Continuous Time Stochastic Optimal Control

June 7, 2011

Please cite as

Movellan J. R. (2011) *Continuous Time Stochastic Optimal Control* MPLab Tutorials, University of California San Diego

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

$$dX_t = a(X_t, U_t)dt + c(X_t, U_t)dB_t \tag{1}$$

where dB_t is a Brownian motion differential. One way to think of this equation is the limit as $\Delta_t \to 0$ of the following process

$$\Delta X_t = a(X_t, U_t)\Delta_t + c(X_t, U_t)\sqrt{\Delta_t} Z_t \tag{2}$$

where Z_t is a vector of independent standard Gaussian random variables.

0.1 The HJB Equation for Finite Horizon Value Functions

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

$$v(x,t) = E[\int_t^T e^{-\frac{1}{\tau}(s-t)} r(X_s, U_s, s)ds + e^{-\frac{1}{\tau}(T-t)} g(X_T) \mid X_t = x] \tag{3}$$

where r is the instantaneous reward, τ the time constant for the temporal discount of the reward, and g is the terminal reward.

For $s \geq t$ let

$$Y_s \stackrel{\text{def}}{=} v(X_s, s) \tag{4}$$

Using Ito's rule we get

$$\begin{aligned} dY_s = dv(X_s, s) &= v_t(X_s, s)ds + v_x(X_s, s) \cdot dX_s \\ &+ \frac{1}{2}\mathrm{Tr}\Big(c(X_s, U_s)c'(X_s, U_s)v_{xx}(X_s, s)\Big)ds \end{aligned} \tag{5}$$

where

$$U_s \stackrel{\text{def}}{=} \pi(X_s, s) \tag{6}$$

$$v_t(x, s) \stackrel{\text{def}}{=} \frac{\partial v(x, s)}{\partial s} \tag{7}$$

$$v_x(x, s) \stackrel{\text{def}}{=} \frac{\partial v(x, s)}{\partial x} \tag{8}$$

$$v_{xx}(x, s) \stackrel{\text{def}}{=} \frac{\partial^2 v(x, s)}{\partial x \partial x'} \tag{9}$$

Taking expected values given $X_t = x$, and noting that expected values of stochastic integrals are zero

$$\begin{aligned} \frac{dE[Y_s \mid X_t = x]}{ds} &= E\Big[v_t(X_s, s) + v_x(X_s, s)' a(X_s, U_s) \\ &+ \frac{1}{2}\mathrm{Tr}\Big(c(X_s, U_s)c'(X_s, U_s)v_{xx}(X_s, s)\Big) \mid X_t = x\Big] \end{aligned} \tag{10}$$

Evaluating it at $s = t$ we get

$$\left.\frac{dE[Y_s \mid X_t = x]}{ds}\right|_{s=t} = v_t(x,t) + v_x(x,t)' a(x,t) + \frac{1}{2}\mathrm{Tr}\Big(c(x,u)c'(x,u)v_{xx}(x,t)\Big) \tag{11}$$

Next we show that the left hand side of the equation above takes the following form

$$\left.\frac{dE[Y_s \mid X_t = x]}{ds}\right|_{s=t} = \frac{1}{\tau}v(x,t) - r(x,\pi(x),t) \tag{12}$$

First let's get a better understanding of what this time derivative means

$$\begin{aligned}\frac{dE[Y_s \mid X_t = x]}{ds} &= \lim_{\epsilon\to 0}\frac{1}{\epsilon}E[Y_{s+\epsilon} - Y_s \mid X_t = x] \\ &= \lim_{\epsilon\to 0}\frac{1}{\epsilon}E[v(X_{s+\epsilon}, s+\epsilon) - v(X_s, s) \mid X_t = x]\end{aligned} \tag{13}$$

and

$$\left.\frac{dE[Y_s \mid X_t = x]}{ds}\right|_{s=t} = \lim_{\epsilon\to 0}\frac{1}{\epsilon}E[v(X_{t+\epsilon}, t+\epsilon) - v(x,t) \mid X_t = x] \tag{14}$$

