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 $X \in \mathbb{R}^n$ is shorthand for $X : \Omega \rightarrow \mathbb{R}^2$, i.e., the random variable $X$ takes values in $\mathbb{R}^n$. We use $E$ for expected values and $\text{Var}$ 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 $E[Y | x]$ 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., $X_{1:t} \overset{\text{def}}{=} \{X_1, \ldots, X_t\}$. 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

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

where

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

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 \(\Phi_t\) be defined as follows

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

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 \(E[\bar{R}_{t+1} | x_{t+1}, c_{t+1:T}]\) for all \(x_{t+1}\), i.e.,

\[
E[\bar{R}_{t+1} | x_{t+1}, \hat{c}_{t+1:T}] = \max_{c_{t+1:T}} E[\bar{R}_{t+1} | x_{t+1}, c_{t+1:T}] \tag{5}
\]
and let $\hat{c}_t(x_t)$ maximize $E[\tilde{R}_t | x_t, c_t, \hat{c}_{t+1:T}]$ for all $x_t$ with $\hat{c}_{t:T}$ fixed, i.e.,

$$E[\tilde{R}_{t+1} | x_{t+1}, \hat{c}_t, \hat{c}_{t+1:T}] = \max_{c_t} E[\tilde{R}_{t+1} | x_{t+1}, c_t, \hat{c}_{t+1:T}]$$  \hspace{1cm} (6)

Then

$$E[\tilde{R}_t | x_t, \hat{c}_{t:T}] = \max_{c_{t:T}} E[\tilde{R}_t | x_t, c_{t:T}]$$  \hspace{1cm} (7)

for all $x_t$.

Proof.

$$\Phi_t(x_t) = \max_{c_t} E[\tilde{R}_t | x_t, c_t] = \max_{c_t} E[\tilde{R}_t + \alpha \tilde{R}_{t+1} | x_t, c_t]$$

$$= \max_{c_t} \left\{ E[\tilde{R}_t | x_t, c_t] + \alpha \max_{c_{t+1:T}} E[\tilde{R}_{t+1} | x_t, c_{t:T}] \right\}$$  \hspace{1cm} (8)

where we used the fact that

$$E[\tilde{R}_t | x_t, c_{t:T}] = E[\tilde{R}_t | x_t, c_t]$$  \hspace{1cm} (9)

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

$$\max_{c_{t+1:T}} E[\tilde{R}_{t+1} | x_t, c_{t:T}] = \max_{c_{t+1:T}} \sum_{x_{t+1}} p(x_{t+1} | x_t, c_t) E[\tilde{R}_{t+1} | x_{t+1}, c_{t+1:T}]$$  \hspace{1cm} (10)

where we used the fact that

$$p(x_{t+1} | x_t, c_{t:T}) = p(x_{t+1} | x_t, c_t)$$  \hspace{1cm} (11)

and

$$E[\tilde{R}_{t+1} | x_t, c_{t+1:T}] = E[\tilde{R}_{t+1} | x_{t+1}, c_{t+1:T}]$$  \hspace{1cm} (12)

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

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

$$= \sum_{x_{t+1}} p(x_{t+1} | x_t, c_t) \max_{c_{t+1:T}} E[\tilde{R}_{t+1} | x_{t+1}, c_{t+1:T}]$$  \hspace{1cm} (13)

$$= \sum_{x_{t+1}} p(x_{t+1} | x_t, c_t) E[\tilde{R}_{t+1} | x_{t+1}, \hat{c}_{t+1:T}] = E[\tilde{R}_{t+1} | x_t, c_t, \hat{c}_{t+1:T}]$$  \hspace{1cm} (14)

Thus we have that

$$\Phi_t(x_t) = \max_{c_{t:T}} E[\tilde{R}_t | x_t, c_{t:T}] = \max_{c_t} \left( E[\tilde{R}_t | x_t, c_t] + \alpha E[\tilde{R}_{t+1} | x_t, c_t, \hat{c}_{t+1:T}] \right)$$  \hspace{1cm} (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.,

$$E[R_T \mid x_t, c_T] = \max_{c_T} E[R_T \mid x_t, c_T]$$  \hspace{1cm} (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 the Bellman Optimality Equation which we state here as a corollary of the Optimality Principle.

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

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

for $t = 1 \cdots T$ where

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

and

$$\Phi_{T+1}(x) \overset{ad}{=} 0, \text{ for all } x$$  \hspace{1cm} (19)

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

$$\max_{c_{t+1}; T} E[R_{t+1} \mid x_t, c_{t:T}] = \sum_{x_{t+1}} p(x_{t+1} \mid x_t, c_t) \max_{c_{t+1}; T} E[R_{t+1} \mid x_{t+1}, c_{t+1}; T]$$  \hspace{1cm} (20)

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

Combining this with equation (8) completes the proof. \qed

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

- **Assumption 1:**
  
  $$E[R_t \mid x_t, c_t; T] = E[R_t \mid x_t, c_t]$$  \hspace{1cm} (22)

- **Assumption 2:**

  $$p(x_{t+1} \mid x_t, c_t, c_{t+1}; T) = p(x_{t+1} \mid x_t, c_t)$$  \hspace{1cm} (23)

- **Assumption 3:**

  $$E[\bar{R}_{t+1} \mid x_t, c_t, x_{t+1}, c_{t+1}; T] = E[\bar{R}_{t+1} \mid x_{t+1}, c_{t+1}; T]$$  \hspace{1cm} (24)
• Assumption 4: Most importantly we 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.

![Graphical Representation of the a time slice of a process satisfying the required assumptions. Arrows represent dependency relationships between variables.](image)

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

\[
p(x_{t+1} \mid x_{1:t}, c_{1:t}) = p(x_{t+1} \mid x_t, c_t)
\]  

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

\[
U_t = C_t(X_t) \in C_t(X_t)
\]

where \( C_t(x_t) \) 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

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

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

where

$$\Phi_t(x_t, c_{t:T}) = E[R_t \mid x_t, c_{t:T}]$$ (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

$$U_t = c_t(X_1)$$ (30)

which would violate Assumption 4. since

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

$$\sum_{t=0}^{T} f_t(C_t)$$ (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 $X_0 = x_0$.

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

\[
\Phi_t(x_t) = \max_{0 \leq u \leq 1} E[\Phi_{t+1}(X_{t+1}) | x_t, u_t]
\]

(34)

with boundary condition

\[
\Phi_T(x) = \log(x)
\]

(36)

Thus

\[
\Phi_{T-1}(x) = \max_{0 \leq u \leq 1} \{p \log(x + ux) + (1 - p) \log(x - ux)\}
\]

(37)

\[
= \log(x) + \max_{0 \leq u \leq 1} \{p \log(1 + u) + (1 - p) \log(1 - u)\}
\]

(38)

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

\[
\frac{2p - 1 - u}{1 - u^2} = 0
\]

(39)

Thus

\[
\hat{u}_{T-1}(x) = 2p - 1, \text{ provided } p > 0.5
\]

(40)

\[
\hat{U}_{T-1}(x) = \log(x) + p \log(2p) + (1 - p) \log(2(1 - p)) = \log(x) + K
\]

(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

\[
U_t = (2p - 1)X_t
\]

(42)

provided \( p \geq 0.5 \). 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

\[
X_{t+1} = aX_t + bu_t + cZ_t
\]

(43)

\[
X_1 = x_1
\]

(44)

where \( X_t \in \mathbb{R}^n \), is the system’s state, \( a \in \mathbb{R}^n \otimes \mathbb{R}^n \), \( u_t \in \mathbb{R}^m \), \( b \in \mathbb{R}^n \otimes \mathbb{R}^m \), \( Z_t \in \mathbb{R}^d \), \( c \in \mathbb{R}^n \otimes \mathbb{R}^d \) 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

\[
R_t = X_t'q_tX_t + U_t'g_tU_t
\]

(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 $\pi$ and terminal time $T \geq t$ as follows

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

(46)

### 3.1 Linear Policies: Policy Evaluation

We will consider first linear policies of the form $u_t = \theta_t x_t$, where $\theta_t$ 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.,

$$\Phi(x_t) = x_t^T \alpha_t + \beta_t$$

(47)

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

$$\Phi_T(x_T) = x_T^T q_T x_T = x_T^T \alpha_T x_T + \beta_T$$

(48)

where

$$\alpha_T = q_T, \quad \beta_T = 0$$

(49)

