To appear in *Bayesian Brain*, Doya, K. (ed), MIT Press (2006)

Optimal Control Theory

Emanuel Todorov
University of California San Diego

Optimal control theory is a mature mathematical discipline with numerous applications in both science and engineering. It is emerging as the computational framework of choice for studying the neural control of movement, in much the same way that probabilistic inference is emerging as the computational framework of choice for studying sensory information processing. Despite the growing popularity of optimal control models, however, the elaborate mathematical machinery behind them is rarely exposed and the big picture is hard to grasp without reading a few technical books on the subject. While this chapter cannot replace such books, it aims to provide a self-contained mathematical introduction to optimal control theory that is sufficiently broad and yet sufficiently detailed when it comes to key concepts. The text is not tailored to the field of motor control (apart from the last section, and the overall emphasis on systems with continuous state) so it will hopefully be of interest to a wider audience. Of special interest in the context of this book is the material on the duality of optimal control and probabilistic inference; such duality suggests that neural information processing in sensory and motor areas may be more similar than currently thought. The chapter is organized in the following sections:

1. Dynamic programming, Bellman equations, optimal value functions, value and policy iteration, shortest paths, Markov decision processes.

2. Hamilton-Jacobi-Bellman equations, approximation methods, finite and infinite horizon formulations, basics of stochastic calculus.

3. Pontryagin's maximum principle, ODE and gradient descent methods, relationship to classical mechanics.

4. Linear-quadratic-Gaussian control, Riccati equations, iterative linear approximations to nonlinear problems.

5. Optimal recursive estimation, Kalman filter, Zakai equation.

6. Duality of optimal control and optimal estimation (including new results).

7. Optimality models in motor control, promising research directions.

1 Discrete control: Bellman equations

Let $x \in \mathcal{X}$ denote the state of an agent's environment, and $u \in \mathcal{U}(x)$ the action (or control) which the agent chooses while at state x. For now both $\mathcal{X}$ and $\mathcal{U}(x)$ are finite sets. Let $\mathit{next}(x,u) \in \mathcal{X}$ denote the state which results from applying action u in state x, and $\mathit{cost}(x,u) \geq 0$ the cost of applying action u in state x. As an example, x may be the city where we are now, u the flight we choose to take, $\mathit{next}(x,u)$ the city where that flight lands, and $\mathit{cost}(x,u)$ the price of the ticket. We can now pose a simple yet practical optimal control problem: find the cheapest way to fly to your destination. This problem can be formalized as follows: find an action sequence $(u_0, u_1, \cdots u_{n-1})$ and corresponding state sequence $(x_0, x_1, \cdots x_n)$ minimizing the total cost

$$J(x_\cdot, u_\cdot) = \sum\nolimits_{k=0}^{n-1} \mathit{cost}(x_k, u_k)$$

where $x_{k+1} = \mathit{next}(x_k, u_k)$ and $u_k \in \mathcal{U}(x_k)$. The initial state $x_0 = x^{\text{init}}$ and destination state $x_n = x^{\text{dest}}$ are given. We can visualize this setting with a directed graph where the states are nodes and the actions are arrows connecting the nodes. If $\mathit{cost}(x,u) = 1$ for all (x,u) the problem reduces to finding the shortest path from x^{init} to x^{dest} in the graph.

1.1 Dynamic programming

Optimization problems such as the one stated above are efficiently solved via *dynamic programming* (DP). DP relies on the following obvious fact: if a given state-action sequence is optimal, and we were to remove the first state and action, the remaining sequence is also optimal (with the second state of the original sequence now acting as initial state). This is the *Bellman optimality principle.* Note the close resemblance to the Markov property of stochastic processes (a process is Markov if its future is conditionally independent of the past given the present state). The optimality principle can be reworded in similar language: the choice of optimal actions in the future is independent of the past actions which led to the present state. Thus optimal state-action sequences can be constructed by starting at the final state and extending backwards. Key to this procedure is the *optimal value function* (or optimal cost-to-go function)

$$v(x) = \text{"minimal total cost for completing the task starting from state } x\text{"}$$

This function captures the long-term cost for starting from a given state, and makes it possible to find optimal actions through the following algorithm:

Consider every action available at the current state,
add its immediate cost to the optimal value of the resulting next state,
and choose an action for which the sum is minimal.

The above algorithm is "greedy" in the sense that actions are chosen based on local information, without explicit consideration of all future scenarios. And yet the resulting actions are optimal. This is possible because the optimal value function contains all information about future scenarios that is relevant to the present choice of action. Thus the optimal value function is an extremely useful quantity, and indeed its calculation is at the heart of many methods for optimal control.

The above algorithm yields an optimal action $u = \pi(x) \in \mathcal{U}(x)$ for every state x. A mapping from states to actions is called *control law* or control policy. Once we have a control law $\pi : \mathcal{X} \rightarrow \mathcal{U}(\mathcal{X})$ we can start at any state x_0, generate action $u_0 = \pi(x_0)$, transition to state $x_1 = next(x_0, u_0)$, generate action $u_1 = \pi(x_1)$, and keep going until we reach x^{dest}.

Formally, an optimal control law π satisfies

$$\pi(x) = \arg\min_{u \in \mathcal{U}(x)} \{cost(x,u) + v(next(x,u))\} \tag{1}$$

The minimum in (1) may be achieved for multiple actions in the set $\mathcal{U}(x)$, which is why π may not be unique. However the optimal value function v is always uniquely defined, and satisfies

$$v(x) = \min_{u \in \mathcal{U}(x)} \{cost(x,u) + v(next(x,u))\} \tag{2}$$

Equations (1) and (2) are the *Bellman equations*.

If for some x we already know $v(next(x,u))$ for all $u \in \mathcal{U}(x)$, then we can apply the Bellman equations directly and compute $\pi(x)$ and $v(x)$. Thus dynamic programming is particularly simple in acyclic graphs where we can start from x^{dest} with $v\left(x^{\text{dest}}\right) = 0$, and perform a backward pass in which every state is visited after all its successor states have been visited. It is straightforward to extend the algorithm to the case where we are given non-zero final costs for a number of destination states (or absorbing states).

1.2 Value iteration and policy iteration

The situation is more complex in graphs with cycles. Here the Bellman equations are still valid, but we cannot apply them in a single pass. This is because the presence of cycles makes it impossible to visit each state only after all its successors have been visited. Instead the Bellman equations are treated as consistency conditions and used to design iterative relaxation schemes – much like partial differential equations (PDEs) are treated as consistency conditions and solved with corresponding relaxation schemes. By "relaxation scheme" we mean guessing the solution, and iteratively improving the guess so as to make it more compatible with the consistency condition.

The two main relaxation schemes are *value iteration* and *policy iteration*. Value iteration uses only (2). We start with a guess $v^{(0)}$ of the optimal value function, and construct a sequence of improved guesses:

$$v^{(i+1)}(x) = \min_{u \in \mathcal{U}(x)} \left\{cost(x,u) + v^{(i)}(next(x,u))\right\} \tag{3}$$

This process is guaranteed to converge to the optimal value function v in a finite number of iterations. The proof relies on the important idea of contraction mappings: one defines the approximation error $e\left(v^{(i)}\right) = \max_x \left|v^{(i)}(x) - v(x)\right|$, and shows that the iteration (3) causes $e\left(v^{(i)}\right)$ to decrease as i increases. In other words, the mapping $v^{(i)} \rightarrow v^{(i+1)}$ given by (3) contracts the "size" of $v^{(i)}$ as measured by the error norm $e\left(v^{(i)}\right)$.

Policy iteration uses both (1) and (2). It starts with a guess $\pi^{(0)}$ of the optimal control law, and constructs a sequence of improved guesses:

$$\begin{aligned} v^{\pi^{(i)}}(x) &= \mathit{cost}\left(x, \pi^{(i)}(x)\right) + v^{\pi^{(i)}}\left(\mathit{next}\left(x, \pi^{(i)}(x)\right)\right) \\ \pi^{(i+1)}(x) &= \arg\min_{u \in \mathcal{U}(x)} \left\{ \mathit{cost}(x, u) + v^{\pi^{(i)}}(\mathit{next}(x, u)) \right\} \end{aligned} \tag{4}$$

The first line of (4) requires a separate relaxation to compute the value function $v^{\pi^{(i)}}$ for the control law $\pi^{(i)}$. This function is defined as the total cost for starting at state x and acting according to $\pi^{(i)}$ thereafter. Policy iteration can also be proven to converge in a finite number of iterations. It is not obvious which algorithm is better, because each of the two nested relaxations in policy iteration converges faster than the single relaxation in value iteration. In practice both algorithms are used depending on the problem at hand.

1.3 Markov decision processes

The problems considered thus far are deterministic, in the sense that applying action u at state x always yields the same next state $\mathit{next}(x, u)$. Dynamic programming easily generalizes to the stochastic case where we have a probability distribution over possible next states:

$$p(y|x, u) = \text{"probability that } \mathit{next}(x, u) = y\text{"}$$

In order to qualify as a probability distribution the function p must satisfy

$$\begin{aligned} \textstyle\sum_{y \in \mathcal{X}} p(y|x, u) &= 1 \\ p(y|x, u) &\geq 0 \end{aligned}$$

In the stochastic case the value function equation (2) becomes

$$v(x) = \min_{u \in \mathcal{U}(x)} \left\{ \mathit{cost}(x, u) + E\left[v(\mathit{next}(x, u))\right] \right\} \tag{5}$$

where E denotes expectation over $\mathit{next}(x, u)$, and is computed as

$$E\left[v(\mathit{next}(x, u))\right] = \textstyle\sum_{y \in \mathcal{X}} p(y|x, u)\, v(y)$$

Equations (1, 3, 4) generalize to the stochastic case in the same way as equation (2) does.

An optimal control problem with discrete states and actions and probabilistic state transitions is called a *Markov decision process* (MDP). MDPs are extensively studied in reinforcement learning – which is a sub-field of machine learning focusing on optimal control problems with discrete state. In contrast, optimal control theory focuses on problems with continuous state and exploits their rich differential structure.

2 Continuous control: Hamilton-Jacobi-Bellman equations

We now turn to optimal control problems where the state $\mathbf{x} \in \mathbb{R}^{n_x}$ and control $\mathbf{u} \in \mathcal{U}(\mathbf{x}) \subseteq \mathbb{R}^{n_u}$ are real-valued vectors. To simplify notation we will use the shortcut $\min_u$ instead of

$\min_{u \in \mathcal{U}(\mathbf{x})}$, although the latter is implied unless noted otherwise. Consider the stochastic differential equation

$$d\mathbf{x} = \mathbf{f}(\mathbf{x}, \mathbf{u})\, dt + F(\mathbf{x}, \mathbf{u})\, d\mathbf{w} \tag{6}$$

where $d\mathbf{w}$ is n_w-dimensional Brownian motion. This is sometimes called a *controlled Ito diffusion*, with $\mathbf{f}(\mathbf{x}, \mathbf{u})$ being the drift and $F(\mathbf{x}, \mathbf{u})$ the diffusion coefficient. In the absence of noise, i.e. when $F(\mathbf{x}, \mathbf{u}) = 0$, we can simply write $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u})$. However in the stochastic case this would be meaningless because the sample paths of Brownian motion are not differentiable (the term $d\mathbf{w}/dt$ is infinite). What equation (6) really means is that the integral of the left hand side is equal to the integral of the right hand side:

$$\mathbf{x}(t) = \mathbf{x}(0) + \int_0^t \mathbf{f}(\mathbf{x}(s), \mathbf{u}(s))\, ds + \int_0^t F(\mathbf{x}(s), \mathbf{u}(s))\, d\mathbf{w}(s)$$

The last term is an Ito integral, defined for square-integrable functions $g(t)$ as

$$\int_0^t g(s)\, dw(s) = \lim_{n \to \infty} \sum\nolimits_{k=0}^{n-1} g(s_k)\left(w(s_{k+1}) - w(s_k)\right)$$
$$\text{where } 0 = s_0 < s_2 < \cdots < s_n = t$$

We will stay away from the complexities of stochastic calculus to the extent possible. Instead we will discretize the time axis and obtain results for the continuous-time case in the limit of infinitely small time step.

The appropriate Euler discretization of (6) is

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \Delta \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k) + \sqrt{\Delta} F(\mathbf{x}_k, \mathbf{u}_k)\, \varepsilon_k$$

where Δ is the time step, $\varepsilon_k \sim \mathcal{N}(0, \mathrm{I}^{n_w})$ and $\mathbf{x}_k = \mathbf{x}(k\Delta)$. The $\sqrt{\Delta}$ term appears because the variance of Brownian motion grows linearly with time, and thus the standard deviation of the discrete-time noise should scale as $\sqrt{\Delta}$.