Note

$$\begin{aligned}v(x,t) = E[&\int_t^{t+\epsilon} e^{-\frac{1}{\tau}(s-t)} r(X_s, U_s, s) ds \mid X_t = x] \\ &+ E[\int_{t+\epsilon}^{T} e^{-\frac{1}{\tau}(s-(t+\epsilon))} r(X_s, U_s, s) ds \mid X_t = x]\ e^{-\frac{1}{\tau}\epsilon} \\ = E[&\int_t^{t+\epsilon} e^{-\frac{1}{\tau}(s-t)} r(X_s, U_s, s) ds \mid X_t = x] \\ &+ E[v(X_{t+\epsilon}, t+\epsilon) \mid X_t = x]\ e^{-\frac{1}{\tau}\epsilon}\end{aligned} \tag{15}$$

and

$$\begin{aligned}&\frac{1}{\epsilon}\Big(E[v(X_{t+\epsilon}, t+\epsilon) \mid X_t = x]\ e^{-\epsilon/\tau} - v(x,t)\Big) \\ &\quad = -\frac{1}{\epsilon}E[\int_t^{t+\epsilon} e^{-\frac{1}{\tau}(s-t)} r(X_s, U_s, s) ds \mid X_t = x]\end{aligned} \tag{16}$$

Taking limits on the right hand side of (0.1)

$$\lim_{\epsilon\to 0} -\frac{1}{\epsilon}E[\int_t^{t+\epsilon} e^{-\frac{1}{\tau}(s-t)} r(X_s, U_s, s) ds \mid X_t = x] = -r(x,\pi(x),t) \tag{17}$$

Thus

$$\lim_{\epsilon\to 0}\frac{1}{\epsilon}\Big(E[v(X_{t+\epsilon}, t+\epsilon) \mid X_t = x]\ e^{-\epsilon/\tau} - v(x,t)\Big) = -r(x,\pi(x),t) \tag{18}$$

Regarding the left hand side of (0.1), let

$$f(\epsilon) \stackrel{\text{def}}{=} E[v(X_{t+\epsilon}, t+\epsilon) \mid X_t = x] \tag{19}$$

Thus

$$\frac{1}{\epsilon}\Big(E[v(X_{t+\epsilon}, t+\epsilon) \mid X_t = x]\; e^{-\frac{1}{\tau}\epsilon} - v(x)\Big) = \frac{f(\epsilon)e^{-\frac{1}{\tau}\epsilon} - f(0)}{\epsilon} \tag{20}$$

Using the product rule for derivatives it follows that

$$\lim_{\epsilon \to 0} = \frac{f(\epsilon)g(\epsilon) - f(0)g(0)}{\epsilon} = \dot{f}(0)g(0) + f(0)\dot{g}(0) \tag{21}$$

where $\dot{f}\dot{g}$ are the first derivative of f, g. Thus with

$$g(x) = e^{-\frac{1}{\tau}x} \tag{22}$$

$$\dot{g}(x) = -\frac{1}{\tau}g(x) \tag{23}$$

it follows that

$$\lim_{\epsilon \to 0} = \frac{f(\epsilon)e^{-\frac{1}{\tau}\epsilon} - f(0)}{\epsilon} = \dot{f}(0) - \frac{1}{\tau}f(0) \tag{24}$$

where

$$\begin{aligned}\dot{f}(0) &= \lim_{\epsilon \to 0} \frac{f(\epsilon) - f(0)}{\epsilon} = \lim_{\epsilon \to 0} \Big(E[v(X_{t+\epsilon}, t+\epsilon) \mid X_t = x] - v(x,t)\Big) \\ &= \frac{dE[Y_s \mid X_t = x]}{ds}\bigg|_{s=t}\end{aligned} \tag{25}$$

Thus

$$\begin{aligned}&\lim_{\epsilon \to 0} \frac{1}{\epsilon}\Big(E[v(X_{t+\epsilon}, t+\epsilon) \mid X_t = x]\; e^{-\frac{1}{\tau}\epsilon} - v(x,t)\Big) \\ &= \frac{dE[Y_s \mid X_t = x]}{ds}\bigg|_{s=t} - \frac{1}{\tau}v(x,t)\end{aligned} \tag{26}$$

and using (18)

$$\frac{dE[v(X_t) \mid X_t = x]}{dt} - \frac{1}{\tau}v(x,t) = -r(x, \pi(x), t) \tag{27}$$

