Discrete Time Stochastic Optimal Control

February 10, 2011

Please cite as

Movellan J. R. (2009) *Primer on Stochastic Optimal Control* MPLab Tutorials, University of California San Diego

1 Conventions

Unless otherwise stated, capital letters are used for random variables, small letters for specific values taken by random variables, and Greek letters for fixed parameters and important functions. We leave implicit the properties of the probability space $(\Omega, \mathcal{F}, P)$ in which the random variables are defined. Notation of the form $X \in \Re^n$ is shorthand for $X : \Omega \to \Re^2$, i.e., the random variable X takes values in $\Re^n$. We use E for expected values and Var for variance. When the context makes it clear, we identify probability functions by their arguments. For example $p(x, y)$ is shorthand for the joint probability mass or joint probability density that the random variable X takes the specific value x and the random variable Y takes the value y. Similarly $E[Y \mid x]$ is shorthand for the expected value of the variable Y given that the random variable X takes value x. We use subscripted colons to indicate sequences: e.g., $X_{1:t} \stackrel{\text{def}}{=} \{X_1 \cdots X_t\}$. Given a random variable X and a function f we use $df(X)/dX$ to represent a random variable that maps values of X into the derivative of f evaluated at the values taken by X. When safe we gloss over the distinction between discrete and continuous random variables. Unless stated otherwise, conversion from one to the other simply calls for the use of integrals and probability density functions instead of sums and probability mass functions.

Optimal policies are presented in terms of maximization of a reward function. Equivalently they could be presented as minimization of costs, by simply setting the cost function equal the reward with opposite sign. Below is a list of useful words and their equivalents

- Cost = - Value = - Reward = - Utility.= - Payoff
- The goal is to minimize Costs, or equivalent to maximize Value, Reward, Utility.
- We will use the terms Return and Performance to signify Cost or Value.
- Step = Stage
- One Step Cost = Running Cost
- Terminal cost = Bequest cost
- Policy = control law = controller = control
- Optimal n-step to go cost = optimal 1 step cost + optimal (n-1) step to go cost
- n-step to go cost given policy = 1 step cost given policy + (n-1) step to go cost given policy

2 Finite Horizon Problems

Consider a stochastic process $\{(X_t, , U_t, , C_t, R_t) : t = 1 : T\}$ where X_t is the state of the system, U_t actions, C_t the control law specific to time t, i.e., $U_t = C_t(X_t)$, and R_t a reward process (aka utility, cost, etc.). We use the convention that an action U_t is produced at time t after X_t is observed (see Figure 1). This results on a new state X_{t+1} and a reward R_t that can depend on X_t, U_t and on the future state X_{t+1}. This point of view has the disadvantage that the reward R_t "looks into the future", i.e., we need to know X_{t+1} to determine R_t. The advantage is that the approach is more natural for situations in which R_t depends only on X_t, U_t. In this special case R_t does not look into the future. In any case all the derivations work for the more general case in which the reward may depend on X_t, U_t, X_{t+1}.

Remark 2.1. Alternative Conventions In some cases it is useful to think of the action at time t to have an instantaneous effect on the state, which evolve at a longer time scale. This is equivalent to the convention adopted here but with the action shifted by one time step, i.e., U_t in our convention corresponds to U_{t-1} in the instantaneous action effect convention.

This section focuses on episodic problems of fixed length, i.e.., each episode starts at time 1 and ends at a fixed time $T \geq 1$.

Our goal is to find a control law $c_1, c_2, \cdots$ which maximizes a performance function of the following form

$$\rho(c_{1:T}) = E[\bar{R}_1 \mid c_{1:T}] \tag{1}$$

where

$$\bar{R}_t = \sum_{\tau=t}^{T} \alpha^{\tau-t} R_\tau, \;\; t = 1 \cdots T \tag{2}$$

i.e.,

$$\bar{R}_t = R_t + \alpha \bar{R}_{t+1} \tag{3}$$

When $\alpha \in [0, 1]$ it is called the *discount factor* because it tends to discount rewards that occur far into the future. If $\alpha > 1$ then future rewards become more important than present rewards. Note

We let the *optimal value function* Φ_t be defined as follows

$$\Phi_t(x_t) = \max_{c_t:T} E[\bar{R}_t \mid x_t, c_{t:T}] \tag{4}$$

In general this maximization problem is very difficult for it involves finding T jointly optimal functions. Fortunately, as we will see next, the problem decouples into solving T independent optimization problems.

Theorem 2.1 (Optimality Principle). *Let $\hat{c}_{t+1:T}$ be a policy that maximizes $E[\bar{R}_{t+1} \mid x_{t+1}, c_{t+1:T}]$ for all x_{t+1}, i.e.,*

$$E[\bar{R}_{t+1} \mid x_{t+1}, \hat{c}_{t+1:T}] = \max_{c_{t+1:T}} E[\bar{R}_{t+1} \mid x_{t+1}, c_{t+1:T}] \tag{5}$$

and let $\hat{c}_t(x_t)$ maximize $E[\bar{R}_t \mid x_t, c_t, \hat{c}_{t+1:T}]$ for all x_t with $\hat{c}_{t:T}$ fixed, i.e.,

$$E[\bar{R}_{t+1} \mid x_{t+1}, \hat{c}_t, \hat{c}_{t+1:T}] = \max_{c_t} E[\bar{R}_{t+1} \mid x_{t+1}, c_t, \hat{c}_{t+1:T}] \tag{6}$$

Then

$$E[\bar{R}_t \mid x_t, \hat{c}_{t:T}] = \max_{c_{t:T}} E[\bar{R}_t \mid x_t, c_{t:T}] \tag{7}$$

for all x_t

Proof.

$$\begin{aligned}\Phi_t(x_t) &= \max_{c_{t:T}} E[\bar{R}_t \mid x_t, c_{t:T}] = \max_{c_{t:T}} E[R_t + \alpha\bar{R}_{t+1} \mid x_t, c_{t:T}] \\ &= \max_{c_t} \left\{ E[R_t \mid x_t, c_t] + \alpha \max_{c_{t+1:T}} E[\bar{R}_{t+1} \mid x_t, c_{t:T}] \right\}\end{aligned} \tag{8}$$

where we used the fact that

$$E[R_t \mid x_t, c_{t:T}] = E[R_t \mid x_t, c_t] \tag{9}$$

which does not depend on $c_{t+1:T}$. Moreover,

$$\max_{c_{t+1:T}} E[\bar{R}_{t+1} \mid x_t, c_{t:T}] = \max_{c_{t+1:T}} \sum_{x_{t+1}} p(x_{t+1} \mid x_t, c_t)\, E[\bar{R}_{t+1} \mid x_{t+1}, c_{t+1:T}] \tag{10}$$

where we used the fact that

$$p(x_{t+1} \mid x_t, c_{t:T}) = p(x_{t+1} \mid x_t, c_t) \tag{11}$$

and

$$E[\bar{R}_{t+1} \mid x_t, c_t, x_{t+1}, c_{t+1:T}] = E[\bar{R}_{t+1} \mid x_{t+1}, c_{t+1:T}] \tag{12}$$

Using Lemma 8.1 and the fact that there is a policy $\hat{c}_{t+1:T}$ that maximizes $E[\bar{R}_{t+1}, x_{t+1}, c_{t+1:T}]$ for all x_{t+1} it follows that

$$\begin{aligned}\max_{c_{t+1:T}} E[\bar{R}_{t+1} \mid x_t, c_{t:T}] &= \max_{c_{t+1:T}} \sum_{x_{t+1}} p(x_{t+1} \mid x_t, c_t)\, E[\bar{R}_{t+1} \mid x_{t+1}, c_{t+1:T}] \\ &= \sum_{x_{t+1}} p(x_{t+1} \mid x_t, c_t) \max_{c_{t+1:T}} E[\bar{R}_{t+1} \mid x_{t+1}, c_{t+1:T}] \qquad (13) \\ &= \sum_{x_{t+1}} p(x_{t+1} \mid x_t, c_t)\, E[\bar{R}_{t+1} \mid x_{t+1}, \hat{c}_{t+1:T}] = E[\bar{R}_{t+1} \mid x_t, c_t, \hat{c}_{t+1:T}] \qquad (14)\end{aligned}$$

