

Optimal control

Optimal control theory deals with the problem of finding a control law for a given system such that a certain optimality criterion is achieved.

It is an extension of the calculus of variations, and is a mathematical optimization method for deriving control policies. The method is largely due to the work of Lev Pontryagin and Richard Bellman in the 1950s, after contributions to calculus of variations by Edward J. McShane.^[1] Optimal control can be seen as a control strategy in control theory.

Contents

- 1** **General method**
- 2** **Linear quadratic control**
- 3** **Numerical methods for optimal control**
- 4** **Discrete-time optimal control**
- 5** **Examples**
 - 5.1 **Finite time**
- 6** **See also**
- 7** **References**
- 8** **Further reading**
- 9** **External links**

General method

Optimal control deals with the problem of finding a control law for a given system such that a certain optimality criterion is achieved. A control problem includes a cost functional that is a function of state and control variables. An **optimal control** is a set of differential equations describing the paths of the control variables that minimize the cost function. The optimal control can be derived using Pontryagin's maximum principle (a necessary condition also known as Pontryagin's minimum principle or simply Pontryagin's Principle),^[2] or by solving the Hamilton–Jacobi–Bellman equation (a sufficient condition).

We begin with a simple example. Consider a car traveling in a straight line on a hilly road. The question is, how should the driver press the accelerator pedal in order to *minimize* the total traveling time? Clearly in this example, the term *control law* refers specifically to the way in which the driver presses the accelerator and shifts the gears. The *system* consists of both the car and the road, and the *optimality criterion* is the minimization of the total traveling time. Control problems usually include ancillary constraints. For example, the amount of available fuel might be limited, the accelerator pedal cannot be pushed through the floor of the car, speed limits, etc.

This approach may be applied in cosmology and astrophysics,^[3] A proper cost function will be a mathematical expression giving the traveling time as a function of the speed, geometrical considerations, and initial conditions of the system. It is often the case that the constraints are interchangeable with the cost function.

Another optimal control problem is to find the way to drive the car so as to minimize its fuel consumption, given that it must complete a given course in a time not exceeding some amount. Yet another control problem is to minimize the total monetary cost of completing the trip, given assumed monetary prices for time and fuel.

A more abstract framework goes as follows. Minimize the continuous-time cost functional

$$J = \Phi [\mathbf{x}(t_0), t_0, \mathbf{x}(t_f), t_f] + \int_{t_0}^{t_f} \mathcal{L} [\mathbf{x}(t), \mathbf{u}(t), t] dt$$

subject to the first-order dynamic constraints (the **state equation**)

$$\dot{\mathbf{x}}(t) = \mathbf{a} [\mathbf{x}(t), \mathbf{u}(t), t],$$

the algebraic *path constraints*

$$\mathbf{b} [\mathbf{x}(t), \mathbf{u}(t), t] \leq \mathbf{0},$$

and the boundary conditions

$$\phi [\mathbf{x}(t_0), t_0, \mathbf{x}(t_f), t_f] = \mathbf{0}$$

where $\mathbf{x}(t)$ is the *state*, $\mathbf{u}(t)$ is the *control*, t is the independent variable (generally speaking, time), t_0 is the initial time, and t_f is the terminal time. The terms Φ and \mathcal{L} are called the *endpoint cost* and *Lagrangian*, respectively. Furthermore, it is noted that the path constraints are in general *inequality* constraints and thus may not be active (i.e., equal to zero) at the optimal solution. It is also noted that the optimal control problem as stated above may have multiple solutions (i.e., the solution may not be unique). Thus, it is most often the case that any solution $[\mathbf{x}^*(t^*), \mathbf{u}^*(t^*), t^*]$ to the optimal control problem is *locally minimizing*.