From which (12) follows. Putting together (12) and (11) we get the Hamilton Jacoby Belman equation (HJB) for the value function of a fixed policy π

$$\boxed{\begin{array}{l}\frac{1}{\tau}\, v(x,t) = r(x,u,t) + \frac{\partial v(x,t)}{\partial t} + \frac{\partial v(x,t)}{\partial x}' a(x,u) + \frac{1}{2}\mathrm{Tr}\Big(c(x,u)c(x,u)' \frac{\partial^2 v(x,t)}{\partial x^2}\Big) \\ u = \pi(x,t) \\ v(x,T) = g(x)e^{-\frac{1}{\tau}(T-t)}\end{array}} \tag{28}$$

Stochastic Discrete Time Approximation Basically we approximate the continuous time HJB with a discrete time Bellman equation. From (15) we note

$$\boxed{v(x,t) \approx r(x,\pi(u_t))\Delta t + e^{-\Delta t/\tau} E[v(X_{t+\Delta t}, t+\Delta t) \mid X_t = x]} \tag{29}$$

We know $v(\cdot, T)$ so we can approximate $v(x, T-\Delta t)$ by running the SDE forward from time $T-\Delta t$ to time T with initial condition x. We can use this for a set of states and use function interpolation to get an estimate for all the other states. This gives us $v(\cdot, T-\Delta t)$ we can then keep moving backwards until we reach the initial time$t$. One problem with this approach is that it does not use any knowledge about the spatial derivatives of the value function.

Deterministic Discrete Time Approximation 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 $\partial v(x,T)\ \partial t$. This determines $v(x, T-\Delta t)$.

$$v(x, T-\Delta_t) \approx v(x,T) - \Delta_t \frac{\partial v(x,T)}{\partial t} \tag{30}$$

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

The temporal derivative at time T equals the temporal derivative at time $T-\Delta t$. We approximate approximate v at time $T-\Delta t$ as a weighted sum of features of x. The spatial derivaites are then also a weighted sum of features of x. This results on a regression problem. For a set of sates of interest the predictors are

$$-\frac{1}{\tau} v(x,t) + v_x(x,t)' a(x,u) + \frac{1}{2} \mathrm{Tr}\Big(c(x,u)c(x,u)' \nabla^2_{xx} v(x,t)\Big) \tag{31}$$

which are a linear function of features of the state. The predicted values are

$$-r(x,u,t) - \frac{\partial v(x,t)}{\partial t} \tag{32}$$

we do non-linear regression to find w. We can then use this to find v for time step $T-\Delta_t$ for a set of points. We can then move our way backwards until we reach the startint time t.

0.2 The Bellman Equation for the Q function

I have the impression that changing an action at a specific point in time will not change the value. We may need to divide by dt or something like that. So this section is ifi Let v^π represent the value function under policy π. Note

$$v^\pi(x,t) = \tau\Big\{r(x,u,t) + v^\pi_t(x,t) + v^\pi_x(x,t)\cdot a(x,u) + \frac{1}{2}\mathrm{Tr}\Big(c(x,u)c(x,u)'v^\pi_{xx}(x,t)\Big)\Big\} \tag{33}$$

where $u = \pi(x,t)$. Consider what would happen if we defined a new policy π identical to π except for the fact that at time t it maps x into another action u', i.e., $\pi'(x,t) = u'$. The value function under the new policy would be as follows

$$v^{\pi'}(x,t) = \tau\Big\{r(x,u',t) + v^{\pi'}_t(x,t) + v^{\pi'}_x(x,t)\cdot a(x,u') + \frac{1}{2}\mathrm{Tr}\Big(c(x,u')c(x,u')'v^{\pi'}_{xx}(x,t)\Big)\Big\} \tag{34}$$

We can thus think of the value of responding to x at time t with action u and then following the fixed policy π as follows

$$Q^\pi(x,u,t) = \tau\Big\{r(x,u,t) + v^\pi_t(x,t) + v^\pi_x(x,t)\cdot a(x,u) + \frac{1}{2}\mathrm{Tr}\Big(c(x,u)c(x,u)'v^\pi_{xx}(x,t)\Big)\Big\} \tag{35}$$