To define an optimal control problem we also need a cost function. In finite-horizon problems, i.e. when a final time t_f is specified, it is natural to separate the total cost into a time-integral of a *cost rate* $\ell(\mathbf{x}, \mathbf{u}, t) \geq 0$, and a *final cost* $h(\mathbf{x}) \geq 0$ which is only evaluated at the final state $\mathbf{x}(t_f)$. Thus the total cost for a given state-control trajectory $\{\mathbf{x}(t), \mathbf{u}(t) : 0 \leq t \leq t_f\}$ is defined as

$$J(\mathbf{x}(\cdot), \mathbf{u}(\cdot)) = h(\mathbf{x}(t_f)) + \int_0^{t_f} \ell(\mathbf{x}(t), \mathbf{u}(t), t)\, dt$$

Keep in mind that we are dealing with a stochastic system. Our objective is to find a control law $\mathbf{u} = \pi(\mathbf{x}, t)$ which minimizes the *expected* total cost for starting at a given $(\mathbf{x}, t)$ and acting according π thereafter.

In discrete time the total cost becomes

$$J(\mathbf{x}_\cdot, \mathbf{u}_\cdot) = h(\mathbf{x}_n) + \Delta \sum\nolimits_{k=0}^{n-1} \ell(\mathbf{x}_k, \mathbf{u}_k, k\Delta)$$

where $n = t_f/\Delta$ is the number of time steps (assume that t_f/Δ is integer).

2.1 Derivation of the HJB equations

We are now ready to apply dynamic programming to the time-discretized stochastic problem. The development is similar to the MDP case except that the state space is now infinite: it consists of $n+1$ copies of $\mathbb{R}^{n_x}$. The reason we need multiple copies of $\mathbb{R}^{n_x}$ is that we have a finite-horizon problem, and therefore the time when a given $\mathbf{x} \in \mathbb{R}^{n_x}$ is reached makes a difference.

The state transitions are now stochastic: the probability distribution of $\mathbf{x}_{k+1}$ given $\mathbf{x}_k, \mathbf{u}_k$ is the multivariate Gaussian

$$\begin{gathered}\mathbf{x}_{k+1} \sim \mathcal{N}\left(\mathbf{x}_k + \Delta \mathbf{f}\left(\mathbf{x}_k, \mathbf{u}_k\right),\ \Delta S\left(\mathbf{x}_k, \mathbf{u}_k\right)\right) \\ \text{where } S\left(\mathbf{x}, \mathbf{u}\right) = F\left(\mathbf{x}, \mathbf{u}\right) F\left(\mathbf{x}, \mathbf{u}\right)^{\mathsf{T}}\end{gathered}$$

The Bellman equation for the optimal value function v is similar to (5), except that v is now a function of space and time. We have

$$\begin{aligned}& v\left(\mathbf{x}, k\right) = \min_{\mathbf{u}} \left\{\Delta \ell\left(\mathbf{x}, \mathbf{u}, k\Delta\right) + E\left[v\left(\mathbf{x} + \Delta \mathbf{f}\left(\mathbf{x}, \mathbf{u}\right) + \xi,\ k+1\right)\right]\right\} \qquad (7) \\ & \text{where } \xi \sim \mathcal{N}\left(0,\ \Delta S\left(\mathbf{x}, \mathbf{u}\right)\right) \quad \text{and } v\left(\mathbf{x}, n\right) = h\left(\mathbf{x}\right)\end{aligned}$$

Consider the second-order Taylor-series expansion of v, with the time index $k+1$ suppressed for clarity:

$$\begin{gathered}v\left(\mathbf{x} + \delta\right) = v\left(\mathbf{x}\right) + \delta^{\mathsf{T}} v_{\mathbf{x}}\left(\mathbf{x}\right) + \tfrac{1}{2}\delta^{\mathsf{T}} v_{\mathbf{xx}}\left(\mathbf{x}\right)\delta + o\left(\delta^3\right) \\ \text{where } \delta = \Delta \mathbf{f}\left(\mathbf{x}, \mathbf{u}\right) + \xi,\ v_{\mathbf{x}} = \tfrac{\partial}{\partial \mathbf{x}} v,\ v_{\mathbf{xx}} = \tfrac{\partial^2}{\partial \mathbf{x} \partial \mathbf{x}} v\end{gathered}$$

Now compute the expectation of the optimal value function at the next state, using the above Taylor-series expansion and only keeping terms up to first-order in Δ. The result is:

$$E\left[v\right] = v\left(\mathbf{x}\right) + \Delta \mathbf{f}\left(\mathbf{x}, \mathbf{u}\right)^{\mathsf{T}} v_{\mathbf{x}}\left(\mathbf{x}\right) + \tfrac{1}{2}\operatorname{tr}\left(\Delta S\left(\mathbf{x}, \mathbf{u}\right) v_{\mathbf{xx}}\left(\mathbf{x}\right)\right) + o\left(\Delta^2\right)$$

The trace term appears because

$$E\left[\xi^{\mathsf{T}} v_{\mathbf{xx}} \xi\right] = E\left[\operatorname{tr}\left(\xi \xi^{\mathsf{T}} v_{\mathbf{xx}}\right)\right] = \operatorname{tr}\left(\operatorname{Cov}\left[\xi\right] v_{\mathbf{xx}}\right) = \operatorname{tr}\left(\Delta S v_{\mathbf{xx}}\right)$$

Note the second-order derivative $v_{\mathbf{xx}}$ in the first-order approximation to $E\left[v\right]$. This is a recurrent theme in stochastic calculus. It is directly related to *Ito's lemma*, which states that if $x\left(t\right)$ is an Ito diffusion with coefficient σ, then

$$dg\left(x\left(t\right)\right) = g_x\left(x\left(t\right)\right) dx\left(t\right) + \tfrac{1}{2}\sigma^2 g_{xx}\left(x\left(t\right)\right) dt$$

Coming back to the derivation, we substitute the expression for $E\left[v\right]$ in (7), move the term $v\left(\mathbf{x}\right)$ outside the minimization operator (since it does not depend on $\mathbf{u}$), and divide by Δ. Suppressing $\mathbf{x}, \mathbf{u}, k$ on the right hand side, we have

$$\frac{v\left(\mathbf{x}, k\right) - v\left(\mathbf{x}, k+1\right)}{\Delta} = \min_{\mathbf{u}} \left\{\ell + \mathbf{f}^{\mathsf{T}} v_{\mathbf{x}} + \tfrac{1}{2}\operatorname{tr}\left(S v_{\mathbf{xx}}\right) + o\left(\Delta\right)\right\}$$

Recall that $t = k\Delta$, and consider the optimal value function $v(\mathbf{x}, t)$ defined in continuous time. The left hand side in the above equation is then

$$\frac{v(\mathbf{x}, t) - v(\mathbf{x}, t + \Delta)}{\Delta}$$

In the limit $\Delta \to 0$ the latter expression becomes $-\frac{\partial}{\partial t} v$, which we denote $-v_t$. Thus for $0 \leq t \leq t_f$ and $v(\mathbf{x}, t_f) = h(\mathbf{x})$, the following holds:

$$-v_t(\mathbf{x}, t) = \min_{\mathbf{u} \in \mathcal{U}(\mathbf{x})} \left\{ \ell(\mathbf{x}, \mathbf{u}, t) + \mathbf{f}(\mathbf{x}, \mathbf{u})^{\mathsf{T}} v_{\mathbf{x}}(\mathbf{x}, t) + \tfrac{1}{2} \operatorname{tr}(S(\mathbf{x}, \mathbf{u}) v_{\mathbf{xx}}(\mathbf{x}, t)) \right\} \tag{8}$$

Similarly to the discrete case, an optimal control $\pi(\mathbf{x}, t)$ is a value of $\mathbf{u}$ which achieves the minimum in (8):

$$\pi(\mathbf{x}, t) = \arg \min_{\mathbf{u} \in \mathcal{U}(\mathbf{x})} \left\{ \ell(\mathbf{x}, \mathbf{u}, t) + \mathbf{f}(\mathbf{x}, \mathbf{u})^{\mathsf{T}} v_{\mathbf{x}}(\mathbf{x}, t) + \tfrac{1}{2} \operatorname{tr}(S(\mathbf{x}, \mathbf{u}) v_{\mathbf{xx}}(\mathbf{x}, t)) \right\} \tag{9}$$

Equations (8) and (9) are the *Hamilton-Jacobi-Bellman* (HJB) equations.

2.2 Numerical solutions of the HJB equations

The HJB equation (8) is a non-linear (due to the min operator) second-order PDE with respect to the unknown function v. If a differentiable function v satisfying (8) for all $(\mathbf{x}, t)$ exists, then it is the unique optimal value function. However, non-linear differential equations do not always have classic solutions which satisfy them everywhere. For example, consider the equation $|\dot{g}(t)| = 1$ with boundary conditions $g(0) = g(1) = 0$. The slope of g is either $+1$ or -1, and so g has to change slope (discontinuously) somewhere in the interval $0 \leq t \leq 1$ in order to satisfy the boundary conditions. At the points where that occurs the derivative $\dot{g}(t)$ is undefined. If we decide to admit such "weak" solutions, we are faced with infinitely many solutions to the same differential equation. In particular when (8) does not have a classic solution, the optimal value function is a weak solution but there are many other weak solutions. How can we then solve the optimal control problem? The recent development of non-smooth analysis and the idea of *viscosity solutions* provide a reassuring answer. It can be summarized as follows: **(i)** "viscosity" provides a specific criterion for selecting a single weak solution; **(ii)** the optimal value function is a viscosity solution to the HJB equation (and thus it is the only viscosity solution); **(iii)** numerical approximation schemes which take the limit of solutions to discretized problems converge to a viscosity solution (and therefore to the optimal value function). The bottom line is that in practice one need not worry about the absence of classic solutions.

Unfortunately there are other practical issues to worry about. The only numerical methods guaranteed to converge to the optimal value function rely on discretizations of the state space, and the required number of discrete states is exponential in the state-space dimensionality n_x. Bellman called this the *curse of dimensionality.* It is a problem which most likely does not have a general solution. Nevertheless, the HJB equations have motivated a number of methods for approximate solution. Such methods rely on parametric models of the optimal value function, or the optimal control law, or both. Below we outline one such method.

Consider an approximation $\widetilde{v}(\mathbf{x}, t; \theta)$ to the optimal value function, where θ is some vector of parameters. Particularly convenient are models of the form

$$\widetilde{v}(\mathbf{x}, t; \theta) = \sum\nolimits_i \phi^i(\mathbf{x}, t)\, \theta_i$$

where $\phi^i(\mathbf{x}, t)$ are some predefined basis functions, and the unknown parameters θ appear linearly. Linearity in θ simplifies the calculation of derivatives:

$$\widetilde{v}_{\mathbf{x}}(\mathbf{x}, t; \theta) = \sum\nolimits_i \phi^i_{\mathbf{x}}(\mathbf{x}, t)\, \theta_i$$

and similarly for $\widetilde{v}_{\mathbf{xx}}$ and $\widetilde{v}_t$. Now choose a large enough set of states $(\mathbf{x}, t)$ and evaluate the right hand side of (8) at those states (using the approximation to v and minimizing over $\mathbf{u}$). This procedure yields target values for the left hand side of (8). Then adjust the parameters θ so that $-\widetilde{v}_t(\mathbf{x}, t; \theta)$ gets closer to these target values. The discrepancy being minimized by the parameter adjustment procedure is the *Bellman error*.