Assuming that

$$\Phi(x_{t+1}) = x_{t+1}^T \alpha_{t+1} x_t + \beta_{t+1}$$

(50)

and applying Bellman’s equation

$$\Phi_t(x_t) = x_t^T q_t x_t + x_t^T \theta_t^T \theta_t x_t$$

(51)

$$+ \gamma E[\Phi_{t+1}(X_{t+1}) \mid x_t, \theta_{t+1:T}] +$$

(52)

$$= x_t^T q_t x_t + x_t^T \theta_t^T \theta_t x_t$$

(53)

$$+ \gamma E[\Phi_{t+1}(ax_t + b\theta_t u_t + cZ_t,)] +$$

(54)

$$= x_t^T q_t x_t + x_t^T \theta_t^T \theta_t x_t$$

(55)

$$+ \gamma (ax_t + b\theta_t u_t)' \alpha_{t+1} (ax_t + b\theta_t u_t) + \gamma \text{Tr}(c' \alpha_{t+1} c) + \beta_{t+1}$$

(56)

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

$$E[Z_t c' \alpha_{t+1} c Z_t \mid x_t, u_t] = \sum_{ij} (c' \alpha_{t+1} c)_{ij} E[Z_{t,i} Z_{t,j}]$$

(57)

$$= \sum_{i} (c' \alpha_{t+1} c)_{ij} = \text{Tr}(c' \alpha_{t+1} c)$$

(58)

$^1$To avoid clutter we leave implicit the dependency of $\Phi$ on $t$ and $\theta$.
Thus
\[\Phi_t(x_t) = x_t' \left( q_t + \theta_t' g_t \theta_t + \gamma (a_t + b_t \theta_t)' \alpha_t+1 (a_t + b_t \theta_t) \right) x_t + \gamma \text{Tr}(c_t' \alpha_t+1 c_t) + \beta_{t+1} \]

(59)

Thus
\[\Phi(x_t) = x_t' \alpha_t x_t + \beta_t \]

(61)

where
\[\alpha_t = q_t + \theta_t' g_t \theta_t + \gamma (a_t + b_t \theta_t)' \alpha_{t+1} (a_t + b_t \theta_t) \]

(62)

\[= \theta_t' (q_t + \gamma b_t' \alpha_{t+1} \beta_t) + q_t + \gamma a_t' \alpha_t a_t \]

(63)

\[\beta_t = \text{Tr}(c_t' \alpha_{t+1} c_t) + \beta_{t+1} \]

(64)

3.2 Linear Policies: Policy Improvement

Taking the gradient with respect to \(\theta_t\) of the state value
\[\nabla \text{vec}[\theta_t] \Phi(x_t) = \nabla \text{vec}[\theta_t] x_t' \alpha_t x_t + \beta_t \]

(65)

\[= \nabla_{\theta_t} (\theta_t x_t)' g_t(\theta x_t) + \gamma \nabla_{\theta_t} x_t' \left( (a_t + b_t \theta_t)' \alpha_t+1 (a_t + b_t \theta_t) \right) x_t \]

(66)

Note
\[\nabla \text{vec}[\theta_t] (\theta_t x_t)' g_t(\theta x_t) = \nabla \text{vec}[\theta_t] \theta x_t \nabla_{\theta x} (\theta x)' g_t(\theta x) = x \otimes I \text{vec}[\theta x] \]

(67)

Thus
\[\nabla_{\theta} (\theta_t x_t)' g_t(\theta x) = g_t \theta x x' \]

(68)

Moreover
\[\nabla \text{vec}[\theta] x' (a + b \theta)' \alpha (a + b \theta) x = \nabla \text{vec}[\theta] (a + b \theta) x \]

(69)

\[= x \otimes b' \text{vec}[\alpha (a + b \theta) x] \]

(70)

Thus
\[\nabla_{\theta} x' (a + b \theta)' \alpha (a + b \theta) x = b' \alpha (a + b \theta) x x' \]

(72)

and
\[\nabla_{\theta} \Phi(x_t) = \left( g_t \theta_t + \gamma b_t' \alpha_{t+1} (a_t + b_t \theta_t) \right) x_t x_t' \]

(73)

We can thus improve the policy by performing gradient ascent
\[\theta_t \leftarrow \theta_t + \epsilon \left( g_t \theta_t + \gamma b_t' \alpha_{t+1} (a_t + b_t \theta_t) \right) x_t x_t' \]

(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 $\theta_t$ can also be found by setting the gradient to zero and solving the resulting algebraic equation. Note for

$$\hat{\theta}_t = \theta_t - \gamma \left( g_t + \gamma b_t' \alpha_{t+1} b_t \right) a_t$$ (75)

then

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

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

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

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

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

$$\Phi(x_T) = x_T' a_T x_T + \beta_T$$ (80)

and if

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

then, applying the Bellman Equations

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

$$+ \gamma (a x_t + b u_t)' \alpha_{t+1} (a x_t + b u_t) + \gamma \text{Tr}(c' \alpha_{t+1} c) + \beta_{t+1}$$ (83)

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

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

Setting the gradient to zero we get the optimal $u_t$

$$\hat{u}_t = \theta_t x_t$$ (85)

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

which is a linear policy.
3.4 Summary of Equations for Optimal Policy

Let

\[ \alpha_T = q_T \]  
\[ \hat{u}_T = 0 \]  

then move your way from \( t = T - 1 \) to \( t = 1 \) using the following recursion

\[ K_t = (b'\alpha_{t+1}b + g_t)^{-1}b'\alpha_{t+1}a \]  
\[ \alpha_t = q_t + a'\alpha_{t+1}(a - bK_t) \]

and the optimal action at time \( t \) is given by

\[ \hat{u}_t = -K_t x_t \]

If desired, the value function can be obtained as follows

\[ \Phi_t(x_t) = x_t'\alpha_t x_t + \gamma_t \]

where

\[ \gamma_t = \gamma_{t+1} + \text{Tr}(c'\alpha_{t+1}c) \]