Policy improvement: This leads to the following approach to improve a policy π. For state t and time t define a new policy π' that chooses an action u' such that

$$Q^{\pi'}(x,u',t) > Q^\pi(x,u,t) \tag{36}$$

One way to do so would be to have

$$u' = u + \epsilon\frac{\partial v^\pi(x,t)}{\partial u} \tag{37}$$

0.3 Optimal Value Function for Finite Horizon Problems

The optimal value function is defined as follows

$$\hat{v}(x,t) = \sup_\pi v^\pi(x,t) \tag{38}$$

where v^π is the value function with respect to policy π. Thus

$$\hat{v}(x,t) = \sup_\pi \tau\left\{r(x,\pi(x),t) + v^\pi_t(x,t) + v^\pi_x(x,t)\cdot a(x,\pi(x)) + \frac{1}{2}\mathrm{Tr}\Big(c(x,\pi(x))c(x,\pi(x))'v^\pi_{xx}(x,t)\Big)\right\} \tag{39}$$

and since at the extremum π takes the value of the optimal policy

$$\hat{v}(x,t) = \sup_{\pi} \tau \left\{ r(x,\pi(x)) + \hat{v}_t(x,t) + \hat{v}_x(x,t) \cdot a(x,\pi(x)) + \frac{1}{2}\mathrm{Tr}\Big(c(x,\pi(x))c(x,\pi(x))'\hat{v}_{xx}(x,t)\Big)\right\} \tag{40}$$

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

$$\boxed{\begin{array}{l} \frac{1}{\tau}\ \hat{v}(x,t) = \sup_u \left\{ r(x,u) + \hat{v}_t(x,t) + \hat{v}_x(x,t) \cdot a(x,u) + \frac{1}{2}\mathrm{Tr}\Big(c(x,u)c(x,u)'\hat{v}_{xx}(x,t)\Big)\right\} \\ \hat{v}(x,T) = g(x) \end{array}} \tag{41}$$

0.4 Value Function for Infinite Horizon Problems

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

$$v(x) = \lim_{T\to\infty} E[\int_t^T e^{-\frac{1}{\tau}(s-t)} r(X_s,U_s)\ s \mid X_t = x, \pi] \tag{42}$$

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

$$\boxed{\begin{array}{l} \frac{1}{\tau} v(x) = r(x,u) + v_x(x) \cdot a(x,u) + \frac{1}{2}\mathrm{Tr}\Big(c^2(x,u)v_{xx}(x)\Big) \\ u = \pi(x) \end{array}} \tag{43}$$

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

$$\boxed{\frac{1}{\tau}\ \hat{v}(x) = \sup_u \left\{ r(x,u) + \hat{v}_x(x) \cdot a(x,u) + \frac{1}{2}\mathrm{Tr}\Big(c^2(x,u)\hat{v}_{xx}(x)\Big)\right\}} \tag{44}$$

0.5 An important special case

Consider a process defined by the following stochastic differential: equation

$$dX_t = a(X_t)dt + b(X_t)U_t dt + c(X_t)dB_t \tag{45}$$

For an arbitrary t we let

$$v(x,t) \stackrel{\text{def}}{=} \max_\pi E[\int_t^T e^{-\frac{1}{\tau}(s-t)}\ r(X_s, U_s)\ ds + g_T(X_T) \mid X_t = x, \pi] \tag{46}$$

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

$$r(x,u) \stackrel{\text{def}}{=} g(x) - \frac{1}{2}u'qu \tag{47}$$

In this case the HJB equation looks as follows

$$\begin{aligned} \frac{1}{\tau}v(x,t) = \max_u \Big\{ g(x) - \frac{1}{2}u'qu + \frac{\partial v(x,t)}{\partial t} + a(x)'\frac{\partial v(x,t)}{\partial x} + u'b(x)'\frac{\partial v(x,t)}{\partial x} \\ + \frac{1}{2}\mathrm{Tr}[c(x)c(x)'\frac{\partial^2 v(x,t)}{\partial x^2}]\Big\} \end{aligned} \tag{48}$$