Linear quadratic control

A special case of the general nonlinear optimal control problem given in the previous section is the *linear quadratic* (LQ) optimal control problem. The LQ problem is stated as follows. Minimize the *quadratic* continuous-time cost functional

$$J = \frac{1}{2} \mathbf{x}^\top(t_f) \mathbf{S}_f \mathbf{x}(t_f) + \frac{1}{2} \int_{t_0}^{t_f} [\mathbf{x}^\top(t) \mathbf{Q}(t) \mathbf{x}(t) + \mathbf{u}^\top(t) \mathbf{R}(t) \mathbf{u}(t)] dt$$

Subject to the *linear* first-order dynamic constraints

$$\dot{\mathbf{x}}(t) = \mathbf{A}(t) \mathbf{x}(t) + \mathbf{B}(t) \mathbf{u}(t),$$

and the initial condition

$$\mathbf{x}(t_0) = \mathbf{x}_0$$

A particular form of the LQ problem that arises in many control system problems is that of the *linear quadratic regulator* (LQR) where all of the matrices (i.e., \mathbf{A} , \mathbf{B} , \mathbf{Q} , and \mathbf{R}) are *constant*, the initial time is arbitrarily set to zero, and the terminal time is taken in the limit $t_f \rightarrow \infty$ (this last assumption is what is known as *infinite horizon*). The LQR problem is stated as follows. Minimize the infinite horizon quadratic continuous-time cost functional

$$J = \frac{1}{2} \int_0^{\infty} [\mathbf{x}^T(t) \mathbf{Q} \mathbf{x}(t) + \mathbf{u}^T(t) \mathbf{R} \mathbf{u}(t)] dt$$

Subject to the *linear time-invariant* first-order dynamic constraints

$$\dot{\mathbf{x}}(t) = \mathbf{A} \mathbf{x}(t) + \mathbf{B} \mathbf{u}(t),$$

and the initial condition

$$\mathbf{x}(t_0) = \mathbf{x}_0$$

In the finite-horizon case the matrices are restricted in that \mathbf{Q} and \mathbf{R} are positive semi-definite and positive definite, respectively. In the infinite-horizon case, however, the matrices \mathbf{Q} and \mathbf{R} are not only positive-semidefinite and positive-definite, respectively, but are also *constant*. These additional restrictions on \mathbf{Q} and \mathbf{R} in the infinite-horizon case are enforced to ensure that the cost functional remains positive. Furthermore, in order to ensure that the cost function is *bounded*, the additional restriction is imposed that the pair (\mathbf{A}, \mathbf{B}) is *controllable*. Note that the LQ or LQR cost functional can be thought of physically as attempting to minimize the *control energy* (measured as a quadratic form).

The infinite horizon problem (i.e., LQR) may seem overly restrictive and essentially useless because it assumes that the operator is driving the system to zero-state and hence driving the output of the system to zero. This is indeed correct. However the problem of driving the output to a desired nonzero level can be solved *after* the zero output one is. In fact, it can be proved that this secondary LQR problem can be solved in a very straightforward manner. It has been shown in classical optimal control theory that the LQ (or LQR) optimal control has the feedback form

$$\mathbf{u}(t) = -\mathbf{K}(t) \mathbf{x}(t)$$

where $\mathbf{K}(t)$ is a properly dimensioned matrix, given as

$$\mathbf{K}(t) = \mathbf{R}^{-1} \mathbf{B}^T \mathbf{S}(t),$$

and $\mathbf{S}(t)$ is the solution of the differential Riccati equation. The differential Riccati equation is given as

$$\dot{\mathbf{S}}(t) = -\mathbf{S}(t) \mathbf{A} - \mathbf{A}^T \mathbf{S}(t) + \mathbf{S}(t) \mathbf{B} \mathbf{R}^{-1} \mathbf{B}^T \mathbf{S}(t) - \mathbf{Q}$$

For the finite horizon LQ problem, the Riccati equation is integrated backward in time using the terminal boundary condition

$$\mathbf{S}(t_f) = \mathbf{S}_f$$

For the infinite horizon LQR problem, the differential Riccati equation is replaced with the *algebraic* Riccati equation (ARE) given as

$$\mathbf{0} = -\mathbf{S} \mathbf{A} - \mathbf{A}^T \mathbf{S} + \mathbf{S} \mathbf{B} \mathbf{R}^{-1} \mathbf{B}^T \mathbf{S} - \mathbf{Q}$$

Understanding that the ARE arises from infinite horizon problem, the matrices \mathbf{A} , \mathbf{B} , \mathbf{Q} , and \mathbf{R} are all *constant*. It is noted that there are in general multiple solutions to the algebraic Riccati equation and the *positive definite* (or positive semi-definite) solution is the one that is used to compute the feedback gain. The LQ (LQR) problem was elegantly solved by Rudolf Kalman.^[4]