Below is Matlab code

```matlab
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 \( \mu_t \) 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
\[ R_t = (X_t - \xi_t)'q_t(X_t - \xi_t) + U_t'g_tU_t \] (95)
where \(\xi_{1:T}\) is a desired sequence of states. We can handle this case by augmenting the system as follows
\[ \tilde{X}_{t+1} = \tilde{a}X_t + \tilde{b}u_t + \tilde{c}Z_t \] (96)
where
\[ \tilde{X}_t = \begin{pmatrix} X_t \\ \xi_t \\ 1 \end{pmatrix} \in \mathbb{R}^{2n} \] (97)
\[ \tilde{a} = \begin{pmatrix} a_{n \times n} & 0_{n \times n} & 0_{n \times 1} \\ 0_{n \times n} & 0_{n \times n} & \Delta \xi_t \\ 0_{1 \times n} & 0_{1 \times n} & 1 \end{pmatrix} \in \mathbb{R}^{2n+1} \otimes \mathbb{R}^{2n+1} \] (98)
\[ \tilde{b} = \begin{pmatrix} b_{n \times m} \\ 0_{n \times m} \\ 0_{1 \times m} \end{pmatrix} \in \mathbb{R}^{2n+1} \otimes \mathbb{R}^m \] (99)
\[ \tilde{c} = \begin{pmatrix} c_{n \times d} \\ 0_{n \times d} \\ 0_{1 \times d} \end{pmatrix} \in \mathbb{R}^{2n+1} \otimes \mathbb{R}^d \] (100)
where \(\Delta \xi_t \equiv \xi_{t+1} - \xi_t\) and we use subscripts as a reminder of the dimensionality of matrices. The return function is now strictly quadratic on the extended state space
\[ \tilde{R}_t = \tilde{X}_t'\tilde{q}_t\tilde{X}_t + U_t'g_tU_t \] (103)
where
\[ \tilde{q}_t = \begin{pmatrix} q_t & -q_t & 0_{n \times 1} \\ -q_t & q_t & 0_{n \times 1} \\ 0_{1 \times n} & 0_{1 \times n} & 0 \end{pmatrix} \in \mathbb{R}^{2n+1} \otimes \mathbb{R}^{2n+1} \] (104)

3.5 Example
Consider the simple case in which
\[ X_{t+1} = aX_t + u_t + cZ_t \] (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 \(b = I\).
\[ q_t = I, \quad q_t = 0, \quad g_t = 0, \quad \xi_t = 0. \] Thus we have

- \[ \mu_T = 0 \] (106)
- \[ \alpha_T = I \] (107)
- \[ \hat{u}_T = 0 \] (108)

\[ K_{T-1} = \alpha_T = I \] (110)
\[ \kappa_{T-1} = 0 \] (111)
\[ \alpha_{T-1} = I \] (112)
\[ \mu_{T-1} = 0 \] (113)

Thus, from which it follows that

- \[ \epsilon_t = I, \] (115)
- \[ \hat{u}_t = -ax_t, \text{ for } t = 1 \cdots T - 1 \] (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

\[ v_{t+\tau} = v_t + \int_0^\tau u_t ds = v_t + u_t \tau \] (117)
\[ x_{t+\Delta_t} = x_t + \int_0^{\Delta_t} v_{t+s} ds = x_t + v_t \Delta_t + u_t \frac{\Delta_t^2}{2} \] (118)

or in matrix form

\[
\begin{pmatrix}
  x_{t+\Delta_t} \\
  v_{t+\Delta_t}
\end{pmatrix}
= \begin{pmatrix}
  1 & \Delta_t \\
  0 & 1
\end{pmatrix}
\begin{pmatrix}
  x_t \\
  v_t
\end{pmatrix}
+ \begin{pmatrix}
  \frac{\Delta_t^2}{\Delta_t}
\end{pmatrix}
u_t
\] (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

\[
\begin{pmatrix}
  x_{t+\Delta_t} \\
  v_{t+\Delta_t}
\end{pmatrix}
= \begin{pmatrix}
  1 & \Delta_t - \epsilon \Delta_t^2/2 \\
  0 & 1 - \epsilon \Delta_t
\end{pmatrix}
\begin{pmatrix}
  x_t \\
  v_t
\end{pmatrix}
+ \begin{pmatrix}
  \frac{\Delta_t^2}{\Delta_t}
\end{pmatrix}
u_t
+ \begin{pmatrix}
  0 & \sigma \Delta_t^2/2 \\
  0 & \sigma \Delta_t
\end{pmatrix}
\begin{pmatrix}
  Z_{1,t} \\
  Z_{2,t}
\end{pmatrix}
\] (120)
We can express this as a 2-dimensional discrete time system

\[ \tilde{x}_{t+1} = a\tilde{x}_t + bu_t + cZ_t \]  

(121)

where

\[ \tilde{x}_t = \begin{pmatrix} x_t \\ v_t \end{pmatrix}, \quad a = \begin{pmatrix} 1 & \Delta t - \epsilon \Delta t^2/2 \\ 0 & 1 - \epsilon \Delta t \end{pmatrix}, \quad b = \begin{pmatrix} \frac{\Delta t^2}{2} \\ \Delta t \end{pmatrix}, \quad c = \begin{pmatrix} 0 & \sigma \Delta t^2/2 \\ 0 & \sigma \Delta t \end{pmatrix} \]  

(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.

\[ \text{Figure 2:} \]

4 Infinite Horizon Case

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

\[ \alpha = q + a'\alpha(a - b(b'ab + g)^{-1}b'aa) \]  

(123)
The stationary control function

\[ u_t = -Kx_t \]  
\[ K = (b'\alpha b + g)^{-1}b'\alpha a \]

minimizes the stationary cost

\[ \rho = \lim_{t \to \infty} \frac{1}{T} \sum_{t=1}^{T} E[X_t'qX_t + U_t'gU_t] \]

Regarding \( \beta \), given the definition of \( \rho \)

\[ \beta_t = \frac{t-1}{t} \beta_{t-1} + \frac{1}{t} \text{Tr}(c'\alpha c) \]

and in the stationary case

\[ \beta = \frac{t-1}{t} \beta + \frac{1}{t} \text{Tr}(c'\alpha c) \]
\[ \beta = \text{Tr}(c'\alpha c) \]

Thus the stationary value of state \( x_t \) is

\[ \Phi(x_t) = x_t'\alpha x_t + \text{Tr}(c'\alpha c) \]

4.1 Example

We want to control

\[ X_{t+1} = X_t + U_t + Z_t \]

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

We enter

\[ (\text{alpha}, L, K) = \text{dare}(a, b, q, g, 0, 1) \]

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 opposite sign. For \( q = 1, g = 10 \) we get \( K = 0.27 \), i.e., we need to reduce the gain of our response.

5 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.,

$$U_t = C_t(O_t)$$  \hspace{1cm} (133)

$$O_t = \left( \begin{array}{c} Y_{1:t} \\ U_{1:t-1} \end{array} \right)$$  \hspace{1cm} (134)

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

$$X_1 \sim \nu$$  \hspace{1cm} (135)

a sensor model

$$p(y_t | x_t, u_{t-1})$$  \hspace{1cm} (136)

and state dynamics model

$$p(x_{t+1} | x_t, u_t)$$  \hspace{1cm} (137)

**Remark 5.1.** 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:

$$\rho(c_{1:T}) = E[\bar{R}_1 | c_{1:T}]$$  \hspace{1cm} (138)

where

$$\bar{R}_t = \sum_{\tau=t}^{T} \alpha^{T-\tau} R_{\tau}, \ t = 1 \cdots T$$  \hspace{1cm} (139)

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

### 5.1 Equivalence with Fully Observable Case

- **Assumption 1:**

$$E[R_t | o_t, c_{1:T}] = E[R_t | o_t, c_t]$$  \hspace{1cm} (140)
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.

- Assumption 2:

\[ p(o_{t+1} \mid o_t, c_t, c_{t+1:T}) = p(o_{t+1} \mid o_t, c_t) \]  

(141)

- Assumption 3:

\[ E[\hat{R}_{t+1} \mid o_t, c_t, o_{t+1}, c_{t+1:T}] = E[\hat{R}_{t+1} \mid o_{t+1}, c_{t+1:T}] \]  

(142)

Remark 5.2. 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 5.3. 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).

5.2 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: 
  \[ S_1 = f_1(Y_1) \]  
  (143)

• Assumption 2: 
  \[ S_{t+1} = f_t(S_t, Y_{t+1}, U_t) \]  
  (144)