Most importantly the maximum over u can be computed analytically. Taking the gradient of the right hand side of (56) with respect to u and setting it to zero we get

$$-qu + b(x)'\frac{\partial v(x,t)}{\partial x} = 0 \tag{49}$$

Thus the optimal action is

$$\hat{u} = q^{-1}b(x)'\frac{\partial v(x,t)}{\partial x} \tag{50}$$

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

$$\begin{aligned} \frac{1}{\tau}v(x,t) = g(x) - \frac{1}{2}\hat{u}'q\hat{u} + \frac{\partial v(x,t)}{\partial t} \\ + a(x)'\frac{\partial v(x,t)}{\partial x} + \hat{u}'q\hat{u} + \frac{1}{2}\mathrm{Tr}[c(x)c(x)'\frac{\partial^2 v(x,t)}{\partial x^2}] \end{aligned} \tag{51}$$

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

$$\boxed{\begin{aligned} &-\tfrac{\partial v(x,t)}{\partial t} = -\tfrac{1}{\tau}v(x,t) + g(x) + \tfrac{1}{2}\hat{u}'q\hat{u} + \tfrac{\partial v(x,t)}{\partial x}'a(x) \\ &\qquad + \tfrac{1}{2}\mathrm{Tr}[c(x)c(x)'\tfrac{\partial^2 v(x,t)}{\partial x^2}] \\ &\hat{u}(x) = q^{-1}b(x)'\tfrac{\partial v(x,t)}{\partial x} \end{aligned}} \tag{52}$$

0.6 Action Dependent Noise

We can generalize the previous case to include action dependent noise

$$dX_t = a(X_t)dt + b(X_t)U_t dt + \Big(c(X_t) + \sum_k U_{k,t} h_k(X_t)\Big) dB_t \tag{53}$$

where $U_{k,t}$ is the k^{th} component of U_t. For an arbitrary t we let

$$v(x,t) \stackrel{\text{def}}{=} \max_\pi E[\int_t^T e^{-\frac{1}{\tau}(s-t)}\ r(X_s, U_s)\ ds + g_T(X_T) \mid X_t = x, \pi] \tag{54}$$

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

$$r(x,u) \stackrel{\text{def}}{=} g(x) - \frac{1}{2} u' q(x,t) u \tag{55}$$

In this case the HJB equation looks as follows

$$\begin{aligned}\frac{1}{\tau} v(x,t) = \max_u \Big\{ g(x) - \frac{1}{2} u' \tilde{q}(x,t) u + \frac{\partial v(x,t)}{\partial t} + a(x)' \frac{\partial v(x,t)}{\partial x} + u' b(x)' \frac{\partial v(x,t)}{\partial x} \\ + \frac{1}{2} \mathrm{Tr}\Big(c(x) c(x') \frac{\partial^2 v(x,t)}{\partial x^2} \Big) \Big\}\end{aligned} \tag{56}$$

where

$$\tilde{q}(x,t) = q(x,t) + w(x,t) \tag{57}$$

and

$$w(x,t)_{i,j} = -\mathrm{Tr}\Big(h_i(x) h_j(x)' \frac{\partial^2 v(x,t)}{\partial x^2} \Big) \tag{58}$$

Note that having noise proportional to the action is equivalent to having a state dependent quadratic cost on the action. The maximum over u can be computed analytically. Taking the gradient of the right hand side of (56) with respect to u and setting it to zero we get

$$-(\tilde{q}(x,t) u + b(x)' \frac{\partial v(x,t)}{\partial x} = 0 \tag{59}$$

Thus the optimal action is

$$\hat{u} = (\tilde{q}(x,t))^{-1} b(x)' \frac{\partial v(x,t)}{\partial x} \tag{60}$$

If $\tilde{q}$ is not full rank then there is an infinite number of optimal actions. We can choose one by using the pseudo-inverse of $\tilde{q}$. 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

$$\frac{1}{\tau}v(x,t) = g(x) - \frac{1}{2}\hat{u}'\tilde{q}(x,t)\hat{u} + \frac{\partial v(x,t)}{\partial t} + a(x)'\frac{\partial v(x,t)}{\partial x} + \hat{u}'\tilde{q}(x,t)\hat{u} + \frac{1}{2}\mathrm{Tr}[c(x)c(x)'\frac{\partial^2 v(x,t)}{\partial x^2}] \tag{61}$$