2.3 Infinite-horizon formulations

Thus far we focused on finite-horizon problems. There are two infinite-horizon formulations used in practice, both of which yield time-invariant forms of the HJB equations. One is the discounted-cost formulation, where the total cost for an (infinitely long) state-control trajectory is defined as

$$J(\mathbf{x}(\cdot), \mathbf{u}(\cdot)) = \int_0^\infty \exp(-\alpha t)\, \ell(\mathbf{x}(t), \mathbf{u}(t))\, dt$$

with $\alpha > 0$ being the discount factor. Intuitively this says that future costs are less costly (whatever that means). Here we do not have a final cost $h(\mathbf{x})$, and the cost rate $\ell(\mathbf{x}, \mathbf{u})$ no-longer depends on time explicitly. The HJB equation for the optimal value function becomes

$$\alpha v(\mathbf{x}) = \min_{\mathbf{u} \in \mathcal{U}(\mathbf{x})} \left\{ \ell(\mathbf{x}, \mathbf{u}) + \mathbf{f}(\mathbf{x}, \mathbf{u})^{\mathsf{T}} v_{\mathbf{x}}(\mathbf{x}) + \tfrac{1}{2} \operatorname{tr}(S(\mathbf{x}, \mathbf{u})\, v_{\mathbf{xx}}(\mathbf{x})) \right\} \tag{10}$$

Another alternative is the average-cost-per-stage formulation, with total cost

$$J(\mathbf{x}(\cdot), \mathbf{u}(\cdot)) = \lim_{t_f \to \infty} \frac{1}{t_f} \int_0^{t_f} \ell(\mathbf{x}(t), \mathbf{u}(t))\, dt$$

In this case the HJB equation for the optimal value function is

$$\lambda = \min_{\mathbf{u} \in \mathcal{U}(\mathbf{x})} \left\{ \ell(\mathbf{x}, \mathbf{u}) + \mathbf{f}(\mathbf{x}, \mathbf{u})^{\mathsf{T}} v_{\mathbf{x}}(\mathbf{x}) + \tfrac{1}{2} \operatorname{tr}(S(\mathbf{x}, \mathbf{u})\, v_{\mathbf{xx}}(\mathbf{x})) \right\} \tag{11}$$

where $\lambda \geq 0$ is the average cost per stage, and v now has the meaning of a differential value function.

Equations (10) and (11) do not depend on time, which makes them more amenable to numerical approximations in the sense that we do not need to store a copy of the optimal value function at each point in time. Form another point of view, however, (8) may be easier to solve numerically. This is because dynamic programming can be performed in a single backward pass through time: initialize $v\left(\mathbf{x},t_f\right)=h\left(\mathbf{x}\right)$ and simply integrate (8) backward in time, computing the spatial derivatives numerically along the way. In contrast, (10) and (11) call for relaxation methods (such as value iteration or policy iteration) which in the continuous-state case may take an arbitrary number of iterations to converge. Relaxation methods are of course guaranteed to converge in a finite number of iterations for any finite state approximation, but that number may increase rapidly as the discretization of the continuous state space is refined.

3 Deterministic control: Pontryagin's maximum principle

Optimal control theory is based on two fundamental ideas. One is dynamic programming and the associated optimality principle, introduced by Bellman in the United States. The other is the *maximum principle*, introduced by Pontryagin in the Soviet Union. The maximum principle applies only to deterministic problems, and yields the same solutions as dynamic programming. Unlike dynamic programming, however, the maximum principle avoids the curse of dimensionality. Here we derive the maximum principle indirectly via the HJB equation, and directly via Lagrange multipliers. We also clarify its relationship to classical mechanics.

3.1 Derivation via the HJB equations

For deterministic dynamics $\dot{\mathbf{x}}=\mathbf{f}\left(\mathbf{x},\mathbf{u}\right)$ the finite-horizon HJB equation (8) becomes

$$-v_t\left(\mathbf{x},t\right)=\min_{\mathbf{u}}\left\{\ell\left(\mathbf{x},\mathbf{u},t\right)+\mathbf{f}\left(\mathbf{x},\mathbf{u}\right)^{\mathsf{T}}v_{\mathbf{x}}\left(\mathbf{x},t\right)\right\} \tag{12}$$

Suppose a solution to the minimization problem in (12) is given by an optimal control law $\pi\left(\mathbf{x},t\right)$ which is differentiable in $\mathbf{x}$. Setting $\mathbf{u}=\pi\left(\mathbf{x},t\right)$ we can drop the min operator in (12) and write

$$0=v_t\left(\mathbf{x},t\right)+\ell\left(\mathbf{x},\pi\left(\mathbf{x},t\right),t\right)+\mathbf{f}\left(\mathbf{x},\pi\left(\mathbf{x},t\right)\right)^{\mathsf{T}}v_{\mathbf{x}}\left(\mathbf{x},t\right)$$

This equation is valid for all $\mathbf{x}$, and therefore can be differentiated w.r.t. $\mathbf{x}$ to obtain (in shortcut notation)

$$0=v_{t\mathbf{x}}+\ell_{\mathbf{x}}+\pi_{\mathbf{x}}^{\mathsf{T}}\ell_{\mathbf{u}}+\left(\mathbf{f}_{\mathbf{x}}^{\mathsf{T}}+\pi_{\mathbf{x}}^{\mathsf{T}}\mathbf{f}_{\mathbf{u}}^{\mathsf{T}}\right)v_{\mathbf{x}}+v_{\mathbf{x}\mathbf{x}}\mathbf{f}$$

Regrouping terms, and using the identity $\dot{v}_{\mathbf{x}}=v_{\mathbf{x}\mathbf{x}}\dot{\mathbf{x}}+v_{t\mathbf{x}}=v_{\mathbf{x}\mathbf{x}}\mathbf{f}+v_{t\mathbf{x}}$, yields

$$0=\dot{v}_{\mathbf{x}}+\ell_{\mathbf{x}}+\mathbf{f}_{\mathbf{x}}^{\mathsf{T}}v_{\mathbf{x}}+\pi_{\mathbf{x}}^{\mathsf{T}}\left(\ell_{\mathbf{u}}+\mathbf{f}_{\mathbf{u}}^{\mathsf{T}}v_{\mathbf{x}}\right)$$

We now make a key observation: the term in the brackets is the gradient w.r.t. $\mathbf{u}$ of the quantity being minimized w.r.t. $\mathbf{u}$ in (12). That gradient is zero (assuming unconstrained minimization), which leaves us with

$$-\dot{v}_{\mathbf{x}}\left(\mathbf{x},t\right)=\ell_{\mathbf{x}}\left(\mathbf{x},\pi\left(\mathbf{x},t\right),t\right)+\mathbf{f}_{\mathbf{x}}^{\mathsf{T}}\left(\mathbf{x},\pi\left(\mathbf{x},t\right)\right)v_{\mathbf{x}}\left(\mathbf{x},t\right) \tag{13}$$

This may look like a PDE for v, but if we think of $v_{\mathbf{x}}$ as a vector $\mathbf{p}$ instead of a gradient of a function which depends on $\mathbf{x}$, then (13) is an ordinary differential equation (ODE) for $\mathbf{p}$. That equation holds along any trajectory generated by $\pi(\mathbf{x}, t)$. The vector $\mathbf{p}$ is called the *costate* vector.

We are now ready to formulate the maximum principle. If $\{\mathbf{x}(t), \mathbf{u}(t) : 0 \leq t \leq t_f\}$ is an optimal state-control trajectory (obtained by initializing $\mathbf{x}(0)$ and controlling the system optimally until t_f), then there exists a costate trajectory $\mathbf{p}(t)$ such that (13) holds with $\mathbf{p}$ in place of $v_{\mathbf{x}}$ and $\mathbf{u}$ in place of π. The conditions on $\{\mathbf{x}(t), \mathbf{u}(t), \mathbf{p}(t)\}$ are

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)) \\ -\dot{\mathbf{p}}(t) &= \ell_{\mathbf{x}}(\mathbf{x}(t), \mathbf{u}(t), t) + \mathbf{f}_{\mathbf{x}}^{\mathsf{T}}(\mathbf{x}(t), \mathbf{u}(t))\,\mathbf{p}(t) \\ \mathbf{u}(t) &= \arg\min_{\underline{\mathbf{u}}} \left\{ \ell(\mathbf{x}(t), \underline{\mathbf{u}}, t) + \mathbf{f}(\mathbf{x}(t), \underline{\mathbf{u}})^{\mathsf{T}}\, \mathbf{p}(t) \right\} \end{aligned} \tag{14}$$

The boundary condition is $\mathbf{p}(t_f) = h_{\mathbf{x}}(\mathbf{x}(t_f))$, and $\mathbf{x}(0), t_f$ are given. Clearly the costate is equal to the gradient of the optimal value function evaluated along the optimal trajectory.

The maximum principle can be written in more compact and symmetric form with the help of the *Hamiltonian* function

$$H(\mathbf{x}, \mathbf{u}, \mathbf{p}, t) = \ell(\mathbf{x}, \mathbf{u}, t) + \mathbf{f}(\mathbf{x}, \mathbf{u})^{\mathsf{T}}\, \mathbf{p} \tag{15}$$

which is the quantity we have been minimizing w.r.t. $\mathbf{u}$ all along (it was about time we gave it a name). With this definition, (14) becomes

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \tfrac{\partial}{\partial \mathbf{p}} H(\mathbf{x}(t), \mathbf{u}(t), \mathbf{p}(t), t) \\ -\dot{\mathbf{p}}(t) &= \tfrac{\partial}{\partial \mathbf{x}} H(\mathbf{x}(t), \mathbf{u}(t), \mathbf{p}(t), t) \\ \mathbf{u}(t) &= \arg\min_{\underline{\mathbf{u}}} H(\mathbf{x}(t), \underline{\mathbf{u}}, \mathbf{p}(t), t) \end{aligned} \tag{16}$$

The remarkable property of the maximum principle is that it is an ODE, even though we derived it starting from a PDE. An ODE is a consistency condition which singles out specific trajectories without reference to neighboring trajectories (as would be the case in a PDE). This is possible because the *extremal* trajectories which solve (14) make $H_{\mathbf{u}} = \ell_{\mathbf{u}} + \mathbf{f}_{\mathbf{u}}^{\mathsf{T}} \mathbf{p}$ vanish, which in turn removes the dependence on neighboring trajectories. The ODE (14) is a system of $2n_x$ scalar equations subject to $2n_x$ scalar boundary conditions. Therefore we can solve this system with standard boundary-value solvers (such as Matlab's `bvp4c`). The only complication is that we would have to minimize the Hamiltonian repeatedly. This complication is avoided for a class of problems where the control appears linearly in the dynamics and quadratically in the cost rate:

$$\begin{aligned} &\text{dynamics:} && \dot{\mathbf{x}} = \mathbf{a}(\mathbf{x}) + B(\mathbf{x})\,\mathbf{u} \\ &\text{cost rate:} && \ell(\mathbf{x}, \mathbf{u}, t) = \tfrac{1}{2} \mathbf{u}^{\mathsf{T}} R(\mathbf{x})\,\mathbf{u} + q(\mathbf{x}, t) \end{aligned}$$

In such problems the Hamiltonian is quadratic and can be minimized explicitly:

$$\begin{aligned} H(\mathbf{x}, \mathbf{u}, \mathbf{p}, t) &= \tfrac{1}{2} \mathbf{u}^{\mathsf{T}} R(\mathbf{x})\,\mathbf{u} + q(\mathbf{x}, t) + (\mathbf{a}(\mathbf{x}) + B(\mathbf{x})\,\mathbf{u})^{\mathsf{T}}\, \mathbf{p} \\ \arg\min_{\mathbf{u}} H &= -R(\mathbf{x})^{-1} B(\mathbf{x})^{\mathsf{T}}\, \mathbf{p} \end{aligned}$$

The computational complexity (or at least the storage requirement) for ODE solutions based on the maximum principle grows linearly with the state dimensionality n_x, and so the curse of dimensionality is avoided. One drawback is that (14) could have multiple solutions (one of which is the optimal solution) but in practice that does not appear to be a serious problem. Another drawback of course is that the solution to (14) is valid for a single initial state, and if the initial state were to change we would have to solve the problem again. If the state change is small, however, the solution change should also be small, and so we can speed-up the search by initializing the ODE solver with the previous solution.

The maximum principle can be generalized in a number of ways including: terminal state constraints instead of "soft" final costs; state constraints at intermediate points along the trajectory; free (i.e. optimized) final time; first exit time; control constraints. It can also be applied in *model-predictive control* settings where one seeks an optimal state-control trajectory up to a fixed time horizon (and approximates the optimal value function at the horizon). The initial portion of this trajectory is used to control the system, and then a new optimal trajectory is computed. This is closely related to the idea of a *rollout policy* – which is essential in computer chess programs for example.

3.2 Derivation via Lagrange multipliers

The maximum principle can also be derived for discrete-time systems, as we show next. Note that the following derivation is actually the more standard one (in continuous time it relies on the calculus of variations). Consider the discrete-time optimal control problem

$$\begin{array}{ll} \text{dynamics:} & \mathbf{x}_{k+1} = \mathbf{f}\left(\mathbf{x}_k, \mathbf{u}_k\right) \\ \text{cost rate:} & \ell\left(\mathbf{x}_k, \mathbf{u}_k, k\right) \\ \text{final cost:} & h\left(\mathbf{x}_n\right) \end{array}$$

with given initial state $\mathbf{x}_0$ and final time n. We can approach this as a regular constrained optimization problem: find sequences $(\mathbf{u}_0, \mathbf{u}_1, \cdots \mathbf{u}_{n-1})$ and $(\mathbf{x}_0, \mathbf{x}_1, \cdots \mathbf{x}_n)$ minimizing J subject to constraints $\mathbf{x}_{k+1} = \mathbf{f}\left(\mathbf{x}_k, \mathbf{u}_k\right)$. Constrained optimization problems can be solved with the method of Lagrange multipliers. As a reminder, in order to minimize a scalar function $g\left(\mathbf{x}\right)$ subject to $c\left(\mathbf{x}\right) = 0$, we form the Lagrangian $L\left(\mathbf{x}, \lambda\right) = g\left(\mathbf{x}\right) + \lambda c\left(\mathbf{x}\right)$ and look for a pair $(\mathbf{x}, \lambda)$ such that $\frac{\partial}{\partial \mathbf{x}} L = 0$ and $\frac{\partial}{\partial \lambda} L = 0$.