Numerical methods for optimal control

Optimal control problems are generally nonlinear and therefore, generally do not have analytic solutions (e.g., like the linear-quadratic optimal control problem). As a result, it is necessary to employ numerical methods to solve optimal control problems. In the early years of optimal control (c. 1950s to 1980s) the favored approach for solving optimal control problems was that of *indirect methods*. In an indirect method, the calculus of variations is employed to obtain the first-order optimality conditions. These conditions result in a two-point (or, in the case of a complex problem, a multi-point) boundary-value problem. This boundary-value problem actually has a special structure because it arises from taking the derivative of a Hamiltonian. Thus, the resulting dynamical system is a Hamiltonian system of the form

$$\begin{aligned}\dot{\mathbf{x}} &= \partial H / \partial \boldsymbol{\lambda} \\ \dot{\boldsymbol{\lambda}} &= -\partial H / \partial \mathbf{x}\end{aligned}$$

where

$$H = \mathcal{L} + \boldsymbol{\lambda}^\top \mathbf{a} - \boldsymbol{\mu}^\top \mathbf{b}$$

is the *augmented Hamiltonian* and in an indirect method, the boundary-value problem is solved (using the appropriate boundary or *transversality* conditions). The beauty of using an indirect method is that the state and adjoint (i.e., $\boldsymbol{\lambda}$) are solved for and the resulting solution is readily verified to be an extremal trajectory. The disadvantage of indirect methods is that the boundary-value problem is often extremely difficult to solve (particularly for problems that span large time intervals or problems with interior point constraints). A well-known software program that implements indirect methods is BNDSCO.^[5]

The approach that has risen to prominence in numerical optimal control over the past two decades (i.e., from the 1980s to the present) is that of so-called *direct methods*. In a direct method, the state and/or control are approximated using an appropriate function approximation (e.g., polynomial approximation or piecewise constant parameterization). Simultaneously, the cost functional is approximated as a *cost function*. Then, the coefficients of the function approximations are treated as optimization variables and the problem is "transcribed" to a nonlinear optimization problem of the form:

Minimize

$$F(\mathbf{z})$$

subject to the algebraic constraints

$$\begin{aligned}\mathbf{g}(\mathbf{z}) &= \mathbf{0} \\ \mathbf{h}(\mathbf{z}) &\leq \mathbf{0}\end{aligned}$$

Depending upon the type of direct method employed, the size of the nonlinear optimization problem can be quite small (e.g., as in a direct shooting or quasilinearization method), moderate (e.g. pseudospectral optimal control^[6]) or may be quite large (e.g., a direct collocation method^[7]). In the latter case (i.e., a collocation method), the nonlinear optimization problem may be literally thousands to tens of thousands of variables and constraints.

Given the size of many NLPs arising from a direct method, it may appear somewhat counter-intuitive that solving the nonlinear optimization problem is easier than solving the boundary-value problem. It is, however, the fact that the NLP is easier to solve than the boundary-value problem. The reason for the relative ease of computation, particularly of a direct collocation method, is that the NLP is *sparse* and many well-known software programs exist (e.g., SNOPT^[8]) to solve large sparse NLPs. As a result, the range of problems that can be solved via direct methods (particularly direct *collocation methods* which are very popular these days) is significantly larger than the range of problems that can be solved via indirect methods. In fact, direct methods have become so popular these days that many people have written elaborate software programs that employ these methods. In particular, many such programs include *DIRCOL*,^[9] *SOCS*,^[10] *OTIS*,^[11] *GESOP/ASTOS*,^[12] *DITAN*.^[13] and *PyGMO/PyKEP*.^[14] In recent years, due to the advent of the MATLAB programming language, optimal control software in MATLAB has become more common. Examples of academically developed MATLAB software tools implementing direct methods include *RIOTS*,^[15] *DIDO*,^[16] *DIRECT*,^[17] and *GPOPS*,^[18] while an example of an industry developed MATLAB tool is *PROPT*.^[19] These software tools have increased significantly the opportunity for people to explore complex optimal control problems both for academic research and industrial problems. Finally, it is noted that general-purpose MATLAB optimization environments such as TOMLAB have made coding complex optimal control problems significantly easier than was previously possible in languages such as C and FORTRAN.