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

$$\boxed{\begin{array}{l} -\frac{\partial v(x,t)}{\partial t} = -\frac{1}{\tau}v(x,t) + g(x) + \frac{1}{2}\hat{u}'\tilde{q}(x,t)\hat{u} + \frac{\partial v(x,t)}{\partial x}'a(x) \\ \qquad + \frac{1}{2}\mathrm{Tr}[c(x)c(x)'\frac{\partial^2 v(x,t)}{\partial x^2}] \\ \hat{u}(x) = \tilde{q}^{-1}(x,t)b(x)'\frac{\partial v(x,t)}{\partial x} \\ \tilde{q}(x,t) = q(x,t) + w(x,t) \\ w(x,t)_{i,j} = \mathrm{Tr}[h_i(x)h_j(x)'\frac{\partial^2 v(x,t)}{\partial x^2}] \end{array}} \tag{62}$$

0.7 Linear Quadratic Tracker and Regulator

Let

$$dX_t = aX_t + bU_t + cdB_t \tag{63}$$

with

$$v(x,t) = E[\int_t^T r(X_s,U_s)e^{-\frac{1}{\tau}(s-t)}ds \mid X_t = x, \pi] \tag{64}$$

where

$$U_s = \pi(X_s) \tag{65}$$

$$r(x,u) = -(x-\xi)'p(x-\xi) - u'qu \tag{66}$$

where the target state ξ can be a function of time. This corresponds to the problem of having the state X_t track the trajectory ξ_t. We assume the value function takes the following form

$$v(x,t) = -\Big(x'\alpha_t x - 2\beta_t' x + \gamma_t\Big) \tag{67}$$

Thus,

$$\frac{\partial v(x,t)}{\partial x} = 2(\beta_t - \bar{\alpha}_t x) \tag{68}$$

$$\frac{\partial^2 v(x,t)}{\partial x^2} = -2\bar{\alpha}_t \tag{69}$$

$$\frac{\partial v(x,t)}{\partial t} = -x'\dot{\alpha}_t x + 2\dot{\beta}_t' x - \dot{\gamma}_t \tag{70}$$

where

$$\bar{\alpha}_t = \frac{\alpha_t + \alpha_t'}{2} \tag{71}$$

$$\dot{\alpha}_t = \frac{d\alpha_t}{dt} \tag{72}$$

$$\dot{\beta}_t = \frac{d\beta_t}{dt} \tag{73}$$

$$\dot{\gamma}_t = \frac{d\gamma_t}{dt} \tag{74}$$

Consider the optimal HJB equation (62)

$$\begin{aligned}-\frac{\partial v(x,t)}{\partial t} &= -\frac{1}{\tau}v(x,t) + g(x) + \hat{u}'q\hat{u} + \frac{\partial v(x,t)}{\partial x}' ax \\ &+ \frac{1}{2}\mathrm{Tr}[c(x)'c(x)\frac{\partial^2 v(x,t)}{\partial x^2}]\end{aligned} \tag{75}$$

where

$$g(x) = -(x-\xi)'p(x-\xi) \tag{76}$$

$$\hat{u}(x) = \frac{1}{2}q^{-1}b'\frac{\partial v(x,t)}{\partial x} = q^{-1}b'(\beta_t - \bar{\alpha}_t x) \tag{77}$$

The control law can be expressed as a standard feedback controller

$$\hat{u}(x) = k_t(\omega_t - x_t) \tag{78}$$

$$k_t = q^{-1}b'\bar{\alpha}_t \tag{79}$$

$$\omega_t = \bar{\alpha}_t^{-1}\beta_t \tag{80}$$

where $\bar{\alpha}_t^{-1}$ is the pseudoinverse of $\bar{\alpha}_t$, k_t is the feedback gain and ω_t is a virtual target state tracked by the feedback controller.