In our case there are n constraints, so we need a sequence of n Lagrange multipliers $(\lambda_1, \lambda_2, \cdots \lambda_n)$. The Lagrangian is

$$L\left(\mathbf{x}_\cdot, \mathbf{u}_\cdot, \lambda_\cdot\right) = h\left(\mathbf{x}_n\right) + \sum\nolimits_{k=0}^{n-1} \left(\ell\left(\mathbf{x}_k, \mathbf{u}_k, k\right) + \left(\mathbf{f}\left(\mathbf{x}_k, \mathbf{u}_k\right) - \mathbf{x}_{k+1}\right)^\mathsf{T} \lambda_{k+1}\right)$$

Define the discrete-time Hamiltonian

$$H^{(k)}\left(\mathbf{x}, \mathbf{u}, \lambda\right) = \ell\left(\mathbf{x}, \mathbf{u}, k\right) + \mathbf{f}\left(\mathbf{x}, \mathbf{u}\right)^\mathsf{T} \lambda$$

and rearrange the terms in the Lagrangian to obtain

$$L = h\left(\mathbf{x}_n\right) - \mathbf{x}_n^\mathsf{T} \lambda_n + \mathbf{x}_0^\mathsf{T} \lambda_0 + \sum\nolimits_{k=0}^{n-1} \left(H^{(k)}\left(\mathbf{x}_k, \mathbf{u}_k, \lambda_{k+1}\right) - \mathbf{x}_k^\mathsf{T} \lambda_k\right)$$

Now consider differential changes in L due to changes in $\mathbf{u}$ which in turn lead to changes in $\mathbf{x}$. We have

$$\begin{aligned} dL = {} & \left(h_{\mathbf{x}}\left(\mathbf{x}_n\right) - \lambda_n\right)^{\mathsf{T}} d\mathbf{x}_n + \lambda_0^{\mathsf{T}} d\mathbf{x}_0 + \\ & + \sum\nolimits_{k=0}^{n-1} \left(\left(\tfrac{\partial}{\partial \mathbf{x}} H^{(k)} - \lambda_k \right)^{\mathsf{T}} d\mathbf{x}_k + \left(\tfrac{\partial}{\partial \mathbf{u}} H^{(k)} \right)^{\mathsf{T}} d\mathbf{u}_k \right) \end{aligned}$$

In order to satisfy $\frac{\partial}{\partial \mathbf{x}_k} L = 0$ we choose the Lagrange multipliers λ to be

$$\begin{aligned} \lambda_k &= \tfrac{\partial}{\partial \mathbf{x}} H^{(k)} = \ell_{\mathbf{x}}\left(\mathbf{x}_k, \mathbf{u}_k, k\right) + \mathbf{f}_{\mathbf{x}}^{\mathsf{T}}\left(\mathbf{x}_k, \mathbf{u}_k\right) \lambda_{k+1}, \quad 0 \leq k < n \\ \lambda_n &= h_{\mathbf{x}}\left(\mathbf{x}_n\right) \end{aligned}$$

For this choice of λ the differential dL becomes

$$dL = \lambda_0^{\mathsf{T}} d\mathbf{x}_0 + \sum\nolimits_{k=0}^{n-1} \left(\tfrac{\partial}{\partial \mathbf{u}} H^{(k)} \right)^{\mathsf{T}} d\mathbf{u}_k \tag{17}$$

The first term in (17) is 0 because $\mathbf{x}_0$ is fixed. The second term becomes 0 when $\mathbf{u}_k$ is the (unconstrained) minimum of $H^{(k)}$. Summarizing the conditions for an optimal solution, we arrive at the discrete-time maximum principle:

$$\begin{aligned} \mathbf{x}_{k+1} &= \mathbf{f}\left(\mathbf{x}_k, \mathbf{u}_k\right) \\ \lambda_k &= \ell_{\mathbf{x}}\left(\mathbf{x}_k, \mathbf{u}_k, k\right) + \mathbf{f}_{\mathbf{x}}^{\mathsf{T}}\left(\mathbf{x}_k, \mathbf{u}_k\right) \lambda_{k+1} \\ \mathbf{u}_k &= \arg\min_{\underline{\mathbf{u}}} H^{(k)}\left(\mathbf{x}_k, \underline{\mathbf{u}}, \lambda_{k+1}\right) \end{aligned} \tag{18}$$

with $\lambda_n = h_{\mathbf{x}}\left(\mathbf{x}_n\right)$, and $\mathbf{x}_0, n$ given.

The similarity between the discrete-time (18) and the continuous-time (14) versions of the maximum principle is obvious. The costate $\mathbf{p}$, which before was equal to the gradient $v_{\mathbf{x}}$ of the optimal value function, is now a Lagrange multiplier λ. Thus we have three different names for the same quantity. It actually has yet another name: *influence function*. This is because λ_0 is the gradient of the minimal total cost w.r.t. the initial condition $\mathbf{x}_0$ (as can be seen from 17) and so λ_0 tells us how changes in the initial condition influence the total cost. The minimal total cost is of course equal to the optimal value function, thus λ is the gradient of the optimal value function.

3.3 Numerical optimization via gradient descent

From (17) it is clear that the quantity

$$\tfrac{\partial}{\partial \mathbf{u}} H^{(k)} = \ell_{\mathbf{u}}\left(\mathbf{x}_k, \mathbf{u}_k, k\right) + \mathbf{f}_{\mathbf{u}}^{\mathsf{T}}\left(\mathbf{x}_k, \mathbf{u}_k\right) \lambda_{k+1} \tag{19}$$

is the gradient of the total cost with respect to the control signal. This also holds in the continuous-time case. Once we have a gradient, we can optimize the (open-loop) control sequence for given initial state via gradient descent. Here is the algorithm:

1. Given a control sequence, iterate the dynamics forward in time to find the corresponding state sequence. Then iterate (18) backward in time to find the Lagrange multiplier sequence. In the backward pass use the given control sequence instead of optimizing the Hamiltonian.

2. Evaluate the gradient using (19), and improve the control sequence with any gradient descent algorithm. Go back to step 1, or exit if converged.

As always, gradient descent in high-dimensional spaces is much more efficient if one uses a second-order method (conjugate gradient descent, Levenberg-Marquardt, Gauss-Newton, etc). Care should be taken to ensure stability. Stability of second-order optimization can be ensured via line-search or trust-region methods. To avoid local minima – which correspond to extremal trajectories that are not optimal – one could use multiple restarts with random initialization of the control sequence. Note that extremal trajectories satisfy the maximum principle, and so an ODE solver can get trapped in the same suboptimal solutions as a gradient descent method.

3.4 Relation to classical mechanics

We now return to the continuous-time maximum principle, and note that (16) resembles the Hamiltonian formulation of mechanics, with $\mathbf{p}$ being the generalized momentum (a fifth name for the same quantity). To see where the resemblance comes from, recall the Lagrange problem: given $\mathbf{x}(0)$ and $\mathbf{x}(t_f)$, find curves $\{\mathbf{x}(t) : \ 0 \leq t \leq t_f\}$ which optimize

$$J(\mathbf{x}(\cdot)) = \int_0^{t_f} L(\mathbf{x}(t), \dot{\mathbf{x}}(t))\, dt \tag{20}$$

Applying the calculus of variations, one finds that extremal curves (either maxima or minima of J) satisfy the *Euler-Lagrange equation*

$$\tfrac{d}{dt}\tfrac{\partial}{\partial \dot{\mathbf{x}}} L(\mathbf{x}, \dot{\mathbf{x}}) - \tfrac{\partial}{\partial \mathbf{x}} L(\mathbf{x}, \dot{\mathbf{x}}) = 0$$

Its solutions are known to be equivalent to the solutions of *Hamilton's equation*

$$\begin{aligned} \dot{\mathbf{x}} &= \tfrac{\partial}{\partial \mathbf{p}} H(\mathbf{x}, \mathbf{p}) \\ -\dot{\mathbf{p}} &= \tfrac{\partial}{\partial \mathbf{x}} H(\mathbf{x}, \mathbf{p}) \end{aligned} \tag{21}$$

where the Hamiltonian is defined as

$$H(\mathbf{x}, \mathbf{p}) = \mathbf{p}^{\mathsf{T}} \dot{\mathbf{x}} - L(\mathbf{x}, \dot{\mathbf{x}}) \tag{22}$$

The change of coordinates $\mathbf{p} = \frac{\partial}{\partial \dot{\mathbf{x}}} L(\mathbf{x}, \dot{\mathbf{x}})$ is called a *Legendre transformation.* It may seem strange that $H(\mathbf{x}, \mathbf{p})$ depends on $\dot{\mathbf{x}}$ when $\dot{\mathbf{x}}$ is not explicitly an argument of H. This is because the Legendre transformation is invertible, i.e. $\dot{\mathbf{x}}$ can be recovered from $(\mathbf{x}, \mathbf{p})$ as long as the matrix $\frac{\partial^2}{\partial \dot{\mathbf{x}} \partial \dot{\mathbf{x}}} L$ is non-singular.

Thus the trajectories satisfying Hamilton's equation (21) are solutions to the Lagrange optimization problem (20). In order to explicitly transform (20) into an optimal control problem, define a control signal $\mathbf{u}$ and deterministic dynamics $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u})$. Then the cost rate $\ell(\mathbf{x}, \mathbf{u}) = -L(\mathbf{x}, \mathbf{f}(\mathbf{x}, \mathbf{u})) = -L(\mathbf{x}, \dot{\mathbf{x}})$ yields an optimal control problem equivalent to (20). The Hamiltonian (22) becomes $H(\mathbf{x}, \mathbf{p}) = \mathbf{p}^{\mathsf{T}} \mathbf{f}(\mathbf{x}, \mathbf{u}) + \ell(\mathbf{x}, \mathbf{u})$, which is the optimal-control Hamiltonian (15). Note that we can choose any dynamics $\mathbf{f}(\mathbf{x}, \mathbf{u})$, and then define the corresponding cost rate $\ell(\mathbf{x}, \mathbf{u})$ so as to make the optimal control problem equivalent to the Lagrange problem (20). The simplest choice is $\mathbf{f}(\mathbf{x}, \mathbf{u}) = \mathbf{u}$.

The function L is interpreted as an energy function. In mechanics it is

$$L(\mathbf{x},\dot{\mathbf{x}}) = \tfrac{1}{2}\dot{\mathbf{x}}^{\mathsf{T}} M(\mathbf{x})\,\dot{\mathbf{x}} - g(\mathbf{x})$$

The first term is kinetic energy (with $M(\mathbf{x})$ being the inertia matrix), and the second term is potential energy due to gravity or some other force field. When the inertia is constant, applying the Euler-Lagrange equation to the above L yields Newton's second law $M\ddot{\mathbf{x}} = -g_{\mathbf{x}}(\mathbf{x})$, where the force $-g_{\mathbf{x}}(\mathbf{x})$ is the gradient of the potential field. If the inertia is not constant (joint-space inertia for a multi-joint arm for example) application of the Euler-Lagrange equation yields extra terms which capture nonlinear interaction forces. Geometrically, they contain Christoffel symbols of the Levi-Civita connection for the Riemannian metric given by $\langle \mathbf{y},\mathbf{z}\rangle_{\mathbf{x}} = \mathbf{y}^{\mathsf{T}} M(\mathbf{x})\,\mathbf{z}$. We will not discuss differential geometry in any detail here, but it is worth noting that it affords a coordinate-free treatment, revealing intrinsic properties of the dynamical system that are invariant with respect to arbitrary smooth changes of coordinates. Such invariant quantities are called tensors. For example, the metric (inertia in our case) is a tensor.

4 Linear-quadratic-Gaussian control: Riccati equations

Optimal control laws can rarely be obtained in closed form. One notable exception is the LQG case where the dynamics are linear, the costs are quadratic, and the noise (if present) is additive Gaussian. This makes the optimal value function quadratic, and allows minimization of the Hamiltonian in closed form. Here we derive the LQG optimal controller in continuous and discrete time.