• Assumption 3: 
  \[ E[R_t \mid o_t, u_t] = E[R_t \mid s_t, u_t] \]  
  (145)

• Assumption 4: 
  \[ p(y_{t+1} \mid y_{1:t}, u_{1:t}) = p(y_{t+1} \mid s_t, u_t) \]  
  (146)

where \( f_t \) are known functions. Note 
\[
\Phi_T(o_T) = E[R_T \mid o_t] = E[R_T \mid s_t] = \tilde{\Phi}_T(s_T) \]  
and thus the optimal action at time \( T \) depends only on \( s_T \). We will now show that if this is true at time \( t+1 \) then it is also true at time \( t \)

\[
\Phi_t(o_t) = \min_{u_t} E[R_t + \alpha \Phi_{t+1}(O_{t+1}) \mid o_t, u_t] \]  
(148)

\[
= \min_{u_t} \left\{ E[R_t \mid s_t, u_t] + \alpha \sum_{y_{t+1}} p(y_{t+1} \mid o_t, u_t) \Phi_{t+1}(o_t, u_t, y_{t+1}) \right\} \]  
(149)

\[
= \min_{u_t} E[R_t + \alpha \tilde{\Phi}_{t+1}(S_{t+1}) \mid s_t, u_t] \overset{\text{def}}{=} \tilde{\Phi}_t(s_t) \]  
(150)

where \( s_t \overset{\text{def}}{=} f_t(o_t) \). Thus, we only need to keep track of \( s_t \) to find the optimal policy with respect to \( o_t \).

### 5.3 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

\[ S_1(x_1) = p(x_1 \mid Y_1) = f_1(Y_1) \text{ for all } x_1 \]  
(151)

\[ \text{(152)} \]

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

\[ p(x_{t+1} \mid y_{1:t+1}, u_{1:t}) \propto \sum_{x_t} p(x_t \mid y_{1:t}, u_{1:t}) p(x_{t+1} \mid x_t, u_t) p(y_{t+1} \mid x_t) \]  
(153)

which satisfies the second assumption.
\[ E[R_t \mid o_t, u_t] = \sum_{x_t} p(x_t \mid o_t, u_t) R_t(x_t, u_t) = E[R_t \mid s_t, u_t] \quad (154) \]

and

\[ p(y_{t+1} \mid y_{1:t}, u_{1:t}) = \sum_{x_{t+1}} p(x_t \mid y_{1:t}, u_{1:t-1}) p(x_{t+1} \mid x_t, u_t) p(y_{t+1} \mid x_{t+1}) \quad (155) \]

### 5.4 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.

### 6 Linear Quadratic Gaussian (LQG)

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

\[
X_{t+1} = aX_t + bu_t + cZ_t \\
Y_{t+1} = kX_{t+1} + mW_{t+1} \\
X_1 \sim \nu_1
\]  

(156) (157) (158)

where \( X_t \in \mathbb{R}^n \) is the system’s state, \( a \in \mathbb{R}^{n \times n}, u_t \in \mathbb{R}^m, b \in \mathbb{R}^{n \times m}, Z_t \in \mathbb{R}^d, c \in \mathbb{R}^{n \times d} \) 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 control sequence \( u_{t:T} = u_t \cdots u_T \) that minimizes the following cost

\[ R_t = X_t' q_t X_t + U_t' g_t U_t \]  

(159)

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

\[ O_t \overset{\text{def}}{=} \begin{pmatrix} Y_{1:t} \\ U_{1:t-1} \end{pmatrix} \]  

(160)

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

\[ \Phi_t(o_t) = E[X_t' \alpha_t X_t \mid o_t] + \beta_t(o_t) \]  

(161)

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
$$\Phi_T(o_T) = E[X_T'q_TX_T | o_T] = E[X_T'\alpha_T X_T | o_T] + \beta_T(o_T)$$
and our assumption is correct for the terminal time $T$ with
$$\alpha_T = q_T, \quad \beta_T(o_T) = 0$$
Assuming (161) is correct at time $t + 1$ and applying Bellman’s equation
$$\Phi_t(o_t) = E[X_t'q_t X_t | o_t] + \min_{u_t} E[\Phi_{t+1}(O_{t+1}) + u_t'g_t u_t | o_t, u_t]$$
$$= E[X_t'q_t X_t | o_t] + E[\beta_{t+1}(O_{t+1}) | o_t, u_t]$$
$$+ \min_{u_t} E[(aX_t + bu_t + cZ_t)'\alpha_{t+1}(aX_t + bu_t + cZ_t) + u_t'g_t u_t | o_t, u_t]$$
$$= E[X_t'q_t X_t | o_t] + E[\beta_{t+1}(O_{t+1}) | o_t] + \text{Tr}(c'\alpha_{t+1}c) +$$
$$+ \min_{u_t} E[(aX_t + bu_t)'\alpha_{t+1}(aX_t + bu_t) + u_t'g_t u_t | o_t, u_t]$$
where we used the fact that
$$E[E[X_{t+1}\alpha_{t+1} X_{t+1} | O_{t+1}] | o_t, u_t] = E[X_{t+1}\alpha_{t+1} X_{t+1} | o_t, u_t]$$
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
$$\Phi_t(o_t) = E[X_t'q_t X_t | o_t] + E[\beta_{t+1}(O_{t+1}) | o_t] + \min_{u_t} E[(aX_t + bu_t)'\alpha_{t+1}(aX_t + bu_t)$$
$$+ u_t'g_t u_t | o_t, u_t]$$
(168)
(169)
The minimization part is equivalent to the one presented in (323) with the following equivalence: $b \rightarrow b$, $x \rightarrow u_t$, $a \rightarrow \alpha_{t+1}$, $C \rightarrow aX_t$, $d \rightarrow g_t$. Thus, using (329)
$$\hat{u}_t = -\epsilon_t E[X_t | o_t]$$
(170)
where
$$\epsilon_t = \kappa_t a$$
(171)
$$\kappa_t = (b'\alpha_{t+1} b + g_t)^{-1}b'\alpha_{t+1}$$
(172)
And, using (339)
$$\min_{u_t} E[(aX_t + bu_t)'\alpha_{t+1}(aX_t + bu_t) + u_t'g_t u_t | o_t, u_t]$$
$$= E[X_t'a'(\alpha_{t+1} - k'\alpha_{t+1})aX_t | o_t]$$
$$+ E[(X_t - E[X_t | o_t])'a'\kappa_t b'\alpha_{t+1}a(X_t - E[X_t | o_t])]$$
(173)
We will later show that the last term is constant with respect to $u_{1:t}$. Thus,
\[ \Phi_t(o_t) = E[X_t' \alpha_t X_t \mid o_t] + \beta_t(o_t) \] (174)
where
\[ \alpha_t = a'(\alpha_{t+1} - k' b' \alpha_{t+1})a + q_t \] (175)
\[ = a' \alpha_{t+1} (a - b \epsilon_t) + q_t \] (176)
and
\[ \beta_t(o_t) = E[\beta_{t+1}(O_{t+1}) \mid o_t] + \text{Tr}(c' \alpha_{t+1} c) + E[(X_t - E[X_t \mid o_t])' a' \kappa_t b' \alpha_{t+1} a(X_t - E[X_t \mid o_t])] \] (177)
(178)
By assumption $\beta_{t+1}(o_{t+1})$ is independent of $u_{1:t+1}$ we just need to show that
\[ E[(X_t - E[X_t \mid o_t])' a' \kappa_t b' \alpha_{t+1} a(X_t - E[X_t \mid o_t])] \] (179)
is also independent of $u_{1:t}$ for $\beta_t(o_t)$ to be independent of $u_{1:t}$, completing the induction proof.

**Lemma 6.1.** 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
\[ \tilde{X}_{t+1} = a \tilde{X}_t + c Z_t \] (180)
\[ \tilde{Y}_{t+1} = k \tilde{X}_{t+1} + m W_{t+1} \] (181)
\[ \tilde{O}_t = \tilde{Y}_{1:t} \] (182)
\[ \tilde{X}_1 \sim \nu_1 \] (183)
which shares initial distribution $\nu_1$ and noise variables $Z, W$ with the processes $X, Y, H$ defined in previous sections. Note
\[ X_2 = a X_1 + b U_1 + c Z_1 \] (184)
\[ X_3 = a^2 X_1 + a b U_1 + a c Z_1 + b U_2 + c Z_2 \] (185)
\[ \ldots \] (186)
\[ X_t = a^{t-1} X_1 + \sum_{\tau=1}^{t-2} a^{t-1-\tau} (b U_\tau + c Z_\tau) \] (187)
and by the same token
\[ \tilde{X}_t = a^{t-1} X_1 + \sum_{\tau=1}^{t-2} a^{t-1-\tau} c Z_\tau \] (189)
(190)
Thus

\[ E[X_t | o_t] = a^{t-1} E[X_1 | o_t] + \left( \sum_{\tau=1}^{t-2} a^{t-1-\tau} b u_\tau \right) + \sum_{\tau=1}^{t-2} a^{t-1-\tau} c E[Z_\tau | o_t] \] (191)

\[ E[\tilde{X}_t | o_t] = a^{t-1} E[X_1 | o_t] + \sum_{\tau=1}^{t-2} a^{t-1-\tau} c E[Z_\tau | o_t] \] (192)

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