Thus

$$x'\dot{\alpha}_t x - 2\dot{\beta}_t' x + \dot{\gamma}_t = \frac{1}{\tau}x'\alpha_t x - \frac{2}{\tau}\beta_t' x + \frac{1}{\tau}\gamma - (x-\xi_t)'p(x-\xi_t) \tag{81}$$

$$+ (\beta_t - \bar{\alpha}_t x)' b q^{-1} b' (\beta_t - \bar{\alpha}_t x) \tag{82}$$

$$+ 2(\beta_t - \bar{\alpha}_t x)' a x - \mathrm{Tr}[c'c\bar{\alpha}_t] \tag{83}$$

Expanding some terms

$$x'\dot{\alpha}_t x - 2\dot{\beta}_t' x + \dot{\gamma}_t = \frac{1}{\tau}x'\alpha_t x - \frac{2}{\tau}\beta_t' x + \frac{1}{\tau}\gamma \tag{84}$$

$$- x'px + 2\xi_t' p x - \xi_t' p \xi_t \tag{85}$$

$$+ x'\bar{\alpha}_t b q^{-1} b' \bar{\alpha}_t x - 2\beta_t' b q^{-1} b' \bar{\alpha}_t x + \beta_t' b q^{-1} b' \beta_t \tag{86}$$

$$+ 2\beta_t' a x - 2x'\bar{\alpha}_t a x - \mathrm{Tr}[c'c\bar{\alpha}_t] \tag{87}$$

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

$$\boxed{\begin{array}{l}
\hat{u}_t(x) = k_t(\omega_t - x_t)\\
k_t = q^{-1} b' \bar{\alpha}_t\\
\omega_t = \bar{\alpha}_t^{-1} \beta_t\\
v(x,t) = -x'\alpha_t x + 2\beta_t' x - \gamma_t\\
\dot{\alpha}_t = \frac{1}{\tau}\alpha_t - p + \bar{\alpha}_t b q^{-1} b' \bar{\alpha}_t - 2\bar{\alpha}_t a\\
\dot{\beta}_t = -\frac{1}{\tau}\beta_t - p'\xi_t + \bar{\alpha}_t b q^{-1} b' \beta_t - a'\beta_t\\
\dot{\gamma}_t = \frac{1}{\tau}\gamma_t - \xi_t' p \xi_t + \beta_t' b q^{-1} b' \beta_t - \mathrm{Tr}[c' c \alpha_t]\\
\bar{\alpha}_t = (\alpha_t + \alpha_t')/2\\
\alpha_T = p_T\\
\beta_T = p_T\ \xi_T\\
\gamma_T = \xi_T' p_T \xi_T
\end{array}} \tag{88}$$

Alternatively the optimal action can be computed using the following procedure that does not require to take the pseudoinverse of $\bar{\alpha}_t$

$$\boxed{\hat{u}_t(x) = q^{-1} b' (\beta_t - \bar{\alpha}_t x_t)} \tag{89}$$

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 α, β, γ. Their values at time $t - \Delta_t$ can be obtained from those derivatives. We can then iterate until we reach the current time t. Below shows a simple example code.

```
function [omega, k, alpha beta, gamma]= ctfhlqt(xi,a,b,c,p,pT,q,tau,dt)

  s = length(xi);
  itau = 1/tau;

  qinv = pinv(q);
  qinvbt = qinv*b';

  alpha = pT;
  beta = pT*xi(:,s);
  gamma = xi(:,s)'*pT*xi(:,s);
  nu = length(q);
  nx = length(a);
  omega = zeros(nx,s);
  k2= zeros(nu,nx,s);
```

```
for t=s:-1: 1
  alphaBar= (alpha + alpha')/2;
  dalpha = alpha*itau -p + alphaBar*b*qinv*b'*alphaBar  - 2*alphaBar*a;
  dbeta = -beta*itau - p'*xi(:,t)+ alphaBar*b*qinv*b'*beta  - a'*beta;

  dgamma = - gamma*itau - xi(:,t)'*p*xi(:,t) + beta'*b*qinv*beta- ...
   trace(c'*c*alpha);
  omega(:,t) = pinv(alphaBar)*beta;
  k(:,:,t) = qinvbt*alphaBar;
  alpha = alpha - dt*dalpha;
  beta = beta - dt*dbeta;
  gamma = gamma - dt*dgamma;
end
```