Thus we have that

$$\Phi_t(x_t) = \max_{c_{t:T}} E[\bar{R}_t \mid x_t, c_{t:T}] = \max_{c_t} \left(E[R_t \mid x_t, c_t] + \alpha E[\bar{R}_{t+1} \mid x_t, c_t, \hat{c}_{t+1:T}] \right) \tag{15}$$

□

Remark 2.2. The optimality principle suggests an optimal way for finding optimal policies: It is easy to find an optimal policy at terminal time T. For each state x_T such policy would choose an action that maximizes the terminal reward R_T, i.e.,

$$E[R_T \mid x_t, \hat{c}_T] = \max_{c_T} E[R_T \mid x_t, \hat{c}_T] \tag{16}$$

Provided we have an optimal policy for time $c_{t+1:T}$ we can leave it fixed and then optimize with respect to c_t. This allows to recursively compute an optimal policy starting at time T and finding our way down to time 1

The optimality principle leads to *Bellman Optimality Equation* which we state here as a corollary of the Optimality Principle

Corollary 2.1 (Bellman Optimality Equation)**.**

$$\Phi_t(x_t) = \max_{u_t} E[R_t + \alpha\, \Phi_{t+1}(X_{t+1}) \mid x_t, u_t] \tag{17}$$

for $t = 1 \cdots T$ *where*

$$E[\,\Phi_{t+1}(X_{t+1}) \mid x_t, u_t] = \sum_{x_{t+1}} p(x_{t+1} \mid x_t, u_t)\Phi_{t+1}(x_{t+1}) \tag{18}$$

and

$$\Phi_{T+1}(x) \stackrel{def}{=} 0, \quad \textit{for all } x \tag{19}$$

Proof. Obvious for $t = T$. For $t < T$ revisit equation (13) to get

$$\max_{c_{t+1:T}} E[\bar{R}_{t+1} \mid x_t, c_{t:T}] = \sum_{x_{t+1}} p(x_{t+1} \mid x_t, c_t) \max_{c_{t+1:T}} E[\bar{R}_{t+1} \mid x_{t+1}, c_{t+1:T}] \tag{20}$$

$$= \sum_{x_{t+1}} p(x_{t+1} \mid x_t, c_t)\, \Phi_{t+1}(x_{t+1}) = E[\Phi_{t+1}(X_{t+1}) \mid x_t, c_t] \tag{21}$$

Combining this with equation (8) completes the proof. □

Remark 2.3. It is useful to clarify the assumptions made to prove the optimality principle:

- Assumption 1:
$$E[R_t \mid x_t, c_{t:T}] = E[R_t \mid x_t, c_t] \tag{22}$$

- Assumption 2:
$$p(x_{t+1} \mid x_t, c_t, c_{t+1:T}) = p(x_{t+1} \mid x_t, c_t) \tag{23}$$

- Assumption 3:
$$E[\bar{R}_{t+1} \mid x_t, c_t, x_{t+1}, c_{t+1:T}] = E[\bar{R}_{t+1} \mid x_{t+1}, c_{t+1:T}] \tag{24}$$

- Assumption 4: Most importantly wee assumed that the optimal policy $\hat{c}_{t+1:T}$ did not impose any constraints on the set of policies c_t with respect to which we were performing the optimization. This would be violated, if there were an additional penalty or reward that depended directly on $c_{t:T}$. For example, this assumption would be violated if we were to force the policies of interest to be stationary. This would amount to putting a large penalty for policies that do not satisfy $c_1 = c_2 = \cdots c_{T-1}$.

Figure 1 displays a process that satisfies Assumptions 1-3. Note under the model the reward depends on the start state and the end state and the action. In addition we let the reward to depend on the control law itself. This allows, for example, to have the set of available actions depend on the current time and state.

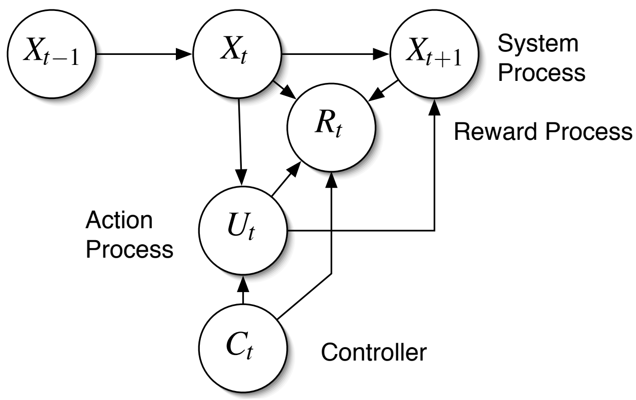

Figure 1: *Graphical Representation of the a time slice of a process satisfying the required assumptions. Arrows represent dependency relationships between variables.*

Remark 2.4. Note the derivations did not require to make the standard Markovian assumption, i.e.,

$$p(x_{t+1} \mid x_{1:t}, c_{1:t}) = p(x_{t+1} \mid x_t, c_t) \tag{25}$$

Remark 2.5. Consider now the case in which the admissible control laws are of the form

$$U_t = C_t(X_t) \in \mathcal{C}_t(X_t) \tag{26}$$

where $\mathcal{C}_t(x_t)$ is a set of available actions when visiting state x_t at time t. We can frame this problem by implicitly adding a large negative constant to the reward function when C_t chooses inadmissible actions. In this case the Bellman equation reduces to the following form

$$\Phi_t(x_t) = \max_{u_t \in \mathcal{C}_t(x_t)} E[R_t + \alpha\, \Phi_{t+1}(X_t) \mid x_t, u_t] \tag{27}$$

Remark 2.6. Now note that we could apply the restriction that the set of admissible actions at time t given x_t is exactly the action chosen by a given policy c_t. This leads to the *Bellman Equation for the Value of a given policy*

$$\Phi_t(x_t, c_{t:T}) = E[R_t + \alpha\Phi_{t+1}(X_{t+1}, c_{t+1:T}) \mid x_t, c_t] \tag{28}$$

where

$$\Phi_t(x_t, c_{t:T}) = E[\bar{R}_t \mid x_t, c_{t:T}] \tag{29}$$

is the value of visiting state x_t at time t given policy $c_{t:T}$.

Remark 2.7. Note that the Bellman equation cannot be used to solve the open loop control problem, i.e., restrict the set of allowable control laws to open loop laws. Such laws would be of the form

$$U_t = c_t(X_1) \tag{30}$$

which would violate *Assumption 4.* since

$$E[R_2 \mid x_2, c_2] \neq E[R_2 \mid x_1, x_2, c_{1:2}] \tag{31}$$

Remark 2.8 (Sutton and Barto (1998) : Reinforcement Learning, page 76 step leading to equation (3.14)). Since assuming stationary policies violates Assumption 4, this step is in Sutton and Barto's proof is not valid. The results are correct however, since for the infinite horizon case it is possible to prove Bellman's equation using other methods (see Bertsakas book, for example).

Remark 2.9. A problem of interest occurs when the set of possible control laws is a parameterized collection. For the general case such a problem will involve interdependencies between the different c_t, i.e., the constraints on C cannot be expressed as

$$\sum_{t=0}^{T} f_t(C_t) \tag{32}$$

which is required for **??** to work. For example, if $c_{1:T}$ is implemented as a feed-forward neural network parameterized by the weights w then would be stationary, i.e., $c_1 = c_2 = \cdots = c_T$. A constraint that cannot be expressed using (32). The problem can be approached by having time be one of the inputs to the model.

Example 2.1 (A simple Gambling Model (from Ross: Introduction to Dynamic Programming)). A gambler's goal is to maximize the log fortune after exactly T bets. The probability of winning on a bet is p. If winning the gambler gets twice the bet, if losing it loses the bet.