Discrete-time optimal control

The examples thus far have shown continuous time systems and control solutions. In fact, as optimal control solutions are now often implemented digitally, contemporary control theory is now primarily concerned with discrete time systems and solutions. The Theory of Consistent Approximations^[20] provides conditions under which solutions to a series of increasingly accurate discretized optimal control problem converge to the solution of the original, continuous-time problem. Not all discretization methods have this property, even seemingly obvious ones. For instance, using a variable step-size routine to integrate the problem's dynamic equations may generate a gradient which does not converge to zero (or point in the right direction) as the solution is approached. The direct method *RIOTS* (<http://www.schwartz-home.com/RIOTS>) is based on the Theory of Consistent Approximation.

Examples

A common solution strategy in many optimal control problems is to solve for the costate (sometimes called the shadow price) $\lambda(t)$. The costate summarizes in one number the marginal value of expanding or contracting the state variable next turn. The marginal value is not only the gains accruing to it next turn but associated with the duration of the program. It is nice when $\lambda(t)$ can be solved analytically, but usually the most one can do is describe it sufficiently well that the intuition can grasp the character of the solution and an equation solver can solve numerically for the values.

Having obtained $\lambda(t)$, the turn- t optimal value for the control can usually be solved as a differential equation conditional on knowledge of $\lambda(t)$. Again it is infrequent, especially in continuous-time problems, that one obtains the value of the control or the state explicitly. Usually the strategy is to solve for thresholds and regions that characterize the optimal control and use a numerical solver to isolate the actual choice values in time.

Finite time

Consider the problem of a mine owner who must decide at what rate to extract ore from his mine. He owns rights to the ore from date 0 to date T . At date 0 there is x_0 ore in the ground, and the instantaneous stock of ore $x(t)$ declines at the rate the mine owner extracts it $u(t)$. The mine owner extracts ore at cost $u(t)^2/x(t)$ and sells ore

at a constant price p . He does not value the ore remaining in the ground at time T (there is no "scrap value"). He chooses the rate of extraction in time $u(t)$ to maximize profits over the period of ownership with no time discounting.

1. Discrete-time version

The manager maximizes profit Π :

$$\Pi = \sum_{t=0}^{T-1} \left[pu_t - \frac{u_t^2}{x_t} \right]$$

subject to the law of evolution for the state variable x_t

$$x_{t+1} - x_t = -u_t$$

Form the Hamiltonian and differentiate:

$$H = pu_t - \frac{u_t^2}{x_t} - \lambda_{t+1}u_t$$

$$\frac{\partial H}{\partial u_t} = p - \lambda_{t+1} - 2\frac{u_t}{x_t} = 0$$

$$\lambda_{t+1} - \lambda_t = -\frac{\partial H}{\partial x_t} = -\left(\frac{u_t}{x_t}\right)^2$$

As the mine owner does not value the ore remaining at time T ,

$$\lambda_T = 0$$

Using the above equations, it is easy to solve for the x_t and λ_t series

$$\lambda_t = \lambda_{t+1} + \frac{(p - \lambda_{t+1})^2}{4}$$

$$x_{t+1} = x_t \frac{2 - p + \lambda_{t+1}}{2}$$

and using the initial and turn-T conditions, the x_t series can be solved explicitly, giving u_t .

2. Continuous-time version

The manager maximizes profit Π :

$$\Pi = \int_0^T \left[pu(t) - \frac{u(t)^2}{x(t)} \right] dt$$

subject to the law of evolution for the state variable $x(t)$

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

Form the Hamiltonian and differentiate:

$$H = pu(t) - \frac{u(t)^2}{x(t)} - \lambda(t)u(t)$$

$$\frac{\partial H}{\partial u} = p - \lambda(t) - 2\frac{u(t)}{x(t)} = 0$$

$$\dot{\lambda}(t) = -\frac{\partial H}{\partial x} = -\left(\frac{u(t)}{x(t)}\right)^2$$

As the mine owner does not value the ore remaining at time T ,

$$\lambda(T) = 0$$

Using the above equations, it is easy to solve for the differential equations governing $u(t)$ and $\lambda(t)$

$$\dot{\lambda}(t) = -\frac{(p - \lambda(t))^2}{4}$$

$$u(t) = x(t) \frac{p - \lambda(t)}{2}$$

and using the initial and turn-T conditions, the functions can be solved to yield

$$x(t) = \frac{(4 - pt + pT)^2}{(4 + pT)^2} x_0$$

See also