\[ X_t - E[X_t | o_t] = \tilde{X}_t - E[\tilde{X}_t | o_t] \] (193)

Note since

\[ Y_t = ka^{t-1} X_1 + m W_k + k \sum_{\tau=1}^{t-2} a^{t-1-\tau}(b U_\tau + c Z_\tau) \] (194)

\[ \tilde{Y}_t = ka^{t-1} X_1 + m W_k + k \sum_{\tau=1}^{t-2} a^{t-1-\tau} c Z_\tau \] (195)

then

\[ \tilde{Y}_t = Y_t - k \sum_{\tau=1}^{t-2} a^{t-1-\tau} b U_\tau \] (196)

and therefore knowing \( o_{1:t} \) determines \( \tilde{a}_{1:t} = y_{1:t} \). Thus

\[ E[\tilde{X}_t | o_t] = E[\tilde{X}_t | y_{1:t}] \] (197)

and

\[ X_t - E[X_t | o_t] = \tilde{X}_t - E[\tilde{X}_t | y_{1:t}] \] (198)

which is constant with respect to \( u_{1:t-1} \).

**Remark 6.1.** 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 \( x_t \).

### 6.1 Summary of Control Equations

Let

\[ \alpha_T = q_T \] (199)

\[ \hat{u}_T = 0 \] (200)

then move your way from \( t = T - 1 \) to \( t = 1 \) using the following recursion

\[ \epsilon_t = (b' \alpha_{t+1} + g_t)^{-1} b' \alpha_{t+1} a \] (201)

\[ \hat{u}_t = -\epsilon_t E[X_t | o_t] \] (202)

\[ \alpha_t = a' \alpha_{t+1} (a - b \epsilon_t) + q_t \] (203)

where \( E[X_t | o_t] \) is computed using the Kalman filter equations.
Continuous Time Control

For this section we recommend to read first the tutorial on stochastic differential equations\(^?\), particularly the section on Ito’s rule.

Consider a dynamical system governed by the following system of stochastic differential equations

\[
dX_t = a(X_t, U_t) + c(X_t, U_t)dB_t \quad (204)
\]

where \(dB_t\) is a Brownian motion differential.

### 7.1 Value Function for Finite Horizon Problems

Consider a fixed policy \(\pi\) and terminal time \(T\). The value of visiting state \(x\) at time \(t\) is defined as follows

\[
v(x, t) = E\left[\int_t^T e^{-\frac{1}{\tau}(s-t)} r(X_s, U_s, t) ds + e^{-\frac{1}{\tau}(T-t)} g(X_T) \mid X_t = x, \pi\right] \quad (205)
\]

where \(r\) is the instantaneous reward function, \(\tau\) the time constant for the temporal discount of the reward, and \(g\) is the terminal reward. Note

\[
v(x, t) = E\left[\int_t^{t+\epsilon} e^{-\frac{1}{\tau}(s-t)} r(X_s, U_s, s) ds \mid x_t, \pi\right]
\]

\[
= E\left[\int_t^{t+\epsilon} e^{-\frac{1}{\tau}(s-t)} r(X_s, U_s, s) ds + r(x_{t+\epsilon}) \mid x_t, \pi\right] e^{-\frac{1}{\tau}\epsilon} \quad (206)
\]

Thus

\[
\frac{1}{\epsilon} \left( E[v(X_{t+\epsilon}, t + \epsilon) \mid x_t, \pi] e^{-\epsilon/\tau} - v(x_t, t) \right) = -\frac{1}{\epsilon} E\left[\int_t^{t+\epsilon} e^{-\frac{1}{\tau}(s-t)} r(X_s, U_s, s) ds \mid x_t, \pi\right] \quad (207)
\]

Taking limits on the right hand side of (207)

\[
\lim_{\epsilon \to 0} \frac{1}{\epsilon} E\left[\int_t^{t+\epsilon} e^{-\frac{1}{\tau}(s-t)} r(X_s, U_s, s) ds \mid x_t, \pi\right] = r(x_t, u_t, t) \quad (208)
\]

Regarding the left hand side of (207), let

\[
f(\epsilon) \overset{\text{def}}{=} E[v(X_{t+\epsilon}, t + \epsilon) \mid x_t, \pi] \quad (209)
\]

Thus

\[
\frac{1}{\epsilon} \left( E[v(X_{t+\epsilon}, t + \epsilon) \mid x_t, \pi] e^{-\epsilon/\tau} - v(x_t) \right) = \frac{f(\epsilon) e^{-\epsilon/\tau} - f(0)}{\epsilon} \quad (210)
\]
It is easy to verify that for any differentiable function \( f \)
\[
\lim_{\epsilon \to 0} \frac{f(\epsilon) e^{-\frac{1}{\tau} \epsilon} - f(0)}{\epsilon} = \dot{f}(0) - \frac{1}{\tau} f(0) \tag{211}
\]
where \( \dot{f} \) is the first derivative of \( f \). Thus
\[
\lim_{\epsilon \to 0} \frac{1}{\epsilon} \left( E[v(X_{t+\epsilon}) \mid X_t, \pi] e^{-\frac{1}{\tau} \epsilon} - v(x_t) \right) = \frac{dv(x_t, t)}{dt} - \frac{1}{\tau} v(x_t, t) \tag{212}
\]
where the total derivative of \( v \) is defined as follows
\[
\frac{dv(x_t, t)}{dt} = \lim_{\epsilon \to 0} \frac{1}{\epsilon} \left( E[v(X_{t+\epsilon}) \mid X_t, \pi] - v(x_t) \right) \tag{213}
\]
Thus taking limits on the left hand side and right hand side of (207) we get
\[
\frac{dv(x_t, t)}{dt} - \frac{1}{\tau} v(x_t, t) = -r(x_t, u_t, t) \tag{214}
\]
We will now expand the total derivative \( dv(x_t, t)/dt \). For \( s \geq t \) let
\[
Y_s \overset{\text{def}}{=} v(X_s, s) \tag{215}
\]
Note \( Y_s \) is defined only for \( s \geq t \) and is already conditioned on \( \{X_t = x_t\} \). Using Ito’s rule we get
\[
dY_s = v_t(X_s, s) ds + v_x(X_s, s) \cdot dX_s \\
+ \frac{1}{2} \text{Tr} \left( c^2(X_s, U_s) v_{xx}(X_s, s) \right) ds \tag{216}
\]
\[
U_s \overset{\text{def}}{=} \pi(X_s, s) \tag{217}
\]
\[
v_t(x, s) \overset{\text{def}}{=} \frac{\partial v(x, s)}{\partial s} \tag{218}
\]
\[
v_x(x, s) \overset{\text{def}}{=} \frac{\partial v(x, s)}{\partial x} \tag{219}
\]
\[
v_{xx}(x, s) \overset{\text{def}}{=} \frac{\partial^2 v(x, s)}{\partial x \partial x'} \tag{220}
\]
\[
c^2(x, s) \overset{\text{def}}{=} c^T(x, u) c(x, u) \tag{221}
\]
Setting \( s = t \) and taking expected values
\[
\frac{dE[Y_t]}{dt} = \frac{dv(x_t, t)}{dt} = v_t(x_t, t) + v_x(x_t, t) \cdot a(x_t, u_t) \\
+ \frac{1}{2} \text{Tr} \left( c^2(x_t, u_t) v_{xx}(x_t, t) \right) \tag{222}
\]
Putting together (214) and (222) we get the Hamilton Jacoby Belman equation (HJB) for the value function of a policy $\pi$

\[
\frac{1}{\tau} v(x, t) = r(x, u, t) + \frac{\partial v(x, t)}{\partial t} + a(x, u) + \frac{1}{2} \text{Tr} \left( c^T(x, u)c(x, u) \frac{\partial v(x, t)}{\partial x^2} \right)
\]

\[
u(x, t) = \pi(x, t)
\]

\[
v(x, T) = g(x)e^{-\frac{1}{\tau}(T-t)}
\]

(223)

**Numerical Solution:** We have the value of $v$ for time $T$. If we can get the first and second derivatives of $v$ with respect to $x$ we can then use the HJB equation to obtain $\frac{\partial v(x, T)}{\partial t}$. We can then use this to find $v$ for time step $T - \Delta t$

\[
v(x, T - \Delta t) \approx v(x, T) - \Delta t \frac{\partial v(x, T)}{\partial t}
\]

(224)

We can then progress backwards in time until we reach the starting time $t$.

**Discrete Time Approximation:** From (206) we note to first order

\[
v(x_t, t) \approx r(x_t, \pi(x_t))\Delta t + e^{-\Delta t/\tau} E[v(X_{t+\Delta t}, t + \Delta t) | x_t, \pi]
\]

(225)

### 7.2 Optimal Value Function for Finite Horizon Problems

The optimal value function is defined as follows

\[