Let X_t represents the fortune after t bets, with initial condition $X_0 = x_0$.

$$R_t = \begin{cases} 0, & \text{for } t = 0, \cdots, n-1 \\ log(X_t), & \text{for } t = n \end{cases} \tag{33}$$

Let the action $U_t \in [0,1]$ represent a gamble of $U_t X_t$ dollars. Thus, using no discount factor $\alpha = 1$, Bellman's optimality equation takes the following form

$$\Phi_t(x_t) = \max_{0 \le u \le 1} E[\Phi_{t+1}(X_{t+1}) \mid x_t, u_t] \tag{34}$$
$$= \max_{0 \le u \le 1} \{p\, \Phi_{t+1}(x_t + u x_t) + (1-p)\, \Phi_{t+1}(x_t - u x_t)\} \tag{35}$$

with boundary condition

$$\Phi_T(x) = \log(x) \tag{36}$$

Thus

$$\Phi_{T-1}(x) = \max_{0 \le u \le 1} \{p \log(x + ux) + (1-p) \log(x - ux)\} \tag{37}$$
$$= \log(x) + \max_{0 \le u \le 1} \{p \log(1+u) + (1-p) \log(1-u)\} \tag{38}$$

Taking the derivative with respect to u and setting it to 0 we get

$$\frac{2p - 1 - u}{1 - u^2} = 0 \tag{39}$$

Thus

$$\hat{u}_{T-1}(x) = 2p - 1, \text{ provided } p > 0.5 \tag{40}$$
$$\Phi_{T-1}(x) = \log(x) + p\log(2p) + (1-p)\log(2(1-p)) = \log(x) + K \tag{41}$$

Thus, since K is a constant with respect to x, the optimal policy will be identical at time $T-2, T-3 ... 1$, i.e., the optimal gambling policy makes

$$U_t = (2p-1)X_t \tag{42}$$

provided $p \ge 0.5$. If $p < 0.5$ the optimal policy is to bet nothing.

3 The Linear Quadratic Regulator (LQR)

We are given a linear stochastic dynamical system

$$X_{t+1} = aX_t + bu_t + cZ_t \tag{43}$$
$$X_1 = x_1 \tag{44}$$

where $X_t \in \Re^n$, is the system's state, $a \in \Re^n \otimes \Re^n$, $u_t \in \Re^m$, $b \in \Re^n \otimes \Re^m$, $Z_t \in \Re^d$, $c \in \Re^n \otimes \Re^d$ where u_t is a control signal and Z_t are zero mean, independent random vectors with covariance equal to the identity matrix. Our goal is to find a a control sequence $u_{t:T} = u_t \cdots u_T$ that minimizes the following cost

$$R_t = X_t' q_t X_t + U_t' g_t U_t \tag{45}$$

where the state cost matrix q_t is **symmetric positive semi definite**, and the control cost matrix g_t is **symmetric positive definite**. Thus the goal is to keep the state X_t as close as possible to zero, while using small control signals. We define the value at time t of a state x_t given a policy π and terminal time $T \geq t$ as follows

$$\Phi_t(x_t, \pi) = \sum_{\tau=t}^{T} \gamma^{\tau-t} E[R_t \mid x_t, \pi] \tag{46}$$

3.1 Linear Policies: Policy Evaluation

We will consider first linear policies of the form $u_t = \theta_t x_t$, where θ_t is an $m \times n$ matrix. Thus the policies of interest are determined by T matrices $\theta_{1:T} = (\theta_1, \cdots, \theta_T)$. If we are interested on affine policies, we just need to augment the state X_t with a new dimension that is always constant. We will now show that the

We will now show, by induction, that the value $\Phi(x_t)$ of reaching state x_t at time t under policy $\phi_{t:T}$ is a quadratic function of the state[1], i.e.,

$$\Phi(x_t) = x_t' \alpha_t x_t + \beta_t \tag{47}$$

First note that since g is positive definite, the optimal control at time T is $\hat{u}_T = 0$. Thus $\hat{\theta}_t = 0$

$$\Phi_T(x_T) = x_T' q_T x_T = x_T' \alpha_T x_T + \beta_T \tag{48}$$

where

$$\alpha_T = q_T, \quad \beta_T = 0 \tag{49}$$

Assuming that

$$\Phi(x_{t+1}) = x_{t+1}' \alpha_{t+1} x_t + \beta_{t+1} \tag{50}$$

and applying Bellman's equation

$$\begin{aligned}
\Phi_t(x_t) &= x_t' q_t x_t + x_t' \theta_t' g_t \theta_t x_t && (51)\\
&\quad + \gamma E\big[\Phi_{t+1}(X_{t+1}) \mid x_t, \theta_{t+1:T}\big] + && (52)\\
&= x_t' q_t x_t + x_t' \theta_t' g_t \theta_t x_t && (53)\\
&\quad + \gamma E\big[\Phi_{t+1}(a x_t + b\theta_t u_t + c Z_t,) \mid x_t, \theta_{t+1:T}\big] + && (54)\\
&= x_t' q_t x_t + x_t' \theta_t' g_t \theta_t x_t && (55)\\
&\quad + \gamma (a x_t + b\theta_t u_t)' \alpha_{t+1} (a x_t + b\theta_t u_t) + \gamma \mathrm{Tr}(c' \alpha_{t+1} c) + \beta_{t+1} && (56)
\end{aligned}$$

where we used the fact that $E[Z_{t,i} Z_{t,j} \mid x_t, u_t] = \delta_{i,j}$ and therefore

$$\begin{aligned}
E\big[Z_t' c' \alpha_{t+1} c Z_t \mid x_t, u_t\big] &= \sum_{ij} (c' \alpha_{t+1} c)_{ij} E\big[Z_{ti} Z_{tj}\big] && (57)\\
&= \sum_{i} (c' \alpha_{t+1} c)_{ij} = \mathrm{Tr}(c' \alpha_{t+1} c) && (58)
\end{aligned}$$

[1] To avoid clutter we leave implicit the dependency of Φ on t and θ

Thus

$$\Phi_t(x_t) = x_t'\Big(q_t + \theta_t' g_t \theta_t + + \gamma(a_t + b_t\theta_t)'\alpha_{t+1}(a_t + b_t\theta_t)\Big)x_t \tag{59}$$
$$+ \gamma \mathrm{Tr}(c_t'\alpha_{t+1}c_t) + \beta_{t+1} \tag{60}$$

Thus

$$\Phi(x_t) = x_t'\alpha_t x_t + \beta_t \tag{61}$$

where

$$\alpha_t = q_t + \theta_t' g_t \theta_t + \gamma(a_t + b_t\theta_t)'\alpha_{t+1}(a_t + b_t\theta_t) \tag{62}$$
$$= \theta_t'(g_t + \gamma b_t'\alpha_{t+1}\beta_t)\theta_t + q_t + \gamma a_t'\alpha_{t+1}a_t \tag{63}$$
$$\beta_t = \mathrm{Tr}(c_t'\alpha_{t+1}c_t) + \beta_{t+1} \tag{64}$$

3.2 Linear Policies: Policy Improvement

Taking the gradient with respect to θ_t of the state value

$$\nabla_{vec[\theta_t]}\Phi(x_t) = \nabla_{vec[\theta_t]} x_t'\alpha_t x_t + \beta_t \tag{65}$$
$$= \nabla_{\theta_t}(\theta_t x_t)' g_t(\theta x_t) + \gamma \nabla_{\theta_t} x_t'\Big((a_t + b_t\theta_t)'\alpha_{t+1}(a_t + b_t\theta_t)\Big)x_t \tag{66}$$

Note

$$\nabla_{vec[\theta]}(\theta_t x)' g_t(\theta x) = \nabla_{vec[\theta]}\theta x \; \nabla_{\theta x}(\theta_t x)' g_t(\theta x) = x \otimes I \mathrm{vec}[\theta x] \tag{67}$$

Thus

$$\nabla_\theta(\theta_t x)' g_t(\theta x) = g_t \theta x x' \tag{68}$$

Moreover

$$\nabla_{vec[\theta]} \; x'(a + b\theta)'\alpha(a + b\theta)x = \nabla_{vec[\theta]} \; (a + b\theta)x \tag{69}$$
$$\nabla_{(a+b\theta)x} x'(a + b\theta)'\alpha(a + b\theta)x \tag{70}$$
$$= x \otimes b' \mathrm{vec}[\alpha(a + b\theta)x] \tag{71}$$

Thus

$$\nabla_\theta \; x'(a + b\theta)'\alpha(a + b\theta)x = b'\alpha(a + b\theta)xx' \tag{72}$$

and

$$\nabla_{\theta_t}\Phi(x_t) = \Big(g_t\theta_t + \gamma b_t'\alpha_{t+1}(a_t + b_t\theta_t)\Big)x_t x_t' \tag{73}$$

We can thus improve the policy by performing gradient ascent

$$\theta_t \leftarrow \theta_t + \epsilon\Big(g_t\theta_t + \gamma b_t'\alpha_{t+1}(a_t + b_t\theta_t)\Big)x_t x_t' \tag{74}$$

This gradient approach is useful for adaptive approaches to non-stationary problems and for iterative approaches to solve non-linear control problems via linearizations.

