

You can probably guess that

$$Df\left(\begin{bmatrix} x \\ y \end{bmatrix}\right) = \begin{bmatrix} 2x & 0 \\ 1 & 2y \end{bmatrix}.$$

However, let us find the derivative using the Gateaux derivative. Then $\frac{d}{d\varepsilon}|_{\varepsilon=0} f(x+\epsilon z)$ (where I am using x and z as vectors, purely for notational convenience) gives us

$$\frac{d}{d\varepsilon} f\left(\begin{bmatrix} x+\epsilon w \\ y+\epsilon z \end{bmatrix}\right)|_{\varepsilon=0} = \frac{d}{d\varepsilon}\begin{bmatrix} (x+\epsilon w)^2 \\ (x+\epsilon w)+(y+\epsilon z)^2 \end{bmatrix}|_{\varepsilon=0} = \begin{bmatrix} 2xw+2\epsilon w^2 \\ w+2yz+2\epsilon z^2 \end{bmatrix}|_{\varepsilon=0}$$
$$= \begin{bmatrix} 2xw \\ w+2yz \end{bmatrix} = \begin{bmatrix} 2x & 0 \\ 1 & 2y \end{bmatrix}\begin{bmatrix} w \\ z \end{bmatrix}$$

Or, equivalently, using vector notion we can obtain the same result.

$$\frac{d}{d\varepsilon} f\left(\begin{bmatrix} x \\ y \end{bmatrix} + \epsilon \begin{bmatrix} w \\ z \end{bmatrix}\right)|_{\varepsilon=0} = Df\left(\begin{bmatrix} x \\ y \end{bmatrix}\right)\cdot\begin{bmatrix} w \\ z \end{bmatrix}|_{\varepsilon=0} = \begin{bmatrix} 2x & 0 \\ 1 & 2y \end{bmatrix}\begin{bmatrix} w \\ z \end{bmatrix}$$

Moreover, we know that the derivative allows us to locally approximate the function.

$$f\left(\begin{bmatrix} x \\ y \end{bmatrix} + \epsilon \begin{bmatrix} w \\ z \end{bmatrix}\right) \approx f\left(\begin{bmatrix} x \\ y \end{bmatrix}\right) + \epsilon \begin{bmatrix} 2x & 0 \\ 1 & 2y \end{bmatrix}\begin{bmatrix} w \\ z \end{bmatrix}$$

EXAMPLE 2. Now, suppose that $J(x(t)) = \frac{1}{2}\int_0^1 x(t)^2 dt$, where $x(t) \in \mathbb{R}$. How do we differentiate J with respect to $x(t)$? Choose an arbitrary perturbation $z(t)$ and evaluate $\frac{d}{d\varepsilon}|_{\varepsilon=0} J(x(t)+\epsilon z(t))$.

$$\frac{d}{d\varepsilon} J(x(t)+\epsilon z(t))|_{\varepsilon=0} = \frac{1}{2}\int_0^1 \frac{d}{d\varepsilon}(x(t)+\epsilon z(t))^2 dt|_{\varepsilon=0} = \frac{1}{2}\int_0^1 (2x(t)z(t)+2\epsilon z(t)^2)dt|_{\varepsilon=0} = \int_0^1 x(t)z(t)dt$$

EXAMPLE 3. Suppose that $f(x,u)$ is the right hand side of the equations of motion for the differential drive car. What does the derivative of those equations look like? Hint:

$$Df(x,u)\circ(\delta x,\delta u) = D_1 f(x,u)\delta x + D_2 f(x,u)\delta u,$$

where $D_i f$ is the derivative of the function f with respect to its i^{th} component—called the *slot derivative.*

Note that the gradient $\nabla f(x)$ is defined by the inner product (i.e., the dot product) in the following way.

$$Df(x)\cdot z = \langle \nabla f(x), z\rangle$$

This *defines* the gradient, so for every inner product, one gets a new gradient. This is not something to get too worried about for our purposes, but it does help make more sense out of the fact that the gradient $\nabla f(x)$ is *added* to the vector x, implying they belong to the same vector space.

2.4 Direct Methods in Optimal Control

Assuming that both an objective function and the dynamics are differentiable, one common approach to computing an optimal $u(t)$ is to use constrained finite dimensional optimization to approximate the optimizer. There are very good—or at least pretty good—optimization tools out there for constrained finite dimensional optimization (e.g., MATLAB's fmincon(), SNOPT, and others). These tools assume a problem of the form

$$\min_x h(x) \text{ such that } g(x) = 0 \tag{3}$$

where both h and g are assumed to be differentiable with respect to x (and often times you need to provide the software with these derivatives). Moreover, depending on the method used, inequality constraints can often be imposed as well (e.g., $g(x) \geq 0$); these can be used to indicate input saturation or unsafe parts of the state.