\hat{v}(x, t) = \sup_{\pi} v^\pi(x, t)
\]

(226)

where $v^\pi$ is the value function with respect to policy $\pi$. Thus

\[
\hat{v}(x, t) = \sup_{\pi} \left\{ r(x, \pi(x), t) + v^T_t(x, t) + v^T_t(x, t) \cdot a(x, \pi(x)) + \frac{1}{2} \text{Tr} \left( c^2(x, \pi(x))v^T_{xx}(x, t) \right) \right\}
\]

(227)

and since at the extremum $\pi$ takes the value of the optimal policy

\[
\hat{v}(x, t) = \sup_{\pi} \left\{ r(x, \pi(x)) + \hat{v}_t(x, t) + \hat{v}_x(x, t) \cdot a(x, \pi(x)) + \frac{1}{2} \text{Tr} \left( c^2(x, \pi(x))\hat{v}_{xx}(x, t) \right) \right\}
\]

(228)
And since the only part of the equation that depends on \( \pi \) is \( u = \pi(x) \) the HJB equation for the optimal value function follows

\[
\frac{1}{\tau} \hat{v}(x, t) = \sup_u \left\{ r(x, u) + \hat{v}_t(x, t) + \hat{v}_x(x, t) \cdot a(x, u) + \frac{1}{2} \text{Tr} \left( c^2(x, u) \hat{v}_{xx}(x, t) \right) \right\}
\]

\[
v(x, T) = g(x)
\]

(229)

### 7.3 Value Function for Infinite Horizon Problems

We can think of the infinite horizon case as the limiting case of a finite horizon problem.

\[
v(x) = \lim_{T \to \infty} E \left[ \int_t^T e^{-\frac{1}{2}(s-t)} r(X_s, U_s) \, ds \mid X_t = x, \pi \right]
\]

(230)

Note we made the reward to be independent of the time \( t \), in which case the value function will also be independent of \( t \). Thus the derivative of \( v \) with respect to time needs to be zero and the HJB for the value function follows

\[
\frac{1}{\tau} v(x) = r(x, u) + v_x(x) \cdot a(x, u) + \frac{1}{2} \text{Tr} \left( c^2(x, u) v_{xx}(x) \right)
\]

\[
u(x, t) = \pi(x)
\]

(231)

Using the same logic, we get the HJB for the optimal value function

\[
\frac{1}{\tau} \hat{v}(x) = \sup_u \left\{ r(x, u) + \hat{v}_x(x) \cdot a(x, u) + \frac{1}{2} \text{Tr} \left( c^2(x, u) \hat{v}_{xx}(x) \right) \right\}
\]

(232)

### 7.4 An important special case

Consider a process defined by the following stochastic differential equation:

\[
dX_t = a(X_t)dt + b(X_t)U_tdt + c(X_t)dB_t
\]

(233)

For an arbitrary \( t \) we let

\[
v(x, t) \overset{\text{def}}{=} \max_{\pi} E \left[ \int_t^T e^{-\frac{1}{2}(s-t)} r(X_s, U_s) \, ds + g_T(X_T) \mid X_t = x, \pi \right]
\]

(234)

where \( U_s = \pi(X_s) \) and the instantaneous reward takes the following form

\[
r(x, u) \overset{\text{def}}{=} g(x) - u^T q u
\]

(235)
In this case the HJB equation looks as follows

\[
\frac{1}{\tau} v(x,t) = \max_u \left\{ g(x) - u^T q u + \frac{\partial v(x,t)}{\partial t} + a(x)^T \frac{\partial v(x,t)}{\partial x} + u^T b(x)^T \frac{\partial v(x,t)}{\partial x} + \frac{1}{2} \text{Tr}[c^T(x)c(x) \frac{\partial^2 v(x,t)}{\partial x^2}] \right\}
\]  

(236)

Most importantly the maximum over \( u \) can be computed analytically. Taking the gradient of the right hand side of (236) with respect to \( u \) and setting it to zero we get

\[-2qu + b(x)^T \frac{\partial v(x,t)}{\partial x} = 0\]  

(237)

Thus the optimal action is

\[\hat{u} = \frac{1}{2}q^{-1}b(x)^T \frac{\partial v(x,t)}{\partial x}\]  

(238)

If \( q \) is not full rank then there is an infinite number of optimal actions. We can choose one by using the pseudo-inverse of \( q \). We need to be careful about \( q \). For example, consider the 1-D case. If we let \( q = 0 \) the optimal gain would go to infinity, which basically sets the state to zero in an infinitesimal time \( dt \).

Substituting the optimal action into the HJB equation we get

\[
\frac{1}{\tau} v(x,t) = g(x) - \hat{u}^T q \hat{u} + \frac{\partial v(x,t)}{\partial t} + a(x)^T \frac{\partial v(x,t)}{\partial x} + 2 \hat{u}^T q \hat{u} + \frac{1}{2} \text{Tr}[c^T(x)c(x) \frac{\partial^2 v(x,t)}{\partial x^2}] \]  

(239)

Simplifying, the HJB equation for the optimal value function looks as follows

\[-\frac{\partial v(x,t)}{\partial t} = -\frac{1}{\tau} v(x,t) + g(x) + \hat{u}^T q \hat{u} + \frac{\partial v(x,t)}{\partial x} a(x) + \frac{1}{2} \text{Tr}[c^T(x)c(x) \frac{\partial^2 v(x,t)}{\partial x^2}] \]  

(240)

\[\hat{u}(x) = \frac{1}{2}q^{-1}b(x)^T \frac{\partial v(x,t)}{\partial x}\]

### 7.5 Linear Quadratic Tracker and Regulator

Let

\[dX_t = aX_t + bU_t + cdB_t\]  

(241)

with

\[v(x,t) = E[\int_t^T r(X_s, U_s)e^{-\frac{1}{2}(s-t)} ds | X_t = x, \pi]\]  

(242)
where
\[ U_s = \pi(X_s) \] (243)
\[ r(x,u) = -(x - \xi)^T p(x - \xi) - u^T q u \] (244)

where the target state \( \xi \) can be a function of time. This corresponds to the problem of having the state \( X_t \) track the trajectory \( \xi_t \). We assume the value function takes the following form
\[ v(x,t) = -(x^T \alpha_t x - 2\beta_t^T x + \gamma_t) \] (245)

Thus,
\[ \frac{\partial v(x,t)}{\partial x} = 2(\beta_t - \varkappa_t x) \] (246)
\[ \frac{\partial^2 v(x,t)}{\partial x^2} = -2\varkappa_t \] (247)
\[ \frac{\partial v(x,t)}{\partial t} = -x' \alpha_t x + 2 \beta_t^T x - \gamma_t \] (248)

where
\[ \varkappa_t = \alpha_t + \alpha_t^T \] (249)
\[ \dot{\varkappa}_t = \frac{d\varkappa_t}{dt} \] (250)
\[ \dot{\beta}_t = \frac{d\beta_t}{dt} \] (251)
\[ \dot{\gamma}_t = \frac{d\gamma_t}{dt} \] (252)

Consider the optimal HJB equation (240)
\[ -\frac{\partial v(x,t)}{\partial t} = -\frac{1}{\tau} v(x,t) + g(x) + \dot{u}^T q \dot{u} + \frac{\partial v(x,t)^T}{\partial x} a x \]
\[ + \frac{1}{2} \text{Tr}[c(x)^T c(x) \frac{\partial^2 v(x,t)}{\partial x^2}] \] (253)

where
\[ g(x) = -(x - \xi)^T p(x - \xi) \] (254)
\[ \dot{u}(x) = \frac{1}{2} q^{-1} b^T \frac{\partial v(x,t)}{\partial x} = q^{-1} b^T (\beta_t - \varkappa_t x) \] (255)

Thus
\[ x^T \varkappa_t x - 2\beta_t^T x + \gamma_t = \frac{1}{\tau} x^T \alpha_t x - \frac{2}{\tau} \beta_t^T x + \frac{1}{\tau} \gamma - (x - \xi_t)^T p(x - \xi_t) \]
\[ + (\beta_t - \varkappa_t x)^T b q^{-1} b^T (\beta_t - \varkappa_t x) \]
\[ + 2(\beta_t - \varkappa_t x)^T a x - \text{Tr}[c^T c \varkappa_t] \] (256)
Expanding some terms

\[
\begin{align*}
x^T \dot{\alpha}_t x - 2 \dot{\beta}_t^T x + \dot{\gamma}_t &= \frac{1}{\tau} x^T \alpha_t x - \frac{2}{\tau} \dot{\beta}_t^T x + \frac{1}{\tau} \\
&- x^T px + 2 \xi^T px - \xi^T p \xi_t \\