The optimal value of θ_t can also be found by setting the gradient to zero and solving the resulting algebraic equation. Note for

$$\hat{\theta}_t = -\gamma\Big(g_t + \gamma b_t'\alpha_{t+1}b_t)\Big)^{-1} b_t'\alpha_{t+1}a_t \tag{75}$$

then

$$\nabla_{\theta_t}\Phi(x_t) = 0, \text{ for all } x_t \tag{76}$$

Note also for $\hat{\theta}_t$ then α_t simplifies as follows

$$\alpha_t = \hat{\theta}_t'(g_t + \gamma b_t'\alpha_{t+1}\beta_t)\hat{\theta}_t + q_t + \gamma a_t'\alpha_{t+1}a_t \tag{77}$$

$$= -\gamma\hat{\theta}_t'b_t'\alpha_{t+1}a_t + q_t + \gamma a_t'\alpha_{t+1}a_t \tag{78}$$

$$\alpha_t = q_t + \gamma(\gamma a_t' - \hat{\theta}_t'b_t')\alpha_{t+1}a_t \tag{79}$$

3.3 Optimal Unconstrained Policies

Here we show that in fact the optimal policy is linear, so a linearity constraint turns out not to be a constraint in this case and the results above produce the optimal policy. The proof works by induction. We note that for the optimal policy

$$\Phi(x_T) = x_T'\alpha_t x_T + \beta_T \tag{80}$$

and if

$$\Phi(x_{t+1}) = x_{t+1}'\alpha_{t+1}x_{t+1} + \beta_{t+1} \tag{81}$$

then, applying the Bellman Equations

$$\Phi(x_t) = \min_{u_t} x_t'q_t x_t + u_t'g_t u_t \tag{82}$$

$$+ \gamma(ax_t + bu_t)'\alpha_{t+1}(ax_t + bu_t) + \gamma\mathrm{Tr}(c'\alpha_{t+1}c) + \beta_{t+1} \tag{83}$$

Taking the gradient with respect to u_t in a manner similar to how we did above for θ_t we get

$$\nabla_{u_t}\Phi(x_t) = g_t u_t + b'\alpha_{t+1}(ax_t + bu_t) \tag{84}$$

Setting the gradient to zero we get the optimal u_t

$$\hat{u}_t = \theta_t x_t \tag{85}$$

$$\theta_t \stackrel{\text{def}}{=} -(g_t + b'\alpha_{t+1}b)^{-1}b'\alpha_{t+1}a \tag{86}$$

which is a linear policy.

3.4 Summary of Equations for Optimal Policy

Let

$$\alpha_T = q_T \tag{87}$$
$$\hat{u}_T = 0 \tag{88}$$

then move your way from $t = T - 1$ to $t = 1$ using the following recursion

$$K_t = (b'\alpha_{t+1}b + g_t)^{-1}b'\alpha_{t+1}a \tag{89}$$
$$\alpha_t = q_t + a'\alpha_{t+1}(a - bK_t) \tag{90}$$

and the optimal action at time t is given by

$$\hat{u}_t = -K_t x_t \tag{91}$$
$$\tag{92}$$

If desired, the value function can be obtained as follows

$$\Phi_t(x_t) = x_t'\alpha_t x_t + \gamma_t \tag{93}$$

where

$$\gamma_t = \gamma_{t+1} + \mathrm{Tr}(c'\alpha_{t+1}c) \tag{94}$$

Below is Matlab code

```
% X_{t+1} = a _t X_t + b u_t + c Z_t
% R_t = X_t' q_t X_t + U_t' g_t U_t

function gain = lqr(a, b, c, q,g,T)
alpha{T} = q{T};
beta{T}=0;
for t = T-1:-1:1
  gain{t} = inv(b'*alpha{t+1}* b + g{t} )*b'*alpha{t+1}*a;
  alpha{t} = q{t}+ a'*alpha{t+1}*(a - b*gain{t});
  beta{t} = beta{t+1}+ trace(c'*alpha{t+1} *c);
end
```

Remark 3.1. The dispersion matrix c has no effect on the optimal control signal, it only affects the expected payoff given the optimal control.

Remark 3.2. Note the optimal action at time t is an error term ax_t premultiplied by a gain term K_t. The gain term K_t and the targets μ_t do not depend on $x_{1:T}$ and thus only need to be computed once.

Remark 3.3. Note K_t in (**??**) is the ridge regression solution to the problem of predicting b using a. The error of that prediction $a - bK_t$ appears in the Riccati equation (**??**)

Remark 3.4. Suppose the cost function is of the form

$$R_t = (X_t - \xi_t)' q_t (X_t - \xi_t) + U_t' g_t U_t \tag{95}$$

where $\xi_{1:T}$ is a desired sequence of states. We can handle this case by augmenting the system as follows

$$\tilde{X}_{t+1} = \tilde{a} X_t + \tilde{b} u_t + \tilde{c} Z_t \tag{96}$$

where

$$\tilde{X}_t = \begin{pmatrix} X_t \\ \xi_t \\ 1 \end{pmatrix} \in \Re^{2n} \tag{97}$$

$$\tilde{a} = \begin{pmatrix} a_{n\times n} & 0_{n\times n} & 0_{n\times 1} \\ 0_{n\times n} & 0_{n\times n} & \Delta\xi_t \\ 0_{1\times n} & 0_{1\times n} & 1 \end{pmatrix} \in \Re^{2n+1} \otimes \Re^{2n+1} \tag{98}$$

$$\tilde{b} = \begin{pmatrix} b_{n\times m} \\ 0_{n\times m} \\ 0_{1\times m} \end{pmatrix} \in \Re^{2n+1} \otimes \Re^{m} \tag{99}$$

$$\tilde{c} = \begin{pmatrix} c_{n\times d} \\ 0_{n\times d} \\ 0_{1\times d} \end{pmatrix} \in \Re^{2n+1} \otimes \Re^{d} \tag{100}$$

$$\tag{101}$$

$$\tag{102}$$

where $\Delta\xi_t \stackrel{\text{def}}{=} \xi_{t+1} - \xi_t$ and we use subscripts as a reminder of the dimensionality of matrices. The return function is now strictly quadratic on the extended state space

$$\tilde{R}_t = \tilde{X}_t' \tilde{q}_t \tilde{X}_t + U_t' g_t U_t \tag{103}$$

where

$$\tilde{q}_t = \begin{pmatrix} q_t & -q_t & 0_{n\times 1} \\ -q_t & q_t & 0_{n\times 1} \\ 0_{1\times n} & 0_{1\times n} & 0 \end{pmatrix} \in \Re^{2n+1} \otimes \Re^{2n+1} \tag{104}$$

3.5 Example

Consider the simple case in which

$$X_{t+1} = a X_t + u_t + c Z_t \tag{105}$$

at time t we are at x_t and we want to get as close to zero as possible at the next time step. There is no cost for the size of the control signal. In this case $b = I$,

$q_t = I$, $q_t = 0$, $g_t = 0$, $\xi_t = 0$. Thus we have

$$\mu_T = 0 \tag{106}$$
$$\alpha_T = I \tag{107}$$
$$\hat{u}_T = 0 \tag{108}$$
$$\tag{109}$$

$$K_{T-1} = \alpha_T = I \tag{110}$$
$$\kappa_{T-1} = 0 \tag{111}$$
$$\alpha_{T-1} = I \tag{112}$$
$$\mu_{T-1} = 0 \tag{113}$$
$$\tag{114}$$

from which it follows that

$$\epsilon_t = I, \tag{115}$$
$$\hat{u}_t = -a x_t, \text{ for } t = 1 \cdots T-1 \tag{116}$$

In this case all the controller does is to anticipate the most likely next state (i.e., ax) and compensates for it accordingly so that the expected value at the next time step is zero.