How can we obtain such a finite dimensional description of the optimization problem? Our approach will be to discretize both ℓ and f using quadrature to obtain something in the form of Eq. (3). For instance, suppose that we use the definition of the derivative to discretize the continuous dynamics, splitting the time interval $[0, T]$ up into N different pieces.

$$\dot{x} = f(x, u) \implies \frac{x(t_{i+1}) - x(t_i)}{dt} \approx f(x(t_i), u(t_i))$$

which in turn implies the discrete time update law

$$x(t_{i+1}) = x(t_i) + dt f(x(t_i), u(t_i)).$$

Note that this equation holds for every time t_i, so there are N of these equations, all of which are constraints between x and u at various times. Also, note that $x(t_0) = x_0$.

Moreover, we can discretize J using Riemann integration approximation of the integral.

$$J = \int_0^T \ell(x(t), u(t))dt + m(x(T)) \approx \sum_{i=0}^{N} \ell(x(t_i), u(t_i))dt + m(x(t_N))$$

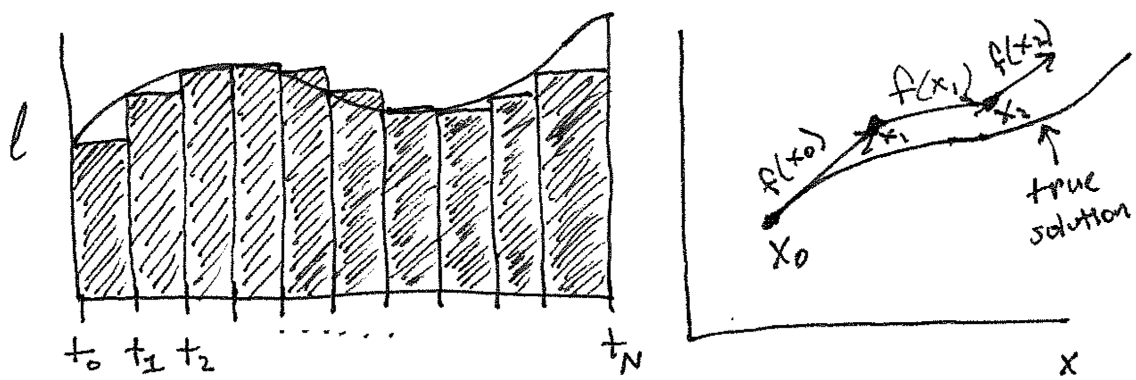

Figure 6: Both the objective function $\ell(\cdot)$ and dynamics can be discretized to create a finite dimensional optimization appropriate for direct methods. For instance, on the left is shown a Riemann sum approximation of the integral of ℓ, where the rectangles approximate the area under the curve. On the right, a trajectory is approximated with a discrete-time solver, such as Euler integration.

If we set

$$h(x(t_1) : x(t_N); u(t_0) : u(t_{N-1})) = \sum_{i=0}^{N} \ell(x(t_i), u(t_i))dt + m(x(T))$$

and set the constraints

$$g(x(t_1) : x(t_N); u(t_0) : u(t_{N-1})) = x(t_{i+1}) - x(t_i) - dt f(x(t_i), u(t_i)) \; \forall i$$

then we have a problem of the needed form.

A few notes on direct methods

First, *any* quadrature/interpolation scheme can be used to generate a finite dimensional description. The one I use above is Euler integration and Riemann integration of integrals, but one could use any Runge-Kutta/implicit Euler/midpoint/etc scheme for the dynamics; one could use any integration (e.g., trapezoidal rule, Simpson's rule, et cetera) for the objective function. Moreover, higher-order representations of both will be better for numerical purposes.

Second, note that one could impose the terminal goal at time T as a constraint. For instance, one could say that $x(t_N) = x_d(T)$ in the list of constraints. If one gets a solution, this can be great. However, problems can become ill-conditioned, particularly if $x_d(T)$ is somehow infeasible.

Exercise

Take the double integrator system, and using the Q, R, P_1 from above, implement (using fmincon or something similar) a direct optimization with $N = 10, 10^2, 10^3$. Run each solution through a continuous simulation of the system. How well do the optimized controls work? What happens if you change the constraints to implicit Euler or the midpoint rule?

Summing Up

The use of optimization as a model of how to go from environmental estimates to control actions has a long history of being very successful. We will want to find objective functions appropriate for active learning. We should expect that we will want them to be differentiable, based on what we have discussed here.