4.1 Derivation via the HJB equations

Consider the following stochastic optimal control problem:

$$\begin{aligned}
&\text{dynamics:} && d\mathbf{x} = (A\mathbf{x} + B\mathbf{u})\,dt + F d\mathbf{w}\\
&\text{cost rate:} && \ell(\mathbf{x},\mathbf{u}) = \tfrac{1}{2}\mathbf{u}^{\mathsf{T}} R\mathbf{u} + \tfrac{1}{2}\mathbf{x}^{\mathsf{T}} Q\mathbf{x}\\
&\text{final cost:} && h(\mathbf{x}) = \tfrac{1}{2}\mathbf{x}^{\mathsf{T}} Q^{f}\mathbf{x}
\end{aligned}$$

where R is symmetric positive-definite, Q and Q^f are symmetric, and $\mathbf{u}$ is now unconstrained. We set $S = FF^{\mathsf{T}}$ as before. The matrices A, B, F, R, Q can be made time-varying without complicating the derivation below.

In order to solve for the optimal value function we will guess its parametric form, show that it satisfies the HJB equation (8), and obtain ODEs for its parameters. Our guess is

$$v(\mathbf{x},t) = \tfrac{1}{2}\mathbf{x}^{\mathsf{T}} V(t)\,\mathbf{x} + a(t) \tag{23}$$

where $V(t)$ is symmetric. The boundary condition $v(\mathbf{x},t_f) = h(\mathbf{x})$ implies $V(t_f) = Q^f$ and $a(t_f) = 0$. From (23) we can compute the derivatives which enter into the HJB equation:

$$\begin{aligned}
v_t(\mathbf{x},t) &= \tfrac{1}{2}\mathbf{x}^{\mathsf{T}}\dot{V}(t)\,\mathbf{x} + \dot{a}(t)\\
v_{\mathbf{x}}(\mathbf{x},t) &= V(t)\,\mathbf{x}\\
v_{\mathbf{xx}}(\mathbf{x},t) &= V(t)
\end{aligned}$$

Substituting these expressions in (8) yields

$$\begin{aligned}&-\tfrac{1}{2}\mathbf{x}^{\mathsf{T}}\dot{V}(t)\,\mathbf{x}-\dot{a}(t)=\\&=\min_{\mathbf{u}}\left\{\tfrac{1}{2}\mathbf{u}^{\mathsf{T}}R\mathbf{u}+\tfrac{1}{2}\mathbf{x}^{\mathsf{T}}Q\mathbf{x}+(A\mathbf{x}+B\mathbf{u})^{\mathsf{T}}V(t)\,\mathbf{x}+\tfrac{1}{2}\operatorname{tr}(SV(t))\right\}\end{aligned}$$

The Hamiltonian (i.e. the term inside the min operator) is quadratic in $\mathbf{u}$, and its Hessian R is positive-definite, so the optimal control can be found analytically:

$$\mathbf{u}=-R^{-1}B^{\mathsf{T}}V(t)\,\mathbf{x} \tag{24}$$

With this $\mathbf{u}$, the control-dependent part of the Hamiltonian becomes

$$\tfrac{1}{2}\mathbf{u}^{\mathsf{T}}R\mathbf{u}+(B\mathbf{u})^{\mathsf{T}}V(t)\,\mathbf{x}=-\tfrac{1}{2}\mathbf{x}^{\mathsf{T}}V(t)\,BR^{-1}B^{\mathsf{T}}V(t)\,\mathbf{x}$$

After grouping terms, the HJB equation reduces to

$$\begin{aligned}&-\tfrac{1}{2}\mathbf{x}^{\mathsf{T}}\dot{V}(t)\,\mathbf{x}-\dot{a}(t)=\\&=-\tfrac{1}{2}\mathbf{x}^{\mathsf{T}}\left(Q+A^{\mathsf{T}}V(t)+V(t)\,A-V(t)\,BR^{-1}B^{\mathsf{T}}V(t)\right)\mathbf{x}+\tfrac{1}{2}\operatorname{tr}(SV(t))\end{aligned}$$

where we replaced the term $2A^{\mathsf{T}}V$ with $A^{\mathsf{T}}V+VA$ to make the equation symmetric. This is justified because $\mathbf{x}^{\mathsf{T}}A^{\mathsf{T}}V\mathbf{x}=\mathbf{x}^{\mathsf{T}}V^{\mathsf{T}}A\mathbf{x}=\mathbf{x}^{\mathsf{T}}VA\mathbf{x}$.

Our guess of the optimal value function is correct if and only if the above equation holds for all $\mathbf{x}$, which is the case when the $\mathbf{x}$-dependent terms are matched:

$$\begin{aligned}-\dot{V}(t)&=Q+A^{\mathsf{T}}V(t)+V(t)\,A-V(t)\,BR^{-1}B^{\mathsf{T}}V(t)\\-\dot{a}(t)&=\tfrac{1}{2}\operatorname{trace}(SV(t))\end{aligned} \tag{25}$$

Functions V,a satisfying (25) can obviously be found by initializing $V(t_f)=Q^f,\ a(t_f)=0$ and integrating the ODEs (25) backward in time. Thus (23) is the optimal value function with V,a given by (25), and (24) is the optimal control law (which in this case is unique).

The first line of (25) is called a *continuous-time Riccati equation*. Note that it does not depend on the noise covariance S. Consequently the optimal control law (24) is also independent of S. The only effect of S is on the total cost. As a corollary, the optimal control law remains the same in the deterministic case – called the *linear-quadratic regulator* (LQR).

4.2 Derivation via the Bellman equations

In practice one usually works with discrete-time systems. To obtain an optimal control law for the discrete-time case one could use an Euler approximation to (25), but the resulting equation is missing terms quadratic in the time step Δ, as we will see below. Instead we apply dynamic programming directly, and obtain an exact solution to the discrete-time LQR problem. Dropping the (irrelevant) noise and discretizing the problem, we obtain

$$\begin{aligned}&\text{dynamics:} && \mathbf{x}_{k+1}=A\mathbf{x}_k+B\mathbf{u}_k\\&\text{cost rate:} && \tfrac{1}{2}\mathbf{u}_k^{\mathsf{T}}R\mathbf{u}_k+\tfrac{1}{2}\mathbf{x}_k^{\mathsf{T}}Q\mathbf{x}_k\\&\text{final cost:} && \tfrac{1}{2}\mathbf{x}_n^{\mathsf{T}}Q^f\mathbf{x}_n\end{aligned}$$

where $n = t_f/\Delta$ and the correspondence to the continuous-time problem is

$$\mathbf{x}_k \leftarrow \mathbf{x}(k\Delta), \; A \leftarrow (\mathrm{I} + \Delta A), \; B \leftarrow \Delta B, \; R \leftarrow \Delta R, \; Q \leftarrow \Delta Q \tag{26}$$

The guess for the optimal value function is again quadratic

$$v(\mathbf{x}, k) = \tfrac{1}{2}\mathbf{x}^\mathsf{T} V_k \mathbf{x}$$

with boundary condition $V_n = Q^f$. The Bellman equation (2) is

$$\tfrac{1}{2}\mathbf{x}^\mathsf{T} V_k \mathbf{x} = \min_{\mathbf{u}} \left\{ \tfrac{1}{2}\mathbf{u}^\mathsf{T} R \mathbf{u} + \tfrac{1}{2}\mathbf{x}^\mathsf{T} Q \mathbf{x} + \tfrac{1}{2}(A\mathbf{x} + B\mathbf{u})^\mathsf{T} V_{k+1} (A\mathbf{x} + B\mathbf{u}) \right\}$$

As in the continuous-time case the Hamiltonian can be minimized analytically. The resulting optimal control law is

$$\mathbf{u} = -\left(R + B^\mathsf{T} V_{k+1} B\right)^{-1} B^\mathsf{T} V_{k+1} A \mathbf{x}$$

Substituting this $\mathbf{u}$ in the Bellman equation, we obtain

$$V_k = Q + A^\mathsf{T} V_{k+1} A - A^\mathsf{T} V_{k+1} B \left(R + B^\mathsf{T} V_{k+1} B\right)^{-1} B^\mathsf{T} V_{k+1} A \tag{27}$$

This completes the proof that the optimal value function is in the assumed quadratic form. To compute V_k for all k we initialize $V_n = Q^f$ and iterate (27) backward in time.

The optimal control law is linear in $\mathbf{x}$, and is usually written as

$$\mathbf{u}_k = -L_k \mathbf{x}_k \tag{28}$$
$$\text{where } L_k = \left(R + B^\mathsf{T} V_{k+1} B\right)^{-1} B^\mathsf{T} V_{k+1} A$$

The time-varying matrix L_k is called the *control gain*. It does not depend on the sequence of states, and therefore can be computed offline. Equation (27) is called a *discrete-time Riccati equation*. Clearly the discrete-time Riccati equation contains more terms that the continuous-time Riccati equation (25), and so the two are not identical. However one can verify that they become identical in the limit $\Delta \to 0$. To this end replace the matrices in (27) with their continuous-time analogues (26), and after rearrangement obtain

$$\frac{V_k - V_{k+1}}{\Delta} = Q + A^\mathsf{T} V_{k+1} + V_{k+1} A - V_{k+1} B \left(R + \Delta B^\mathsf{T} V_{k+1} B\right)^{-1} B^\mathsf{T} V_{k+1} + \frac{o\left(\Delta^2\right)}{\Delta}$$

where $o\left(\Delta^2\right)$ absorbs terms that are second-order in Δ. Taking the limit $\Delta \to 0$ yields the continuous-time Riccati equation (25).

4.3 Applications to nonlinear problems

Apart from solving LQG problems, the methodology described here can be adapted to yield approximate solutions to non-LQG optimal control problems. This is done iteratively, as follows:

1. Given a control sequence, apply it to the (nonlinear) dynamics and obtain a corresponding state sequence.

2. Construct a time-varying linear approximation to the dynamics and a time-varying quadratic approximation to the cost; both approximations are centered at the state-control sequence obtained in step 1. This yields an LQG optimal control problem with respect to the state and control deviations.

3. Solve the resulting LQG problem, obtain the control deviation sequence, and add it to the given control sequence. Go to step 1, or exit if converged. Note that multiplying the deviation sequence by a number smaller than 1 can be used to implement line-search.

Another possibility is to use *differential dynamic programming* (DDP), which is based on the same idea but involves a second-order rather than a first-order approximation to the dynamics. In that case the approximate problem is not LQG, however one can assume a quadratic approximation to the optimal value function and derive Riccati-like equations for its parameters. DDP and iterative LQG (iLQG) have second-order convergence in the neighborhood of an optimal solution. They can be thought of as the analog of Newton's method in the domain of optimal control. Unlike general-purpose second order methods which construct Hessian approximations using gradient information, DDP and iLQG obtain the Hessian directly by exploiting the problem structure. For deterministic problems they converge to state-control trajectories which satisfy the maximum principle, but in addition yield local feedback control laws. In our experience they are more efficient that either ODE or gradient descent methods. iLQG has been generalized to stochastic systems (including multiplicative noise) and to systems subject to control constraints.

5 Optimal estimation: Kalman filter

Optimal control is closely related to optimal estimation, for two reasons: **(i)** the only way to achieve optimal performance in the presence of sensor noise and delays is to incorporate an optimal estimator in the control system; **(ii)** the two problems are dual, as explained below. The most widely used optimal estimator is the *Kalman filter*. It is the dual of the linear-quadratic regulator – which in turn is the most widely used optimal controller.

5.1 The Kalman filter

Consider the partially-observable linear dynamical system

$$\begin{aligned} &\text{dynamics:} && \mathbf{x}_{k+1} = A\mathbf{x}_k + \mathbf{w}_k \\ &\text{observation:} && \mathbf{y}_k = H\mathbf{x}_k + \mathbf{v}_k \end{aligned} \tag{29}$$

where $\mathbf{w}_k \sim \mathcal{N}(0, S)$ and $\mathbf{v}_k \sim \mathcal{N}(0, P)$ are independent Gaussian random variables, the initial state has a Gaussian prior distribution $\mathbf{x}_0 \sim \mathcal{N}(\widehat{\mathbf{x}}_0, \Sigma_0)$, and $A, H, S, P, \widehat{\mathbf{x}}_0, \Sigma_0$ are known. The states are hidden and all we have access to are the observations. The objective is to compute the posterior probability distribution $\widehat{p}_k$ of $\mathbf{x}_k$ given observations $\mathbf{y}_{k-1} \cdots \mathbf{y}_0$:

$$\begin{aligned} \widehat{p}_k &= p(\mathbf{x}_k | \mathbf{y}_{k-1} \cdots \mathbf{y}_0) \\ \widehat{p}_0 &= \mathcal{N}(\widehat{\mathbf{x}}_0, \Sigma_0) \end{aligned}$$

Note that our formulation is somewhat unusual: we are estimating $\mathbf{x}_k$ before $\mathbf{y}_k$ has been observed. This formulation is adopted here because it simplifies the results and also because most real-world sensors provide delayed measurements.