3.6 Example: Controlling a mass subject to random forces

Consider a particle with point mass m located at x_t with velocity v_t subject to a constant force $f_t = m\, u_t$ for the period $[t, t+\Delta_t]$. Using the equations of motion. For $\tau \in [0, \Delta_t]$ we have that

$$v_{t+\tau} = v_t + \int_0^\tau u_t ds = v_t + u_t \tau \tag{117}$$
$$x_{t+\Delta t} = x_t + \int_0^{\Delta_t} v_{t+s} ds = x_t + v_t \Delta_t + u_t \frac{\Delta_t^2}{2} \tag{118}$$

or in matrix form

$$\begin{pmatrix} x_{t+\Delta_t} \\ v_{t+\Delta_t} \end{pmatrix} = \begin{pmatrix} 1 & \Delta_t \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x_t \\ v_t \end{pmatrix} + \begin{pmatrix} \frac{\Delta_t^2}{2} \\ \Delta_t \end{pmatrix} u_t \tag{119}$$

We can add a drag force proportional to v_t and constant through the period $[v_t, v_t + \Delta_t]$ and a random force constant through the same period

$$\begin{pmatrix} x_{t+\Delta_t} \\ v_{t+\Delta_t} \end{pmatrix} = \begin{pmatrix} 1 & \Delta_t - \epsilon\Delta_t^2/2 \\ 0 & 1-\epsilon\Delta_t \end{pmatrix} \begin{pmatrix} x_t \\ v_t \end{pmatrix} + \begin{pmatrix} \frac{\Delta_t^2}{2} \\ \Delta_t \end{pmatrix} u_t + \begin{pmatrix} 0 & \sigma\Delta_t^2/2 \\ 0 & \sigma\Delta_t \end{pmatrix} \begin{pmatrix} Z_{1,t} \\ Z_{2,t} \end{pmatrix} \tag{120}$$

We can express this as a 2-dimensional discrete time system

$$\tilde{x}_{t+1} = a\tilde{x}_t + bu_t + cZ_t \tag{121}$$

where

$$\tilde{x}_t = \begin{pmatrix} x_t \\ v_t \end{pmatrix}, \quad a = \begin{pmatrix} 1 & \Delta_t - \epsilon\Delta_t^2/2 \\ 0 & 1-\epsilon\Delta_t \end{pmatrix}, \quad b = \begin{pmatrix} \frac{\Delta_t^2}{2} \\ \Delta_t \end{pmatrix}, \quad c = \begin{pmatrix} 0 & \sigma\Delta_t^2/2 \\ 0 & \sigma\Delta_t \end{pmatrix} \tag{122}$$

And solve for the problem of finding an optimal application of forces to keep the system at a desired location and/or velocity while minimizing energy consumption.

Figure 2 shows results of a simulation (Matlab Code Available) for a point mass moving along a line. The mass is located at -10 at time zero. There is a constant quadratic cost for applying a force at every time step, and a large quadratic at the terminal time (goal is to be at the origin with zero velocity by 10 seconds). Note the inverted U shape of the obtained velocity. Also note the system applies a positive force during the first half of the run and then a negative force (brakes) increasingly larger as we get close to the desired location. Note this would have been hard to do with a standard proportional controller (a change of sign in the applied force from positive early on to negative as we get close to the objective.

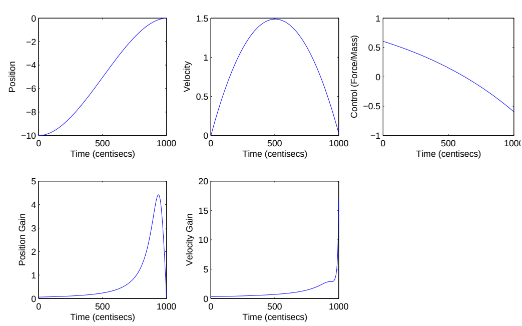

Figure 2:

4 Infinite Horizon Case

As $T \to \infty$ and under rather mild conditions α_t becomes stationary and satisfies the stationary version of (90)

$$\alpha = q + a'\alpha(a - b(b'\alpha b + g)^{-1}b'\alpha a) \tag{123}$$

The stationary control function

$$u_t = -Kx_t \tag{124}$$
$$K = (b'\alpha b + g)^{-1} b'\alpha a \tag{125}$$

minimizes the stationary cost

$$\rho = \lim_{t\to\infty} \frac{1}{T} \sum_{t=1}^{T} E[X_t' q X_t + U_t' g U_t] \tag{126}$$

Regarding β, given the definition of ρ

$$\beta_t = \frac{t-1}{t}\beta_{t-1} + \frac{1}{t}\mathrm{Tr}(c'\alpha_t c) \tag{127}$$

and in the stationary case

$$\beta = \frac{t-1}{t}\beta + \frac{1}{t}\mathrm{Tr}(c'\alpha c) \tag{128}$$
$$\beta = \mathrm{Tr}(c'\alpha c) \tag{129}$$

Thus the stationary value of state x_t is

$$\Phi(x_t) = x_t'\alpha x_t + \mathrm{Tr}(c'\alpha c) \tag{130}$$

4.1 Example

We want to control

$$X_{t+1} = X_t + U_t + Z_t \tag{131}$$

where $U_t = -KX_t$. In Matlab, the algebraic Riccati equation can be solved using the function "dare" (discrete algebraic riccati equation).

We enter

$$(alpha, L, K) = dare(a, b, q, g, 0, 1) \tag{132}$$

For $q = 1, g = 0$ we get $K = 1$, i.e, if there is no action cost the best thing to do is to produce an action equal to the current state but with the oposite sign. For $q = 1, g = 10$ we get $K = 0.27$, i.e., we need to reduce the gain of our response.

5 Feedback Linearization

Proposition 5.1. *Consider a process of the form*

$$X_{t+1} = aX_t + b\, f(X_t, U_t, t) + cZ_t \tag{133}$$

where a, b are fixed matrices, U_t is a control variable and f is a function such that for every x, t the mapping between U_t and $f(X_t, U_t, t)$ is bijective, i.e. there is a function h such that for every x, y, t

$$h(x, f(x, u, t), t) = u \tag{134}$$

Let the instantaneous cost function take the following form

$$R_t = X_t' q_t X_t + f(X_t, U_t, t)' g_t f(X_t, U_t, t) \tag{135}$$

Then the following policy is optimal:

$$U_t = h(X_t, Y_t, t) \tag{136}$$

where Y_t is the solution to the following LQR control problem

$$X_{t+1} = aX_t + bY_t + cZ_t \tag{137}$$

Proof. Let the control process U be defined as follows

$$U_t = \pi(X_t, t) \tag{138}$$

where π is a control policy. Let the virtual control process Y be defined a follows

$$Y_t = \lambda(X_t, t) = f(X_t, U_t, t) \tag{139}$$

Note π and λ are not independent: For every policy π there is one equivalent policy λ. Moreover for every virtual policy λ there is an equivalent policy π

$$U_t = \pi(X_t, t) = h(X_t, \lambda(X_t, t), t) \tag{140}$$

We note that when expressed in terms of the Y variables, the control problem is linear quadratic

$$X_{t+1} = aX_t + bY_t + cZ_t \tag{141}$$

$$R_t = X_t' q_t X_t + Y_t' g_t Y_t \tag{142}$$

Let $\hat{\lambda}$ be the optimal policy mapping states to virtual actions, as found using the standard LQR algorithm on (141), (142). Let

$$\hat{\pi}(X_t, t) = h(X_t, \lambda(X_t, Y_t, t), t) \tag{143}$$

Suppose there is a policy π^* mapping states to actions better than $\hat{\pi}$. Thus the policy

$$\lambda^*(X_t, t) = f(X_t, \pi^*(X_t, t), t) \tag{144}$$

should be better than $\hat{\lambda}$, which is a contradition. □

This is a remarkable result. It let's us solve optimally a non-linear control problem. The key is that we lose control over the action penalty term. Rather than having the penalty be quadratic with respect to the actions U_t, which could be things like motor torques, we have to use a penalty quadratic with respect to $f(X_t, U_t, t)$.