&+ x^T \tilde{\alpha}_t b q^{-1} b^T \tilde{\alpha}_t x - 2 \dot{\beta}_t^T b^{-1} b^T \tilde{\alpha}_t x + \beta_t^T b^{-1} b^T \beta_t \\
&+ 2 \beta_t^T a x - 2 x^T \tilde{\alpha}_t a x - \text{Tr}[c^T c \tilde{\alpha}_t]
\end{align*}
\] (259)

Gathering quadratic, linear, and constant terms we get the continuous time Ricatti equations

\[
\begin{align*}
\hat{u}_t(x) &= \omega_t - \kappa_t x_t \\
\omega_t &= q^{-1} b^T \beta_t \\
\kappa_t &= q^{-1} b^T \tilde{\alpha}_t \\
v(x, t) &= -x^T \alpha_t x + 2 \beta_t^T x - \gamma_t \\
\tilde{\alpha}_t &= \frac{1}{\tau} \alpha_t - p + \tilde{\alpha}_t b q^{-1} b^T \tilde{\alpha}_t - 2 \tilde{\alpha}_t a \\
\tilde{\beta}_t &= -\frac{1}{\tau} \beta_t - p^T \xi_t + \tilde{\alpha}_t b q^{-1} b^T \beta_t - a^T \beta_t \\
\tilde{\gamma}_t &= \frac{1}{\tau} \gamma_t - \xi^T p \xi_t + \beta^T b q^{-1} b^T \beta_t - \text{Tr}[c^T c \alpha_t] \\
\tilde{\alpha}_t &= (\alpha_t + \alpha_t^T)/2 \\
\alpha_T &= (p + p^T)/2 \\
\beta_T &= (p + p^T)/2 \xi_T \\
\gamma_T &= \xi^T p \xi_T
\end{align*}
\] (263)

For initialization we used the fact that \( x' ay = x'(a + a')y/2 \).

We can solve this equation numerically using Euler’s method. We start at time \( T \). This gives us the temporal derivatives for \( \alpha, \beta, \gamma \). Their values at time \( t - \Delta_t \) can be obtained from those derivatives. We can then iterate until we reach the current time \( t \).

**Linear Quadratic Regulator**  A special case of the linear quadratic tracker is the linear quadratic regulator. In this case \( \xi_t = 0 \) for all \( t \).

Thus

\[
\alpha_T = \frac{p + p^T}{2} \\
\beta_T = 0 \\
\gamma_T = 0
\] (264) (265) (266)

The update equation for \( \beta \) show that in this case \( \dot{\beta}_T = 0 \) and therefore \( \beta_t = 0 \).
Thus the update equations for the linear quadratic regulator are as follows

\[
\hat{u}_t(x) = -k_t x \\
k_t = q^{-1} b^T \hat{\alpha}_t \\
\dot{\hat{\alpha}}_t = \frac{1}{\tau} \alpha_t - p + \hat{\alpha}_t b q^{-1} b^T \hat{\alpha}_t - 2 \hat{\alpha}_t a \\
\gamma_t = \frac{1}{\tau} \gamma_t - \text{Tr}[c^T \Gamma c] 
\]

### 7.6 Nonlinear Systems

Here we present a recent approach to non-linear continuous time for the special case in 7.4. The approach is based on (78) but here we adapt it to the finite horizon problem. We will assume \( v \) can be expressed as a linear combination of known features of the state \( x \), i.e.,

\[
v(x, t) = \phi(x)^T w(t) = \sum_{i=1}^{n_f} \phi_i(x) w_i(t) 
\]

where \( \phi : \mathbb{R}^{n_x} \rightarrow \mathbb{R}^{n_f} \) is a known function that maps each state \( x \) into \( n_f \) features of that state. \( w \in \mathbb{R}^{n_f} \) is an unknown weight vector that tells us how to combine the state features to obtain the value function of a state. Thus

\[
\frac{\partial v(x, t)}{\partial x} = \sum_{i=1}^{n_f} \phi_i(x) w_i(t) = \dot{\phi}(x) w(t) 
\]

where

\[
\dot{\phi}(x) \overset{def}{=} \nabla_x \phi(x) 
\]

and \( \dot{\phi} \) is an \( n_x \times n_f \) matrix whose columns are the \( \dot{\phi}_i \) terms

\[
\dot{\phi} = [\dot{\phi}_1, \cdots, \dot{\phi}_{n_f}] 
\]

Moreover

\[
\frac{\partial^2 v(x, t)}{\partial x^2} = \sum_{i=1}^{n_f} w_i(t) \ddot{\phi}_i(x) 
\]

where \( \phi_i \) is an \( n_x \times n_x \) Hessian matrix

\[
\ddot{\phi}_i(x) = \nabla^2_x \phi_i(x) 
\]

Thus the HJB equation takes the following form

\[
\frac{1}{\tau} \phi(x)^T w(t) = g(x) + a(x)^T \dot{\phi}(x) w(t) + \frac{1}{4} w^T(t) \dot{\phi}(x)^T b(x) q^{-1} b^T(x) \dot{\phi}(x) w(t) + \frac{1}{4} \text{Tr}[c(x)^T c(x) \sum_{i=1}^{n_f} \ddot{\phi}_i(x) w_i(t)] 
\]
Discretizing in time

\[
\frac{\partial v(x,t)}{\partial t} = \frac{1}{\Delta t} v(x, t + \Delta t) - \frac{1}{\Delta t} v(x, t) \tag{279}
\]

\[
= \frac{1}{\Delta t} v(x, t + \Delta t) - \frac{1}{\Delta t} \phi(x)^T w(t) \tag{280}
\]

Collecting terms constant, linear and quadratic with respect to \( w \) we get

\[
g(x) + \frac{1}{\Delta t} v(x, t + \Delta t)
+ \left( \phi(x)^T a(x) + h(x) - \left( \frac{1}{\tau} + \frac{1}{\Delta t} \right) \phi(x) \right)^T w(t)
+ \frac{1}{2} w^T (t) \phi(x)^T b(x) q^{-1} b(x) \phi(x) w(t) = 0 \tag{281}
\]

where \( h(x) \) is an \( n_f \) dimensional vector whose \( i^{th} \) element is defined as follows

\[
h_i(x) = \frac{1}{2} \text{Tr}[c^T(x) c(x) \dddot{\phi}_i(x)] \tag{282}
\]

Our goal is to find a value of \( w \) that satisfies (281). To do so we collect a sample \( \{x^1, x^2, \cdots x^{n_s}\} \) of states and define an error function \( \rho \) which captures the extent to which the HJB equation is violated. For time step \( T \) we want to find \( w(T) \) such that

\[
\phi(x)^T w(T) \approx g_T(x) \tag{283}
\]

An estimate of \( w(T) \) can be found by solving the following linear regression problem

\[
\rho(w(T)) = \sum_{i=1}^{n_s} \left[ g(x) - \phi(x^i)^T w(T) \right]^2 \tag{284}
\]

For \( t < T \) we let

\[
\rho(w(t)) = \sum_{i=1}^{n_s} \left[ a_i(t) + b_i^T(t) w(t) + w(t)^T c_i(t) w(t) \right]^2 \tag{285}
\]

where

\[
a_i(t) = g(x^i) + \frac{1}{\Delta t} v(x^i, t + \Delta t) \tag{286}
\]

\[
b_i(t) = \phi(x^i)^T a(x^i) + h(x^i) - \left( \frac{1}{\tau} + \frac{1}{\Delta t} \right) \phi(x^i) \tag{287}
\]

\[
c_i(t) = \frac{1}{4} \phi(x^i)^T b(x^i) q^{-1} b(x^i)^T \phi(x^i) \tag{288}
\]

This is a Quadratic Regression problem that can be solved using iterative methods (see Appendix).. Unfortunately this problem has local minima (or difficult
plateaus. Thus it is important to get good starting points. The solution for time $T$ is unique and we can use it as the starting point for time $t - \Delta t$. Provided $\Delta t$ is small, this should be a good starting solution. For some reason, starting points close to zero seem to also work well. Note to compute the $a_i(t)$ terms we need $v(x, t + \Delta t)$. We can thus solve the problem by doing a backward pass, starting at time $T$.

Another important issue is to have enough samples so that the regression problem to estimate $w(t)$ is not underconstrained. If the number of samples is small one possibility is to use something like Bayesian regression which allows for sequential learning of the parameters.