We will show by induction (moving forward in time) that $\widehat{p}_k$ is Gaussian for all k, and therefore can be represented by its mean $\widehat{\mathbf{x}}_k$ and covariance matrix Σ_k. This holds for $k = 0$ by definition. The Markov property of (29) implies that the posterior $\widehat{p}_k$ can be treated as prior over $\mathbf{x}_k$ for the purposes of estimation after time k. Since $\widehat{p}_k$ is Gaussian and (29) is linear-Gaussian, the joint distribution of $\mathbf{x}_{k+1}$ and $\mathbf{y}_k$ is also Gaussian. Its mean and covariance given the prior $\widehat{p}_k$ are easily computed:

$$E\begin{bmatrix}\mathbf{x}_{k+1}\\ \mathbf{y}_k\end{bmatrix}=\begin{bmatrix}A\widehat{\mathbf{x}}_k\\ H\widehat{\mathbf{x}}_k\end{bmatrix},\quad \mathrm{Cov}\begin{bmatrix}\mathbf{x}_{k+1}\\ \mathbf{y}_k\end{bmatrix}=\begin{bmatrix}S+A\Sigma_kA^\mathsf{T} & A\Sigma_kH^\mathsf{T}\\ H\Sigma_kA^\mathsf{T} & P+H\Sigma_kH^\mathsf{T}\end{bmatrix}$$

Now we need to compute the probability of $\mathbf{x}_{k+1}$ conditional on the new observation $\mathbf{y}_k$. This is done using an important property of multivariate Gaussians summarized in the following lemma:

Let $\mathbf{p}$ and $\mathbf{q}$ be jointly Gaussian, with means $\overline{\mathbf{p}}$ and $\overline{\mathbf{q}}$ and covariances $\Sigma_{\mathbf{pp}}$, $\Sigma_{\mathbf{qq}}$ and $\Sigma_{\mathbf{pq}} = \Sigma_{\mathbf{qp}}^\mathsf{T}$. Then the conditional distribution of $\mathbf{p}$ given $\mathbf{q}$ is Gaussian, with mean and covariance

$$\begin{aligned}E\left[\mathbf{p}|\mathbf{q}\right] &= \overline{\mathbf{p}}+\Sigma_{\mathbf{pq}}\Sigma_{\mathbf{qq}}^{-1}\left(\mathbf{q}-\overline{\mathbf{q}}\right)\\ \mathrm{Cov}\left[\mathbf{p}|\mathbf{q}\right] &= \Sigma_{\mathbf{pp}}-\Sigma_{\mathbf{pq}}\Sigma_{\mathbf{qq}}^{-1}\Sigma_{\mathbf{qp}}\end{aligned}$$

Applying the lemma to our problem, we see that $\widehat{p}_{k+1}$ is Gaussian with mean

$$\widehat{\mathbf{x}}_{k+1}=A\widehat{\mathbf{x}}_k+A\Sigma_kH^\mathsf{T}\left(P+H\Sigma_kH^\mathsf{T}\right)^{-1}\left(\mathbf{y}_k-H\widehat{\mathbf{x}}_k\right) \tag{30}$$

and covariance matrix

$$\Sigma_{k+1}=S+A\Sigma_kA^\mathsf{T}-A\Sigma_kH^\mathsf{T}\left(P+H\Sigma_kH^\mathsf{T}\right)^{-1}H\Sigma_kA^\mathsf{T} \tag{31}$$

This completes the induction proof. Equation (31) is a Riccati equation. Equation (30) is usually written as

$$\begin{aligned}\widehat{\mathbf{x}}_{k+1} &= A\widehat{\mathbf{x}}_k+K_k\left(\mathbf{y}_k-H\widehat{\mathbf{x}}_k\right)\\ \text{where } K_k &= A\Sigma_kH^\mathsf{T}\left(P+H\Sigma_kH^\mathsf{T}\right)^{-1}\end{aligned}$$

The time-varying matrix K_k is called the *filter gain.* It does not depend on the observation sequence and therefore can be computed offline. The quantity $\mathbf{y}_k - H\widehat{\mathbf{x}}_k$ is called the *innovation.* It is the mismatch between the observed and the expected measurement. The covariance Σ_k of the posterior probability distribution $p\left(\mathbf{x}_k|\mathbf{y}_{k-1}\cdots\mathbf{y}_0\right)$ is the *estimation error covariance.* The estimation error is $\mathbf{x}_k - \widehat{\mathbf{x}}_k$.

The above derivation corresponds to the discrete-time Kalman filter. A similar result holds in continuous time, and is called the *Kalman-Bucy filter.* It is possible to write down the Kalman filter in equivalent forms which have numerical advantages. One such approach is to propagate the matrix square root of Σ. This is called a *square-root filter*, and involves Riccati-like equations which are more stable because the dynamic range of the elements of Σ is reduced. Another approach is to propagate the inverse covariance Σ^{-1}. This is called an *information filter*, and again involves Riccati-like equations. The information filter can represent numerically very large covariances (and even infinite covariances – which are useful for specifying "non-informative" priors).

Instead of filtering one can do smoothing, i.e. obtain state estimates using observations from the past and from the future. In that case the posterior probability of each state given all observations is still Gaussian, and its parameters can be found by an additional backward pass known as *Rauch recursion.* The resulting *Kalman smoother* is closely related to the forward-backward algorithm for probabilistic inference in *hidden Markov models* (HMMs).

The Kalman filter is optimal in many ways. First of all it is a Bayesian filter, in the sense that it computes the posterior probability distribution over the hidden state. In addition, the mean $\widehat{\mathbf{x}}$ is the optimal point estimator with respect to multiple loss functions. Recall that optimality of point estimators is defined through a loss function $\ell(\mathbf{x}, \widehat{\mathbf{x}})$ which quantifies how bad it is to estimate $\widehat{\mathbf{x}}$ when the true state is $\mathbf{x}$. Some possible loss functions are $\|\mathbf{x} - \widehat{\mathbf{x}}\|^2$, $\|\mathbf{x} - \widehat{\mathbf{x}}\|$, $\delta(\mathbf{x} - \widehat{\mathbf{x}})$, and the corresponding optimal estimators are the mean, median, and mode of the posterior probability distribution. If the posterior is Gaussian then the mean, median and mode coincide. If we choose an unusual loss function for which the optimal point estimator is not $\widehat{\mathbf{x}}$, even though the posterior is Gaussian, the information contained in $\widehat{\mathbf{x}}$ and Σ is still sufficient to compute the optimal point estimator. This is because $\widehat{\mathbf{x}}$ and Σ are *sufficient statistics* which fully describe the posterior probability distribution, which in turn captures all information about the state that is available in the observation sequence. The set of sufficient statistics can be thought of as an augmented state, with respect to which the partially-observed process has a Markov property. It is called the *belief state* or alternatively the *information state.*

5.2 Beyond the Kalman filter

When the estimation problem involves non-linear dynamics or non-Gaussian noise the posterior probability distribution rarely has a finite set of sufficient statistics (although there are exceptions such as the *Benes* system). In that case one has to rely on numerical approximations. The most widely used approximation is the *extended Kalman filter* (EKF). It relies on local linearization centered at the current state estimate and closely resembles the LQG approximation to non-LQG optimal control problems. The EKF is not guaranteed to be optimal in any sense, but in practice if often yields good results – especially when the posterior is single-peaked. There is a recent improvement, called the *unscented filter*, which propagates the covariance using deterministic sampling instead of linearization of the system dynamics. The unscented filter tends to be superior to the EKF and requires a comparable amount of computation. An even more accurate, although computationally more expensive approach, is *particle filtering.* Instead of propagating a Gaussian approximation it propagates a cloud of points sampled from the posterior (without actually computing the posterior). Key to its success is the idea of *importance sampling.*

Even when the posterior does not have a finite set of sufficient statistics, it is still a well-defined scalar function over the state space, and as such must obey some equation. In discrete time this equation is simply a recursive version of Bayes' rule – which is not too revealing. In continuous time, however, the posterior satisfies a PDE which resembles the HJB equation. Before we present this result we need some notation. Consider the stochastic differential equations

$$\begin{aligned} d\mathbf{x} &= \mathbf{f}(\mathbf{x})\,dt + F(\mathbf{x})\,d\mathbf{w} \\ d\mathbf{y} &= \mathbf{h}(\mathbf{x})\,dt + d\mathbf{v} \end{aligned} \tag{32}$$

where $\mathbf{w}(t)$ and $\mathbf{v}(t)$ are Brownian motion processes, $\mathbf{x}(t)$ is the hidden state, and $\mathbf{y}(t)$ is the observation sequence. Define $S(\mathbf{x}) = F(\mathbf{x})\,F(\mathbf{x})^{\mathsf{T}}$ as before. One would normally think of the increments of $\mathbf{y}(t)$ as being the observations, but in continuous time these increments are infinite and so we work with their time-integral.

Let $p(\mathbf{x}, t)$ be the probability distribution of $\mathbf{x}(t)$ in the absence of any observations. At $t = 0$ it is initialized with a given prior $p(\mathbf{x}, 0)$. For $t > 0$ it is governed by the first line of (32), and can be shown to satisfy

$$p_t = -\mathbf{f}^{\mathsf{T}} p_{\mathbf{x}} + \tfrac{1}{2}\operatorname{tr}(S p_{\mathbf{xx}}) + \left(-\sum\nolimits_i \tfrac{\partial}{\partial x_i} f_i + \tfrac{1}{2}\sum\nolimits_{ij} \tfrac{\partial^2}{\partial x_i \partial x_j} S_{ij}\right) p$$

This is called the *forward Kolmogorov equation*, or alternatively the *Fokker-Planck equation.* We have written it in expanded form to emphasize the resemblance to the HJB equation. The more usual form is

$$\tfrac{\partial}{\partial t} p = -\sum\nolimits_i \tfrac{\partial}{\partial x_i}(f_i p) + \tfrac{1}{2}\sum\nolimits_{ij} \tfrac{\partial^2}{\partial x_i \partial x_j}(S_{ij} p)$$

Let $\widetilde{p}(\mathbf{x}, t)$ be an unnormalized posterior over $\mathbf{x}(t)$ given the observations $\{\mathbf{y}(s) : 0 \leq s \leq t\}$; "unnormalized" means that the actual posterior $\widehat{p}(\mathbf{x}, t)$ can be recovered by normalizing: $\widehat{p}(\mathbf{x}, t) = \widetilde{p}(\mathbf{x}, t) \,/ \int \widetilde{p}(\mathbf{z}, t)\,d\mathbf{z}$. It can be shown that some unnormalized posterior $\widetilde{p}$ satisfies *Zakai's equation*

$$d\widetilde{p} = \left(-\sum\nolimits_i \tfrac{\partial}{\partial x_i}(f_i \widetilde{p}) + \tfrac{1}{2}\sum\nolimits_{ij} \tfrac{\partial^2}{\partial x_i \partial x_j}(S_{ij} \widetilde{p})\right) dt + \mathbf{h}^{\mathsf{T}} \widetilde{p}\, d\mathbf{y}$$

The first term on the right reflects the prior and is the same as in the Kolmogorov equation (except that we have multiplied both sides by dt). The second term incorporates the observation and makes Zakai's equation a stochastic PDE. After certain manipulations (conversion to Stratonovich form and a gauge transformation) the second term can be integrated by parts, leading to a regular PDE. One can then approximate the solution to that PDE numerically via discretization methods. As in the HJB equation, however, such methods are only applicable in low-dimensional spaces due to the curse of dimensionality.