6 Partially Observable Processes

Consider a stochastic process $\{(X_t, Y_t, U_t, C_t) : t = 1 : T\}$ where X_t represents a hidden state, Y_t observable states, and U_t actions.We use the convention that the action at time t is produced after Y_t is observed. This action is determined by a controller C_t whose input is $Y_{1:t}, U_{1:t-1}$, i.e., the information observed up to to time t, and whose output the action at time t, i.e.,

$$U_t = C_t(O_t) \tag{145}$$

$$O_t = \begin{pmatrix} Y_{1:t} \\ U_{1:t-1} \end{pmatrix} \tag{146}$$

Figure 3 display Markovian constraints in the joint distribution of the different variables involved in the model. An arrow from variable X to variable Y indicates that X is a "parent" of Y. The probability of a random variable is conditionally independent of all the other variables given the parent variables. Dotted figures indicate unobservable variables, continuous figures indicate observable variables. Under these constraints, the process is defined by an initial distribution for the hidden states

$$X_1 \sim \nu \tag{147}$$

a sensor model

$$p(y_t \mid x_t, u_{t-1}) \tag{148}$$

and state dynamics model

$$p(x_{t+1} \mid x_t, u_t) \tag{149}$$

Remark 6.1. Alternative Conventions Under our convention effect of actions is not instantaneous, i.e, the action at time $t-1$ affects the state and the observation at time $t+1$. In some cases it is useful to think of the effect of actions occurring at a shorter time scales than the state dynamics. In such cases it may be useful to model the distribution of observations at time t as being determined by the state and action at time t. Under this convention, U_t corresponds to what we call U_{t+1} (See Right Side of Figure 3).

It may also be useful to think of the X_t generates Y_t, which is used by the controller C_t to generate U_t.

We will make our goal to find a controller that optimizes a performance function:

$$\rho(c_{1:T}) = E[\bar{R}_1 \mid c_{1:T}] \tag{150}$$

where

$$\bar{R}_t = \sum_{\tau=t}^{T} \alpha^{\tau-t} R_\tau,\ t = 1 \cdots T \tag{151}$$

The controller maps the information state at time t into actions.

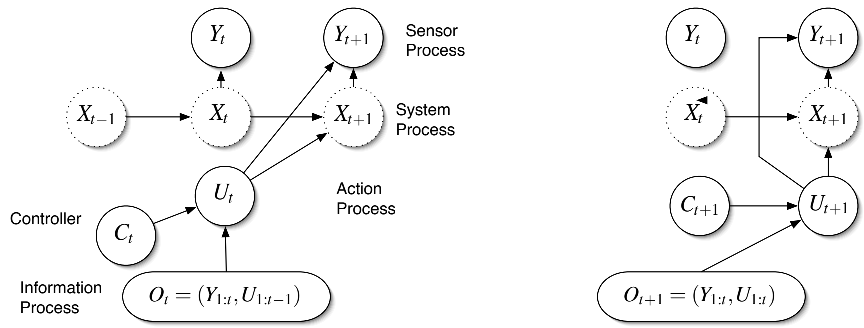

Figure 3: **Left:** *The convention adopted in this document. Arrows represent dependency relationships between variables. Dotted figures indicate unobservable variables, continuous figures indicate observable variables. Under this convention the effect of actions is not instantaneous.* **Right:** *Alternative convention. Under this convention the effect of actions is instantaneous.*

6.1 Equivalence with Fully Observable Case

- Assumption 1:
$$E[R_t \mid o_t, c_{t:T}] = E[R_t \mid o_t, c_t] \tag{152}$$

- Assumption 2:
$$p(o_{t+1} \mid o_t, c_t, c_{t+1:T}) = p(o_{t+1} \mid o_t, c_t) \tag{153}$$

- Assumption 3:
$$E[\bar{R}_{t+1} \mid o_t, c_t, o_{t+1}, c_{t+1:T}] = E[\bar{R}_{t+1} \mid o_{t+1}, c_{t+1:T}] \tag{154}$$

Remark 6.2. The catch is that the number of states to represent the observable process grows exponentially with time. For example, if we have binary observations and actions, the number of possible states by time t is 4^t. Thus it is critical to summarize all the available information.

Remark 6.3. Open Loop Policies We can model open loop processes as special cases of partially observable control processes. In such cases the state at time 1 but thereafter the observation process is uninformative (e.g., it could be a constant).

6.2 Sufficient Statistics

A critical problem for the previous algorithm is that it requires us to keep track of all possible sequences $y_{1:T}, u_{1:T}$, which grow exponentially as a function of T. This issue can be sometimes addressed if all the relevant information about the sequence $y_{1:t}, u_{1:t-1}$ can be described in terms of a summary statistic S_t which can be computed in a recursive manner. In particular we need for S_t to have the following assumption: *Seems like some of these assumptions may be redundant. Clarify where they are used.*

- Assumption 1:
$$S_t = f_t(O_t) \tag{155}$$

- Assumption 2:
$$S_{t+1} = g_t(S_t, Y_{t+1}, U_t) \tag{156}$$

- Assumption 3:
$$E[R_t \mid o_t, u_t] = E[R_t \mid s_t, u_t] \tag{157}$$

- Assumption 4:
$$p(y_{t+1} \mid o_t, u_t) = p(y_{t+1} \mid s_t, u_t) \tag{158}$$

where f_t are known functions. Note

$$\Phi_T(o_T) = E[R_T \mid o_t] = E[R_T \mid s_t] = \tilde{\Phi}_T(s_T) \tag{159}$$

and thus the optimal value function, and the optimal action at time T depend only on s_T. We will now show that if the optimal value function at time $t+1$ is a function of s_{t+1}, i.e., $\Phi_{t+1}(o_{t+1}) = \tilde{\Phi}_{t+1}(f_{t+1}(o_{t+1}))$ then the optimal action and optimal value function at time t are a function of s_t

$$\Phi_t(o_t) = \min_{u_t} E[R_t + \alpha \Phi_{t+1}(O_{t+1}) \mid o_t, u_t] \tag{160}$$

$$= \min_{u_t} \left\{ E[R_t \mid s_t, u_t] + \alpha \sum_{y_{t+1}} p(y_{t+1} \mid o_t, u_t) \Phi_{t+1}(o_t, u_t, y_{t+1}) \right\} \tag{161}$$

$$= \min_{u_t} \left\{ E[R_t \mid s_t, u_t] + \alpha \sum_{y_{t+1}} p(y_{t+1} \mid s_t, u_t) \tilde{\Phi}_{t+1}(g_t(s_t, y_{t+1}, u_t)) \right\} \tag{162}$$

$$= \min_{u_t} \left\{ E[R_t \mid s_t, u_t] + \alpha \sum_{s_{t+1}} p(s_{t+1} \mid s_t, u_t) \tilde{\Phi}_{t+1}(s_{t+1}) \right\} \tag{163}$$

$$= \min_{u_t} \left\{ E[R_t + \alpha \tilde{\Phi}_{t+1}(S_{t+1}) \mid s_t, u_t] \right\} \stackrel{\text{def}}{=} \tilde{\Phi}_t(s_t) \tag{164}$$

Thus, we only need to keep track of s_t to find the optimal policy with respect to o_t.

6.3 The Posterior State Distribution as a Sufficient Statistic

Consider the statistic $S_t = p_{X_t \mid O_t}$, i.e., the entire posterior distribution of states given the observed sequence up to time t. First note

$$S_1(x_1) = p(x_1 \mid Y_1) = f_1(Y_1) \text{ for all } x_1 \tag{165}$$
$$\tag{166}$$

Moreover that the update of the posterior distribution only requires the current posterior distribution, which becomes a prior, and the new action and observation

$$p(x_{t+1} \mid y_{1:t+1}, u_{1:t}) \propto \sum_{x_t} p(x_t \mid y_{1:t}, u_{1:t}) p(x_{t+1} \mid x_t, u_t) p(y_{t+1} \mid x_t) \tag{167}$$

which satisfies the second assumption.

$$E[R_t \mid o_t, u_t] = \sum_{x_t} p(x_t \mid o_t, u_t) R_t(x_t, u_t) = E[R_t \mid s_t, u_t] \tag{168}$$

and

$$p(y_{t+1} \mid y_{1:t}, u_{1:t}) = \sum_{x_{t+1}} p(x_t \mid y_{1:t}, u_{1:t-1}) p(x_{t+1} \mid x_t, u_t) p(y_{t+1} \mid x_{t+1}) \tag{169}$$

6.4 Limited Memory States (Under Construction)

What if we want to make a controller that uses a particular variable at time t as its only source of information and this variable may not necessarily be a sufficient statistic of all the past observations. My current thinking is that the optimality equation will hold, but computation of the necessary distributions may be hard and require sampling.