What’s Needed

- $a(x), b(x)$ can be learned from examples using non-linear regression with error
  \[ e(x) = \Delta x - a(x)\Delta t + g(x)u\Delta t \] (289)
- $c(x)$ can be obtained from model’s error
  \[ c(x)c(x)^T = \text{Cov}(\Delta X/\Delta t - a(X) - g(X)U) \] (290)
- $q$ the matrix for the quadratic error of the action.
- A way to sample from $g(x)$, the cost of the state.

7.6.1 Using Gaussian Radial Basis Functions

Gaussian functions centered at a fixed set of states $\mu^1 \ldots \mu^n$, and with fixed precision matrices $\nu_i$ can be used as feature functions, i.e.,
\[ \phi_i(x) = \exp \left( -\frac{1}{2} (x^i - \mu^i)^T \nu_i (x^i - \mu^i) \right) \] (291)
where $\mu^i$ is a fixed $n_x$ dimensional vector and $\nu_i$ is an $n_x \times n_x$ symmetric positive definite matrix. Thus in this case
\[ \dot{\phi}_i(x) = \phi_i(x) \nu_i (\mu_i - x) \] (292)
\[ \ddot{\phi}(x) = \phi_i(x) \left( \nu_i (x - \mu_i) (x - \mu_i)^T \nu_i - \nu_i \right) \] (293)

7.7 Example

Let
\[ dX_t = X_t dt + U_t dt + dB_t \] (294)
\[ r(x, u) = g(x) - \frac{1}{2} \|u\|^2 \] (295)
\[ g(x) = -\lambda x^2 \] (296)
We will approximate the value function using linear and quadratic features

\[ \phi(x) = \begin{pmatrix} 1 \\ x \\ x^2 \end{pmatrix} \]  

(297)

Thus

\[ a = b = c = 1 \]  

(298)

\[ \dot{\phi}(x) = (0, 1, 2x) \]  

(299)

\[ \ddot{\phi}_1 = 0 \]  

(300)

\[ \ddot{\phi}_2 = 0 \]  

(301)

\[ \ddot{\phi}_3 = 2 \]  

(302)

\[ h_1(x) = 0 \]  

(303)

\[ h_2(x) = 0 \]  

(304)

\[ h_3(x) = 2 \]  

(305)

For a fixed weight vector \( w^k \) we get

\[ q^k(x) = \dot{\phi}(x)^T b(x) b(x)^T \dot{\phi}(x) w^k = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 2x \\ 0 & 2x & 4x^2 \end{pmatrix} \begin{pmatrix} w^k_1 \\ w^k_2 \\ w^k_3 \end{pmatrix} \]  

(306)

\[ = \begin{pmatrix} 0 \\ w^k_2 + 2xw^k_3 \\ 2xw^k_2 + 4x^2w^k_3 \end{pmatrix} \]  

(307)

Thus

\[ y_i = \frac{1}{2} q^k(x^i)^T w^k - g(x^i) = \frac{1}{2} q^k(x^i)^T w^k + \lambda(x^i)^2 \]  

(308)

\[ = \frac{1}{2}(w^k_2)^2 + 2x^i w^k_2 w^k_3 + (x^i)^2 (\lambda + 2(w^k_3)^2) \]  

(309)

and the \( i^{th} \) row of \( \theta \) is

\[ \theta^T_{1i} = q^k(x^i) + \dot{\phi}(x^i)^T a(x^i) + h(x^i) - \frac{1}{\tau} \phi(x^i) \]  

(310)

\[ = \begin{pmatrix} \frac{1}{2}(w^k_2)^2 + 2x^i w^k_3 \\ 2x^i w^k_2 + 4(x^i)^2 w^k_3 \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \end{pmatrix} + \begin{pmatrix} 0 \\ 2 \end{pmatrix} - \frac{1}{\tau} \begin{pmatrix} 1 \\ (x^i)^2 \end{pmatrix} \]  

(311)

\[ = \begin{pmatrix} -\frac{1}{\tau}(w^k_2)^2 + x^i(2w^k_3 - \frac{1}{\tau}) + 1 \\ 2x^i(w^k_2 + 1) + (4w^k_3 - \frac{1}{\tau})(x^i)^2 + 2 \end{pmatrix} \]  

(312)
8 Appendix

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

\[
\max_{\beta} \sum_i w_i f(i, \beta) = \sum_i w_i \max_{\beta} f(i, \beta)
\]  \hspace{1cm} (313)

Proof.

\[
\max_{\beta} \sum_i w_i f(i, \beta) \leq \sum_i \max_{\beta} f(i, \beta) = \sum_i w_i f(i, \hat{\beta})
\]  \hspace{1cm} (314)

moreover

\[
\max_{\beta} \sum_i w_i f(i, \beta) \geq \sum_i f(i, \hat{\beta}) = \sum_i w_i \max_{\beta} f(i\beta)
\]  \hspace{1cm} (315)

\[\square\]

Lemma 8.2. If \( w_i \geq 0 \) and

\[
\max_{\beta} \sum_i w_i f(i, \beta) = \sum_i w_i \max_{\beta} f(i, \beta)
\]  \hspace{1cm} (316)

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

\[
f(i, \hat{\beta}) = \max_{\beta} f(i, \beta)
\]  \hspace{1cm} (317)

Proof. Let

\[
f(i, \hat{\beta}_i) = \max_{\beta} f(i, \beta)
\]  \hspace{1cm} (318)

and

\[
f(i, \hat{\beta}) = \max_{\beta} \sum_i w_i f(i, \beta)
\]  \hspace{1cm} (319)

then

\[
\sum_i w_i (f(i, \hat{\beta}_i) - f(i, \hat{\beta})) = 0
\]  \hspace{1cm} (320)

Thus, since

\[
f(i, \hat{\beta}_i) - f(i, \hat{\beta}) \geq 0
\]  \hspace{1cm} (321)

it follows that

\[
f(i, \hat{\beta}) = f(i, \hat{\beta}_i) = \max_{\beta} f(i, \beta)
\]  \hspace{1cm} (322)

for all \( i \) such that \( w_i > 0 \). \[\square\]

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

\[
\rho(x) = E[(bx - C)'a(bx - C)] + x'\alpha x
\]  \hspace{1cm} (323)
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

$$J_x \rho = E[J_{bx-C}(bx-C)'a(bx-C) J_x(bx-C)] + J_x x'dx \quad (324)$$

$$\nabla_{x'} \rho = (J_x)' = 2b'a(bx - \mu) + 2d x \quad (325)$$

where $\mu = E[C]$. Setting the gradient to zero we get

$$b'ab + d)x = b'a \mu \quad (326)$$

This is commonly known as the Normal Equation. Thus the value $\hat{x}$ that minimizes $\rho$ is

$$\hat{x} = h\mu \quad (327)$$

where

$$h = (b'ab + d)^{-1}b'a \quad (328)$$

Moreover

$$\rho(\hat{x}) = (bh\mu - C)'a(bh\mu - C) + \mu'h'dh\mu \quad (329)$$

$$= \mu'h'b'abh\mu - 2\mu'h'b'a\mu + E[C'aC] + \mu'h'dh\mu \quad (330)$$

Now note

$$\mu'h'b'abh\mu + \mu'h'dh\mu = \mu'h'(b'ab + d)h\mu \quad (331)$$

$$= \mu'a'b(b'ab + d)^{-1}(b'ab + d)(b'ab + d)^{-1}b'a\mu \quad (332)$$

$$= \mu'a'b(b'ab + d)^{-1}b'a\mu \quad (333)$$

$$= \mu'h'b'a\mu \quad (334)$$

Thus

$$\rho(\hat{x}) = E[C'aC] - \mu'h'b'a\mu \quad (335)$$

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

$$\rho(\hat{x}) = c'ac - c'h'b'ac = c'kc \quad (336)$$

where

$$k = a - h'b'a = a - a'b(b'ab + d)^{-1}b'a \quad (337)$$

For the more general case it is sometimes useful to express (336) as follows

$$\rho(\hat{x}) = E[C'aC] - \mu'h'b'a\mu = E[C'(a - h'b'a)C] + E[(C - \mu)'h'b'a(C - \mu)] \quad (338)$$
Lemma 8.4 (Quadratic Regression). We want to minimize

\[ \rho(w) = \sum_i \left( a_i + b_i^T w + w^T c_i w \right)^2 \]  

where \( a_i \) is a scalar, \( b_i, w \) are \( n \)-dimensional vectors and \( c_i \) an \( n \times n \) symmetric 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

\[ w^T c_i w \approx w_k^T c_i w_k + 2w_k^T c_i (w - w_k) = -w_k^T c_i w_k + 2w_k^T c_i w \]  

Thus

\[ a_i + b_i^T w + w^T c_i w \approx a_i - w_k^T c_i w_k + (b_i + 2c_i w_k)^T w \]  

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

\[ y_i = -a_i + w_k^T c_i w_k \]  

and predicting variables into a matrix \( x \) with rows

\[ x_i = (b_i + 2c_i w_k)^T \]  

with

\[ w_{k+1} = (x'x)^{-1} x'y \]  

\(^2\)We can always symetrize \( c_i \) with no loss of generality.