6 Duality of optimal control and optimal estimation

Optimal control and optimal estimation are closely related mathematical problems. The best-known example is the duality of the linear-quadratic regulator and the Kalman filter. To see that duality more clearly, we repeat the corresponding Riccati equations (27) and (31) side by side:

$$\begin{aligned} \text{control:} \quad & V_k = Q + A^{\mathsf{T}} V_{k+1} A - A^{\mathsf{T}} V_{k+1} B \left(R + B^{\mathsf{T}} V_{k+1} B\right)^{-1} B^{\mathsf{T}} V_{k+1} A \\ \text{filtering:} \quad & \Sigma_{k+1} = S + A \Sigma_k A^{\mathsf{T}} - A \Sigma_k H^{\mathsf{T}} \left(P + H \Sigma_k H^{\mathsf{T}}\right)^{-1} H \Sigma_k A^{\mathsf{T}} \end{aligned}$$

These equations are identical up to a time reversal and some matrix transposes. The correspondence is given in the following table:

$$\begin{array}{lcccccc} \text{control:} & V & Q & A & B & R & k \\ \Updownarrow & & & & & & \\ \text{filtering:} & \Sigma & S & A^{\mathsf{T}} & H^{\mathsf{T}} & P & n-k \end{array} \tag{33}$$

The above duality was first described by Kalman in his famous 1960 paper introducing the discrete-time Kalman filter, and is now mentioned in most books on estimation and control. However its origin and meaning are not apparent. This is because the Kalman filter is optimal from multiple points of view and can be written in multiple forms, making it hard to tell which of its properties have a dual in the control domain.

Attempts to generalize the duality to non-LQG settings have revealed that the fundamental relationship is between the optimal value function and the negative log-posterior. This is actually inconsistent with (33), although the inconsistency has not been made explicit before. Recall that V is the Hessian of the optimal value function. The posterior is Gaussian with covariance matrix Σ, and thus the Hessian of the negative log-posterior is Σ^{-1}. However in (33) we have V corresponding to Σ and not to Σ^{-1}. Another problem is that while A^{T} in (33) makes sense for linear dynamics $\dot{\mathbf{x}} = A\mathbf{x}$, the meaning of "transpose" for general non-linear dynamics $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$ in unclear. Thus the duality described by Kalman is specific to linear-quadratic-Gaussian systems and does not generalize.

6.1 General duality of optimal control and MAP smoothing

We now discuss an alternative approach which does generalize. In fact we start with the general case and later specialize it to the LQG setting to obtain something known as a *minimum-energy* estimator. As far as we know, the general treatment presented here is a novel result. The duality we establish is between maximum a posteriori (MAP) smoothing and deterministic optimal control for systems with continuous state. For systems with discrete state (i.e. HMMs) an instance of such duality is the *Viterbi algorithm* – which finds the most likely state sequence using dynamic programming.

Consider the discrete-time partially-observable system

$$\begin{aligned} p(\mathbf{x}_{k+1}|\mathbf{x}_k) &= \exp(-a(\mathbf{x}_{k+1}, \mathbf{x}_k)) \\ p(\mathbf{y}_k|\mathbf{x}_k) &= \exp(-b(\mathbf{y}_k, \mathbf{x}_k)) \\ p(\mathbf{x}_0) &= \exp(-c(\mathbf{x}_0)) \end{aligned} \tag{34}$$

where a, b, c are the negative log-probabilities of the state transitions, observation emissions, and initial state respectively. The states are hidden and we only have access to the observations $(\mathbf{y}_1, \mathbf{y}_2, \cdots \mathbf{y}_n)$. Our objective is to find the most probable sequence of states $(\mathbf{x}_0, \mathbf{x}_1, \cdots \mathbf{x}_n)$, that is, the sequence which maximizes the posterior probability

$$p(\mathbf{x}_\cdot|\mathbf{y}_\cdot) = \frac{p(\mathbf{y}_\cdot|\mathbf{x}_\cdot)\, p(\mathbf{x}_\cdot)}{p(\mathbf{y}_\cdot)}$$

The term $p(\mathbf{y}_\cdot)$ does not affect the maximization and so it can be dropped. Using the

Markov property of (34) we have

$$\begin{aligned} p\left(\mathbf{y}_\cdot|\mathbf{x}_\cdot\right)p\left(\mathbf{x}_\cdot\right) &= p\left(\mathbf{x}_0\right)\prod\nolimits_{k=1}^{n} p\left(\mathbf{x}_k|\mathbf{x}_{k-1}\right)p\left(\mathbf{y}_k|\mathbf{x}_k\right) \\ &= \exp\left(-c\left(\mathbf{x}_0\right)\right)\prod\nolimits_{k=1}^{n}\exp\left(-a\left(\mathbf{x}_k,\mathbf{x}_{k-1}\right)\right)\exp\left(-b\left(\mathbf{y}_k,\mathbf{x}_k\right)\right) \\ &= \exp\Big(-c\left(\mathbf{x}_0\right)-\sum\nolimits_{k=1}^{n}\left(a\left(\mathbf{x}_k,\mathbf{x}_{k-1}\right)+b\left(\mathbf{y}_k,\mathbf{x}_k\right)\right)\Big) \end{aligned}$$

Maximizing $\exp\left(-J\right)$ is equivalent to minimizing J. Therefore the most probable state sequence is the one which minimizes

$$J\left(\mathbf{x}_\cdot\right) = c\left(\mathbf{x}_0\right)+\sum\nolimits_{k=1}^{n}\left(a\left(\mathbf{x}_k,\mathbf{x}_{k-1}\right)+b\left(\mathbf{y}_k,\mathbf{x}_k\right)\right) \tag{35}$$

This is beginning to look like a total cost for a deterministic optimal control problem. However we are still missing a control signal. To remedy that we will define the passive dynamics as the expected state transition:

$$\mathbf{f}\left(\mathbf{x}_k\right) = E\left[\mathbf{x}_{k+1}|\mathbf{x}_k\right] = \int \mathbf{z}\exp\left(-a\left(\mathbf{z},\mathbf{x}_k\right)\right)d\mathbf{z}$$

and then define the control signal as the deviation from the expected state transition:

$$\mathbf{x}_{k+1} = \mathbf{f}\left(\mathbf{x}_k\right)+\mathbf{u}_k \tag{36}$$

The control cost is now defined as

$$r\left(\mathbf{u}_k,\mathbf{x}_k\right) = a\left(\mathbf{f}\left(\mathbf{x}_k\right)+\mathbf{u}_k,\mathbf{x}_k\right), \quad 0\le k<n$$

and the state cost is defined as

$$\begin{aligned} q\left(\mathbf{x}_0,0\right) &= c\left(\mathbf{x}_0\right) \\ q\left(\mathbf{x}_k,k\right) &= b\left(\mathbf{y}_k,\mathbf{x}_k\right), \quad 0<k\le n \end{aligned}$$

The observation sequence is fixed, and so q is well-defined as long as it depends explicitly on the time index k. Note that we could have chosen any $\mathbf{f}$, however the present choice will make intuitive sense later.

With these definitions, the control system with dynamics (36), cost rate

$$\ell\left(\mathbf{x}_k,\mathbf{u}_k,k\right) = r\left(\mathbf{u}_k,\mathbf{x}_k\right)+q\left(\mathbf{x}_k,k\right), \quad 0\le k<n$$

and final cost $q\left(\mathbf{x}_n,n\right)$ achieves total cost (35). Thus the MAP smoothing problem has been transformed into a deterministic optimal control problem. We can now bring any method for optimal control to bear on MAP smoothing. Of particular interest is the maximum principle – which can avoid the curse of dimensionality even when the posterior of the partially-observable system (34) does not have a finite set of sufficient statistics.

6.2 Duality of LQG control and Kalman smoothing

Let us now specialize these results to the LQG setting. Consider again the partially-observable system (29) discussed earlier. The posterior is Gaussian, therefore the MAP smoother and the Kalman smoother yield identical state estimates. The negative log-probabilities from (34) now become

$$\begin{aligned} a\left(\mathbf{x}_k, \mathbf{x}_{k-1}\right) &= \tfrac{1}{2}\left(\mathbf{x}_k - A\mathbf{x}_{k-1}\right)^{\mathsf{T}} S^{-1}\left(\mathbf{x}_k - A\mathbf{x}_{k-1}\right) + a_0 \\ b\left(\mathbf{y}_k, \mathbf{x}_k\right) &= \tfrac{1}{2}\left(\mathbf{y}_k - H\mathbf{x}_k\right)^{\mathsf{T}} P^{-1}\left(\mathbf{y}_k - H\mathbf{x}_k\right) + b_0 \\ c\left(\mathbf{x}_0\right) &= \tfrac{1}{2}\left(\mathbf{x}_0 - \widehat{\mathbf{x}}_0\right)^{\mathsf{T}} \Sigma_0^{-1}\left(\mathbf{x}_0 - \widehat{\mathbf{x}}_0\right) + c_0 \end{aligned}$$

where a_0, b_0, c_0 are normalization constants. Dropping all terms that do not depend on $\mathbf{x}$, and using the fact that $\mathbf{x}_k - A\mathbf{x}_{k-1} = \mathbf{u}_{k-1}$ from (36), the quantity (35) being minimized by the MAP smoother becomes

$$J\left(\mathbf{x}_\cdot, \mathbf{u}_\cdot\right) = \sum\nolimits_{k=0}^{n-1} \tfrac{1}{2}\mathbf{u}_k^{\mathsf{T}} R\mathbf{u}_k + \sum\nolimits_{k=0}^{n}\left(\tfrac{1}{2}\mathbf{x}_k^{\mathsf{T}} Q_k \mathbf{x}_k + \mathbf{x}_k^{\mathsf{T}}\mathbf{q}_k\right)$$

where

$$\begin{aligned} R &= S^{-1} \\ Q_0 &= \Sigma_0^{-1}, \quad \mathbf{q}_0 = -\widehat{\mathbf{x}}_0 \\ Q_k &= H^{\mathsf{T}} P^{-1} H, \quad \mathbf{q}_k = -H^{\mathsf{T}} P^{-1}\mathbf{y}_k, \quad 0 < k \leq n \end{aligned}$$

Thus the linear-Gaussian MAP smoothing problem is equivalent to a linear-quadratic optimal control problem. The linear cost term $\mathbf{x}_k^{\mathsf{T}}\mathbf{q}_k$ was not previously included in the LQG derivation, but it is straightforward to do so.

Let us now compare this result to the Kalman duality (33). Here the estimation system has dynamics $\mathbf{x}_{k+1} = A\mathbf{x}_k + \mathbf{w}_k$ and the control system has dynamics $\mathbf{x}_{k+1} = A\mathbf{x}_k + \mathbf{u}_k$. Thus the time-reversal and the matrix transpose of A are no longer needed. Furthermore the covariance matrices now appear inverted, and so we can directly relate costs to negative log-probabilities. Another difference is that in (33) we had $R \Longleftrightarrow P$ and $Q \Longleftrightarrow S$, while these two correspondences are now reversed.

A minimum-energy interpretation can help better understand the duality. From (29) we have $\mathbf{x}_k - A\mathbf{x}_{k-1} = \mathbf{w}_{k-1}$ and $\mathbf{y}_k - H\mathbf{x}_k = \mathbf{v}_k$. Thus the cost rate for the optimal control problem is of the form

$$\mathbf{w}^{\mathsf{T}} S^{-1}\mathbf{w} + \mathbf{v}^{\mathsf{T}} P^{-1}\mathbf{v}$$

This can be though of as the energy of the noise signals $\mathbf{w}$ and $\mathbf{v}$. Note that an estimate for the states implies estimates for the two noise terms, and the likelihood of the estimated noise is a natural quantity to optimize. The first term above measures how far the estimated state is from the prior; it represents a control cost because the control signal pushes the estimate away from the prior. The second term measures how far the predicted observation (and thus the estimated state) is from the actual observation. One can think of this as a minimum-energy tracking problem with reference trajectory specified by the observations.

7 Optimal control as a theory of biological movement

To say that the brain generates the best behavior in can, subject to the constraints imposed by the body and environment, is almost trivial. After all, the brain has evolved for the sole purpose of generating behavior advantageous to the organism. It is then reasonable to expect that, at least in natural and well-practiced tasks, the observed behavior will be close to optimal. This makes optimal control theory an appealing computational framework for studying the neural control of movement. Optimal control is also a very successful framework in terms of explaining the details of observed movement. However we have recently reviewed this literature [12] and will not repeat the review here. Instead we will briefly summarize existing optimal control models from a methodological perspective, and then list some research directions which we consider promising.

Most optimality models of biological movement assume deterministic dynamics and impose state constraints at different points in time. These constraints can for example specify the initial and final posture of the body in one step of locomotion, or the positions of a sequence of targets which the hand has to pass through. Since the constraints guarantee accurate execution of the task, there is no need for accuracy-related costs which specify what the task is. The only cost is a cost rate which specifies the "style" of the movement. It has been defined as (an approximation to) metabolic energy, or the squared derivative of acceleration (i.e. jerk), or the squared derivative of joint torque. The solution method is usually based on the maximum principle. Minimum-energy models are explicitly formulated as optimal control problems, while minimum-jerk and minimum-torque-change models are formulated in terms of trajectory optimization. However they can be easily transformed into optimal control problems by relating the derivative being minimized to a control signal.