7 Linear Quadratic Gaussian (LQG)

The LQG problem is the partially observable version of LQR. We are given a linear stochastic dynamical system

$$X_{t+1} = aX_t + bu_t + cZ_t \tag{170}$$
$$Y_{t+1} = kX_{t+1} + mW_{t+1} \tag{171}$$
$$X_1 \sim \nu_1 \tag{172}$$

where $X_t \in \Re^n$, is the system's state, $a \in \Re^n \otimes \Re^n$, $u_t \in \Re^m$, $b \in \Re^n \otimes \Re^m$, $Z_t \in \Re^d$, $c \in \Re^n \otimes \Re^d$ where u_t is a control signal and Z_t are zero mean, independent random vectors with covariance equal to the identity matrix. Our goal is to find a a control sequence $u_{t:T} = u_t \cdots u_T$ that minimizes the following cost

$$R_t = X_t' q_t X_t + U_t' g_t U_t \tag{173}$$

where the state cost matrix q_t is **symmetric positive semi definite**, and the control cost matrix g_t is **symmetric positive definite**. Thus the goal is to keep the state X_t as close as possible to zero, while using small control signals. Let

$$O_t \stackrel{\text{def}}{=} \begin{pmatrix} Y_{1:t} \\ U_{1:t-1} \end{pmatrix} \tag{174}$$

represent the information available at time t. We will solve the problem by assuming that the optimal *cost* is of the form

$$\Phi_t(o_t) = E[X_t' \alpha_t X_t \mid o_t] + \beta_t(o_t) \tag{175}$$

where $\beta_t(o_t)$ is constant with respect to $t-1$, and then proving by induction that the assumption is correct.

First note since g is positive definite, the optimal control at time T is $\hat{u}_T = 0$. Thus

$$\Phi_T(o_T) = E[X_T' q_T X_T \mid o_T] = E[X_T' \alpha_T X_T \mid o_T] + \beta_T(o_T) \tag{176}$$

and our assumption is correct for the terminal time T with

$$\alpha_T = q_T, \quad \beta_T(o_T) = 0 \tag{177}$$

Assuming (175) is correct at time $t+1$ and applying Bellman's equation

$$\Phi_t(o_t) = E[X_t' q_t X_t \mid o_t] + \min_{u_t} E\big[\Phi_{t+1}(O_{t+1}) + u_t' g_t u_t \mid o_t, u_t\big] \tag{178}$$

$$= E[X_t' q_t X_t \mid o_t] + E[\beta_{t+1}(O_{t+1}) \mid o_t, u_t]$$
$$+ \min_{u_t} E\big[(aX_t + bu_t + cZ_t)' \alpha_{t+1} (aX_t + bu_t + cZ_t) + u_t' g_t u_t \mid o_t, u_t\big] \tag{179}$$

$$= E[X_t' q_t X_t \mid o_t] + E[\beta_{t+1}(O_{t+1} \mid o_t] + \mathrm{Tr}(c' \alpha_{t+1} c) +$$
$$+ \min_{u_t} E\big[(aX_t + bu_t)' \alpha_{t+1} (aX_t + bu_t) + u_t' g_t u_t \mid o_t, u_t\big] \tag{180}$$

where we used the fact that

$$E[E[X_{t+1} \alpha_{t+1} X_{t+1} \mid O_{t+1}] \mid o_t, u_t] = E[X_{t+1} \alpha_{t+1} X_{t+1} \mid o_t, u_t] \tag{181}$$

and $E[Z_{t,i} Z_{t,j} \,|\, x_t, u_t] = \delta_{i,j}$, and that, by assumption $E[\beta_{t+1}(O_{t+1}) \,|\, o_t, u_t]$ does not depend on u_t. Thus

$$\Phi_t(o_t) = E[X_t' q_t X_t \mid o_t] + E[\beta_{t+1}(O_{t+1}) \mid o_t] + \min_{u_t} E\big[(aX_t + bu_t)' \alpha_{t+1} (aX_t + bu_t)$$
$$+ u_t' g_t u_t \mid o_t, u_t\big] \tag{182}$$
$$\tag{183}$$

The minimization part is equivalent to the one presented in (228) with the following equivalence: $b \to b$, $x \to u_t$, $a \to \alpha_{t+1}$, $C \to aX_t$, $d \to g_t$. Thus, using (234)

$$\hat{u}_t = -\epsilon_t E[X_t \mid o_t] \tag{184}$$

where

$$\epsilon_t = \kappa_t a \tag{185}$$
$$\kappa_t = (b'\alpha_{t+1}b + g_t)^{-1}b'\alpha_{t+1} \tag{186}$$

And, using (244)

$$\begin{aligned}
&\min_{u_t} E\big[(aX_t + bu_t)'\alpha_{t+1}(aX_t + bu_t) + u_t'g_tu_t \mid o_t, u_t\big] \\
&= E[X_t'a'(\alpha_{t+1} - k_t'b'\alpha_{t+1})aX_t \mid o_t] \\
&\quad + E[(X_t - E[X_t \mid o_t])'a'\kappa_t b'\alpha_{t+1}a(X_t - E[X_t \mid o_t])]
\end{aligned} \tag{187}$$

We will later show that the last term is constant with respect to $u_{1:t}$. Thus,

$$\Phi_t(o_t) = E[X_t'\alpha_tX_t \mid o_t] + \beta_t(o_t) \tag{188}$$

where

$$\alpha_t = a'(\alpha_{t+1} - k_t'b'\alpha_{t+1})a + q_t \tag{189}$$
$$= a'\alpha_{t+1}(a - b\epsilon_t) + q_t \tag{190}$$

and

$$\beta_t(o_t) = E[\beta_{t+1}(O_{t+1}) \mid o_t] + \mathrm{Tr}(c'\alpha_{t+1}c) \tag{191}$$
$$+ E[(X_t - E[X_t \mid o_t])'a'\kappa_t b'\alpha_{t+1}a(X_t - E[X_t \mid o_t])] \tag{192}$$

By assumption $\beta_{t+1}(o_{t+1})$ is independent of $u_{1:t+1}$ we just need to show that

$$E[(X_t - E[X_t \mid o_t])'a'\kappa_t b'\alpha_{t+1}a(X_t - E[X_t \mid o_t])] \tag{193}$$

is also independent of $u_{1:t}$ for $\beta_t(o_t)$ to be independent of $u_{1:t}$, completing the induction proof.

Lemma 7.1. *The innovation term $X_t - E[X_t \mid o_t]$ is constant with respect to $u_{1:t}$.*

Bertsekas Volume I. Consider the following reference process

$$\tilde{X}_{t+1} = a\tilde{X}_t + cZ_t \tag{194}$$
$$\tilde{Y}_{t+1} = k\tilde{X}_{t+1} + mW_{t+1} \tag{195}$$
$$\tilde{O}_t = \tilde{Y}_{1:t} \tag{196}$$
$$\tilde{X}_1 \sim \nu_1 \tag{197}$$

which shares initial distribution ν_1 and noise variables Z, W with the processes X, Y, H defined in previous sections. Note

$$X_2 = aX_1 + bU_1 + cZ_1 \tag{198}$$

$$X_3 = a^2X_1 + abU_1 + acZ_1 + bU_2 + cZ_2 \tag{199}$$

$$\cdots \tag{200}$$

$$X_t = a^{t-1}X_1 + \sum_{\tau=1}^{t-2} a^{t-1-\tau}(bU_\tau + cZ_\tau) \tag{201}$$

$$\tag{202}$$

and by the same token

$$\tilde{X}_t = a^{t-1}X_1 + \sum_{\tau=1}^{t-2} a^{t-1-\tau}cZ_\tau \tag{203}$$

$$\tag{204}$$

Thus

$$E[X_t \mid o_t] = a^{t-1}E[X_1 \mid o_t] + \left(\sum_{\tau=1}^{t-2} a^{t-1-\tau}bu_\tau\right) + \sum_{\tau=1}^{t-2} a^{t-1-\tau}cE[Z_\tau \mid o_t] \tag{205}$$

$$E[\tilde{X}_t \mid o_t] = a^{t-1}E[X_1 \mid o_t] + \sum_{\tau=1}^{t-2} a^{t-1-\tau}cE[Z_\tau \mid o_t] \tag{206}$$

where we used the fact that $E[U_{1:t-1} \mid o_t] = u_{1:t-1}$. Thus

$$X_t - E[X_t \mid o_t] = \tilde{X}_t - E[\tilde{X}_t \mid o_t] \tag{207}$$

Note since

$$Y_t = ka^{t-1}X_1 + mW_k + k\sum_{\tau=1}^{t-2} a^{t-1-\tau}(bU_\tau + cZ_\tau) \tag{208}$$

$$\tilde{Y}_t = ka^{t-1}X_1 + mW_k + k\sum_{\tau=1}^{t-2} a^{t-1-\tau}cZ_\tau \tag{209}$$

then

$$\tilde{Y}_t = Y_t - k\sum_{\tau=1}^{t-2} a^{t-1-\tau}bU_\tau \tag{210}$$

and therefore knowing $o_{1:t}$ determines $\tilde{o}_{1:t} = y_{1:t}$. Thus

$$E[\tilde{X}_t \mid o_t] = E[\tilde{X}_t \mid y_{1:t}] \tag{211}$$

and

$$X_t - E[X_t \mid o_t] = \tilde{X}_t - E[\tilde{X}_t \mid y_{1:t}] \tag{212}$$

which is constant with respect to $u_{1:t-1}$. $\square$

Remark 7.1. Note the control equations for the partially observable case are identical to the control equations for the fully observable case, but using $E[X_t|o_t]$ instead of x_t.

7.1 Summary of Control Equations

Let

$$\alpha_T = q_T \tag{213}$$

$$\hat{u}_T = 0 \tag{214}$$

then move your way from $t = T - 1$ to $t = 1$ using the following recursion

$$\epsilon_t = (b'\alpha_{t+1}b + g_t)^{-1}b'\alpha_{t+1}a \tag{215}$$

$$\hat{u}_t = -\epsilon_t E[X_t \mid o_t] \tag{216}$$

$$\alpha_t = a'\alpha_{t+1}(a - b\epsilon_t) + q_t \tag{217}$$

where $E[X_t \mid o_t]$ is computed using the Kalman filter equations.

8 Appendix

Lemma 8.1. *If $w_i \geq 0$ and $\hat{\beta}$ maximizes $f(i, \beta)$ for all i then*

$$\max_\beta \sum_i w_i f(i, \beta) = \sum_i w_i \max_\beta f(i, \beta) \tag{218}$$

Proof.

$$\max_\beta \sum_i w_i f(i, \beta) \leq \sum_i \max_\beta f(i, \beta) = \sum_i w_i f(i, \hat{\beta}) \tag{219}$$

moreover

$$\max_\beta \sum_i w_i f(i, \beta) \geq \sum_i f(i, \hat{\beta}) = \sum_i w_i \max_\beta f(i\beta) \tag{220}$$

□

Lemma 8.2. *If $w_i \geq 0$ and*

$$\max_\beta \sum_i w_i f(i, \beta) = \sum_i w_i \max_\beta f(i, \beta) \tag{221}$$

then there is $\hat{\beta}$ such that for all i with $w_i > 0$

$$f(i, \hat{\beta}) = \max_\beta f(i, \beta) \tag{222}$$

Proof. Let
$$f(i,\hat{\beta}_i) = \max_\beta f(i,\beta) \tag{223}$$
and
$$f(i,\hat{\beta}) = \max_\beta \sum_i w_i f(i,\beta) \tag{224}$$
then
$$\sum_i w_i (f(i,\hat{\beta}_i) - f(i,\hat{\beta})) = 0 \tag{225}$$
Thus, since
$$f(i,\hat{\beta}_i) - f(i,\hat{\beta}) \geq 0 \tag{226}$$
it follows that
$$f(i,\hat{\beta}) = f(i,\hat{\beta}_i) = \max_\beta f(i,\beta) \tag{227}$$
for all i such that $w_i > 0$. □

Lemma 8.3 (Optimization of Quadratic Functions)**.** *This is one of the most useful optimization problem in applied mathematics. Its solution is behind a large variety of useful algorithms including Multivariate Linear Regression, the Kalman Filter, Linear Quadratic Controllers, etc. Let*
$$\rho(x) = E[(bx - C)'a(bx - C)] + x'd\,x \tag{228}$$
where a and d are symmetric positive definite matrices and C is a random vector with the same dimensionality as bx. Taking the Jacobian with respect to x and applying the chain rule we have
$$J_x\rho = E[J_{bx-C}(bx - C)'a(bx - C)\,J_x(bx - C)] + J_x x'd\,x \tag{229}$$
$$= 2E[(bx - C)'ab] + 2x'd \tag{230}$$
$$\nabla_x\rho = (J_x)' = 2b'a(bx - \mu) + 2d\,x \tag{231}$$
where $\mu = E[C]$. Setting the gradient to zero we get
$$(b'ab + d)x = b'a\mu \tag{232}$$
This is commonly known as the Normal Equation*. Thus the value $\hat{x}$ that minimizes ρ is*
$$\hat{x} = h\mu \tag{233}$$
where
$$h = (b'ab + d)^{-1}b'a \tag{234}$$
Moreover
$$\rho(\hat{x}) = (bh\mu - C)'a(bh\mu - C) + \mu'h'dh\mu \tag{235}$$
$$= \mu'h'b'abh\mu - 2\mu'h'b'a\mu + E[C'aC] + \mu'h'dh\mu \tag{236}$$

Now note

$$\begin{aligned}\mu' h' b' abh\mu + \mu' h' dh\mu &= \mu' h'(b'ab + d)h\mu && (237)\\ &= \mu' a' b(b'ab + d)^{-1}(b'ab + d)(b'ab + d)^{-1} b' a\mu && (238)\\ &= \mu' a' b(b'ab + d)^{-1} b' a\mu && (239)\\ &= \mu' h' b' a\mu && (240)\end{aligned}$$

Thus

$$\rho(\hat{x}) = E[C' aC] - \mu' h' b' a\mu \tag{241}$$

An important special case occurs if C is a constant, e.g., it takes the value c with probability one. In such case

$$\rho(\hat{x}) = c' ac - c' h' b' ac = c' kc \tag{242}$$

where

$$k = a - h' b' a = a - a' b(b'ab + d)^{-1} b' a \tag{243}$$

For the more general case it is sometimes useful to express (241) *as follows*

$$\rho(\hat{x}) = E[C' aC] - \mu' h' b' a\mu = E[C'(a - h' b' a)C] + E[(C - \mu)' h' b' a(C - \mu)] \tag{244}$$

Lemma 8.4 (Quadratic Regression)**.** *We want to minimize*

$$\rho(w) = \sum_i \Big(a_i + b_i^T w + w^T c_i w\Big)^2 \tag{245}$$

where a_i is a scalar, b_i, w are n-dimensional vectors and c_i an $n \times n$ symetric matrix[2]*. We solve the problem iteratively starting at a weight vector w_k linearizing the quadratic part of the function and iterating.*

Linearizing about w_k we get

$$\begin{aligned}w^T c_i w &\approx w_k^T c_i w_k + 2 w_k^T c_i (w - w_k)\\ &= -w_k^T c_i w_k + 2 w_k^T c_i w && (246)\end{aligned}$$

Thus

$$a_i + b_i^T w + w^T c_i w \approx a_i - w_k^T c_i w_k + (b_i + 2 c_i w_k)^T w \tag{247}$$

This results in a linear regression problem with predicted variables in a vector y with components of the form

$$y_i = -a_i + w_k^T c_i w_k \tag{248}$$

[2]We can always symetrize c_i with no loss of generality.

and predicting variables into a matrix x with rows

$$x_i = (b_i + 2c_i w_k)^T \tag{249}$$

with

$$w_{k+1} = (x'x)^{-1}x'y \tag{250}$$