Here is an example. Let $\mathbf{q}(t)$ be the vector of generalized coordinates (e.g. joint angles) for an articulated body such as the human arm. Let $\tau(t)$ be the vector of generalized forces (e.g. joint torques). The equations of motion are

$$\tau = M(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{n}(\mathbf{q}, \dot{\mathbf{q}})$$

where $M(\mathbf{q})$ is the configuration-dependent inertia matrix, and $\mathbf{n}(\mathbf{q}, \dot{\mathbf{q}})$ captures non-linear interaction forces, gravity, and any external force fields that depend on position or velocity. Unlike mechanical devices, the musculo-skeletal system has order higher than two because the muscle actuators have their own states. For simplicity assume that the torques τ correspond to the set of muscle activations, and have dynamics

$$\dot{\tau} = \tfrac{1}{c}(\mathbf{u} - \tau)$$

where $\mathbf{u}(t)$ is the control signal sent by the nervous system, and c is the muscle time constant (around 40 msec). The state vector of this system is

$$\mathbf{x} = [\mathbf{q};\ \dot{\mathbf{q}};\ \tau]$$

We will use the subscript notation $\mathbf{x}_{[1]} = \mathbf{q}$, $\mathbf{x}_{[2]} = \dot{\mathbf{q}}$, $\mathbf{x}_{[3]} = \tau$. The general first-order dynamics $\mathbf{x} = \mathbf{f}(\mathbf{x}, \mathbf{u})$ are given by

$$\begin{aligned}
\dot{\mathbf{x}}_{[1]} &= \mathbf{x}_{[2]} \\
\dot{\mathbf{x}}_{[2]} &= M\left(\mathbf{x}_{[1]}\right)^{-1}\left(\mathbf{x}_{[3]} - \mathbf{n}\left(\mathbf{x}_{[1]}, \mathbf{x}_{[2]}\right)\right) \\
\dot{\mathbf{x}}_{[3]} &= \tfrac{1}{c}\left(\mathbf{u} - \mathbf{x}_{[3]}\right)
\end{aligned}$$

Note that these dynamics are affine in the control signal and can be written as

$$\dot{\mathbf{x}} = \mathbf{a}(\mathbf{x}) + B\mathbf{u}$$

Now we can specify a desired movement time t_f, an initial state $\mathbf{x}_0$, and a final state $\mathbf{x}_f$. We can also specify a cost rate, such as

$$\begin{aligned}
&\text{control energy:} && \ell(\mathbf{x}, \mathbf{u}) = \tfrac{1}{2}\|\mathbf{u}\|^2 \\
&\text{torque-change:} && \ell(\mathbf{x}, \mathbf{u}) = \tfrac{1}{2}\|\dot{\tau}\|^2 = \tfrac{1}{2c^2}\left\|\mathbf{u} - \mathbf{x}^{[3]}\right\|^2
\end{aligned}$$

In both cases the cost is quadratic in $\mathbf{u}$ and the dynamics are affine in $\mathbf{u}$. Therefore the Hamiltonian can be minimized explicitly. Focusing on the minimum-energy model, we have

$$\begin{aligned}
H(\mathbf{x}, \mathbf{u}, \mathbf{p}) &= \tfrac{1}{2}\|\mathbf{u}\|^2 + (\mathbf{a}(\mathbf{x}) + B\mathbf{u})^\mathsf{T}\mathbf{p} \\
\pi(\mathbf{x}, \mathbf{p}) &= \arg\min_{\mathbf{u}} H(\mathbf{x}, \mathbf{u}, \mathbf{p}) = -B^\mathsf{T}\mathbf{p}
\end{aligned}$$

We can now apply the maximum principle, and obtain the ODE

$$\begin{aligned}
\dot{\mathbf{x}} &= \mathbf{a}(\mathbf{x}) - BB^\mathsf{T}\mathbf{p} \\
-\dot{\mathbf{p}} &= \mathbf{a}_\mathbf{x}(\mathbf{x})^\mathsf{T}\mathbf{p}
\end{aligned}$$

with boundary conditions $\mathbf{x}(0) = \mathbf{x}_0$ and $\mathbf{x}(t_f) = \mathbf{x}_f$. If instead of a terminal constraint we wish to specify a final cost $\mathbf{h}(\mathbf{x})$, then the boundary condition $\mathbf{x}(t_f) = \mathbf{x}_f$ is replaced with $\mathbf{p}(t_f) = \mathbf{h}_\mathbf{x}(\mathbf{x}(t_f))$. Either way we have as many scalar variables as boundary conditions, and the problem can be solved numerically using an ODE two-point boundary value solver. When a final cost is used the problem can also be solved using iterative LQG approximations.

Some optimal control models have considered stochastic dynamics, and used accuracy costs rather than state constraints to specify the task (state constraints cannot be enforced in a stochastic system). Such models have almost exclusively been formulated within the LQG setting. Control under sensory noise and delays has also been considered; in that case the model involves a sensory-motor loop composed of a Kalman filter and a linear-quadratic regulator. Of particular interest in stochastic models is control-multiplicative noise (also called signal-dependent noise). It is a well-established property of the motor system, and appears to be the reason for speed-accuracy trade-offs such as Fitts' law. Control-multiplicative noise can be formalized as

$$d\mathbf{x} = (\mathbf{a}(\mathbf{x}) + B\mathbf{u})\,dt + \sigma B D(\mathbf{u})\,d\mathbf{w}$$

where $D(\mathbf{u})$ is a diagonal matrix with the components of $\mathbf{u}$ on its main diagonal. In this system, each component of the control signal is polluted with Gaussian noise whose standard deviation is proportional to that component. The noise covariance is then

$$S = \sigma^2 B D(\mathbf{u}) D(\mathbf{u})^\mathsf{T} B^\mathsf{T}$$

With these definitions, one can verify that

$$\operatorname{tr}(SX) = \sigma^2 \operatorname{tr}\left(D(\mathbf{u})^\mathsf{T} B^\mathsf{T} X B D(\mathbf{u})\right) = \sigma^2 \mathbf{u}^\mathsf{T} B^\mathsf{T} X B \mathbf{u}$$

for any matrix X. Now suppose the cost rate is

$$\ell(\mathbf{x}, \mathbf{u}) = \tfrac{1}{2}\mathbf{u}^\mathsf{T} R \mathbf{u} + q(\mathbf{x}, t)$$

Then the Hamiltonian for this stochastic optimal control problem is

$$\tfrac{1}{2}\mathbf{u}^\mathsf{T}\left(R + \sigma^2 B^\mathsf{T} v_{\mathbf{xx}}(\mathbf{x}, t) B\right)\mathbf{u} + q(\mathbf{x}, t) + (\mathbf{a}(\mathbf{x}) + B\mathbf{u})^\mathsf{T} v_{\mathbf{x}}(\mathbf{x}, t)$$

If we think of the matrix $v_{\mathbf{xx}}(\mathbf{x}, t)$ as a given, the above expression is the Hamiltonian for a deterministic optimal control problem with cost rate in the same form as above, and modified control-energy weighting matrix:

$$\widetilde{R}(\mathbf{x}, t) = R + \sigma^2 B^\mathsf{T} v_{\mathbf{xx}}(\mathbf{x}, t) B$$

Thus, incorporating control-multiplicative noise in an optimal control problem is equivalent to increasing the control energy cost. The cost increase required to make the two problems equivalent is of course impossible to compute without first solving the stochastic problem (since it depends on the unknown optimal value function). Nevertheless this analysis affords some insight into the effects of such noise. Note that in the LQG case v is quadratic, its Hessian $v_{\mathbf{xx}}$ is constant, and so the optimal control law under control-multiplicative noise can be found in closed form.

7.1 Promising research directions

There are plenty of examples where motor behavior is found to be optimal under a reasonable cost function. Similarly, there are plenty of examples where perceptual judgements are found to be optimal under a reasonable prior. There is little doubt that many additional examples will accumulate over time, and reinforce the principle of optimal sensory-motor processing. But can we expect future developments that are conceptually novel? Here we summarize four under-explored research directions which may lead to such developments.

Motor learning and adaptation. Optimal control has been used to model behavior in well-practiced tasks where performance is already stable. But the processes of motor learning and adaptation – which are responsible for reaching stable performance – have rarely been modeled from the viewpoint of optimality. Such modeling should be straightforward given the numerous iterative algorithms for optimal controller design that exist.

Neural implementation of optimal control laws. Optimal control modeling has been restricted to the behavioral level of analysis; the control laws used to predict behavior are mathematical functions without an obvious neural implementation. In order to bridge the gap between behavior and single neurons, we will need realistic neural networks trained to mimic the input-output behavior of optimal control laws. Such networks will have to operate in closed loop with a simulated body.

Distributed and hierarchical control. Most existing models of movement control are monolithic. In contrast, the motor system is distributed and includes a number of anatomically distinct areas which presumably have distinct computational roles. To address this discrepancy, we have recently developed a hierarchical framework for approximately optimal control. In this framework, a low-level feedback controller transforms the musculoskeletal system into an augmented system for which high-level optimal control laws can be designed more efficiently [13].

Inverse optimal control. Optimal control models are presently constructed by guessing the cost function, obtaining the corresponding optimal control law, and comparing its predictions to experimental data. Ideally we would be able to do the opposite: record data, and automatically infer a cost function for which the observed behavior is optimal. There are reasons to believe that most sensible behaviors are optimal with respect to some cost function.

Recommended further reading

The mathematical ideas introduced in this chapter are developed in more depth in a number of well-written books. The standard reference on dynamic programming is Bertsekas [3]. Numerical approximations to dynamic programming, with emphasis on discrete state-action spaces, are introduced in Sutton and Barto [10] and formally described in Bertsekas and Tsitsiklis [2]. Discretization schemes for continuous stochastic optimal control problems are developed in Kushner and Dupuis [6]. The classic Bryson and Ho [5] remains one of the best treatments of the maximum principle and its applications (including applications to minimum-energy filters). A classic treatment of optimal estimation is Anderson and Moore [1]. A comprehensive text covering most aspects of continuous optimal control and estimation is Stengel [11]. The advanced subjects of non-smooth analysis and viscosity solutions are covered in Vinter [14]. The differential-geometric approach to mechanics and control (including optimal control) is developed in Bloch [4]. An intuitive yet rigorous introduction to stochastic calculus can be found in Oksendal [7]. The applications of optimal control theory to biological movement are reviewed in Todorov [12] and also in Pandy [8]. The links to motor neurophysiology are explored in Scott [9]. A hierarchical framework for optimal control is presented in Todorov et al [13].

Acknowledgements

This work was supported by US National Institutes of Health grant NS-045915, and US National Science Foundation grant ECS-0524761. Thanks to Weiwei Li for proofreading the manuscript.

References

[1] Anderson, B. and Moore, J. (1979) *Optimal Filtering.* Prentice Hall, Englewood Cliffs, NJ.

[2] Bertsekas, D. and Tsitsiklis, J. (1996) *Neuro-dynamic Programming.* Athena Scientific, Belmont, MA.

[3] Bertsekas, D. (2000) *Dynamic Programming and Optimal Control* (2nd ed). Athena Scientific, Belmont, MA.

[4] Bloch, A. (2003) *Nonholonomic mechanics and control.* Springer, New York.

[5] Bryson, A. and Ho, Y. (1969) *Applied Optimal Control.* Blaisdell Publishing, Walthman, MA.

[6] Kushner, H. and Dupuis, P. (2001) *Numerical Methods for Stochastic Control Problems in Continuous Time* (2nd ed). Springer.

[7] Oksendal, B. (1995) *Stochastic Differential Equations* (4th ed). Springer, Berlin.

[8] Pandy, M. (2001) Computer modeling and simulation of human movement. *Annual Review of Biomedical Engineering* **3**: 245-273.

[9] Scott, S. (2004) Optimal feedback control and the neural basis of volitional motor control. *Nature Reviews Neuroscience* **5**: 534-546.

[10] Sutton, R. and Barto, A. (1998) *Reinforcement Learning: An Introduction.* MIT Press, Cambdrige, MA.

[11] Stengel, R. (1994) *Optimal Control and Estimation.* Dover, New York.

[12] Todorov, E. (2004) Optimality principles in sensorimotor control. *Nature Neuroscience* **7**: 907-915.

[13] Todorov, E., Li, W. and Pan, X. (2005) From task parameters to motor synergies: A hierarchical framework for approximately optimal control of redundant manipulators. *Journal of Robotic Systems* **22**: 691-719.

[14] Vinter, R. (2000) *Optimal Control.* Birkhauser, Boston.