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

$$\alpha_T = \frac{p + p'}{2} \tag{90}$$

$$\beta_T = 0 \tag{91}$$

$$\gamma_t = 0 \tag{92}$$

The update equation for β 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

$$\hat u_t(x) = -k_t x \tag{93}$$

$$k_t = q^{-1} b' \bar\alpha_t \tag{94}$$

$$\dot\alpha_t = \frac{1}{\tau}\alpha_t - p + \bar\alpha_t b q^{-1} b' \bar\alpha_t - 2\bar\alpha_t a \tag{95}$$

$$\dot\gamma_t = \frac{1}{\tau}\gamma_t - \mathrm{Tr}[c' c \alpha_t] \tag{96}$$

0.8 Feedback Linearization

Proposition 0.1. *Consider a process of the form*

$$dX_t = a_t X_t dt + b_t\, f(X_t, U_t, t) dt + c_t dB_t \tag{97}$$

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

$$h(x, f(x, u, t), t) = u \tag{98}$$

Let the instantaneous reward function take the following form

$$r(x,u,t) = -(\xi_t - x_t)' p_t (\xi_t - x_t) - f(x_t,u_t,t)' q_t f(x_t,u_t,t) \tag{99}$$

where ξ is a desired state trajectory. Then the following policy is optimal:

$$U_t = h(X_t, Y_t, t) \tag{100}$$

where Y_t is the solution to the following LQT control problem

$$dX_t = a_t X_t \, dt + b_t Y_t \, dt + c_t \, dB_t \tag{101}$$

Proof. Let the control process U be defined as follows

$$U_t = \pi(X_t, t) \tag{102}$$

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

$$Y_t = \lambda(X_t, t) = f(X_t, U_t, t) \tag{103}$$

Note π and λ are not independent: For every policy π there is a equivalent policy λ:

$$\lambda(X_t, t) = f(X_t, \pi(X_t,t), t) \tag{104}$$

Moreover for every virtual policy λ there is an equivalent policy π

$$U_t = \pi(X_t, t) = h(X_t, \lambda(X_t, t), t) \tag{105}$$

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

$$dX_t = aX_t dt + bY_t dt + cdB_t \tag{106}$$
$$r(x,y,t) = x' q_t x + y' g_t y \tag{107}$$

Let $\hat{\lambda}$ be the optimal policy mapping states to virtual actions, as found using the standard LQT algorithm on (106), (107). Let

$$\hat{\pi}(X_t, t) = h(X_t, \lambda(X_t, Y_t, t), t) \tag{108}$$

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

$$\lambda^*(X_t, t) = f(X_t, \pi^*(X_t, t), t) \tag{109}$$

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 $f(X_t, U_t, t)$.

0.9 Nonlinear Control

Here we present a recent approach to non-linear continuous time for the special case in 0.5. The approach is based on (**?**) but here we 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.,

$$v(x,t) = \phi(x)' w(t) = \sum_{i=1}^{n_f} \phi_i(x) w_i(t) \tag{110}$$

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

$$\frac{\partial v(x,t)}{\partial x} = \sum_{i=1}^{n_f} \dot{\phi}_i(x) w_i(t) = \dot{\phi}(x) w(t) \tag{111}$$

where

$$\dot{\phi}(x) \stackrel{\text{def}}{=} \nabla_x \phi(x) \tag{112}$$

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

$$\dot{\phi} = [\dot{\phi}_1, \cdots, \dot{\phi}_{n_f}] \tag{113}$$

Moreover

$$\frac{\partial^2 v(x,t)}{\partial x^2} = \sum_{i=1}^{n_f} w_i(t) \ddot{\phi}_i(x) \tag{114}$$

where ϕ_i is an $n_x \times n_x$ Hessian matrix

$$\ddot{\phi}_i(x) = \nabla_x^2 \phi_i(x) \tag{115}$$

Thus the HJB equation takes the following form

$$\begin{aligned} -\frac{\partial v(x,t)}{\partial t} =& -\frac{1}{\tau}\phi(x)' w(t) + g(x) + a(x)' \dot{\phi}(x) w(t) \\ &+ \frac{1}{4} w'(t) \dot{\phi}(x)' b(x) q^{-1} b(x)' \dot{\phi}(x) w(t) \\ &+ \frac{1}{4} \mathrm{Tr}[c(x)' c(x) \sum_{i=1}^{n_f} \ddot{\phi}_i(x) w_i(t)] \end{aligned} \tag{116}$$

Discretizing in time

$$\frac{\partial v(x,t)}{\partial t} = \frac{1}{\Delta t} v(x, t+\Delta t) - \frac{1}{\Delta t} v(x,t) \tag{117}$$

$$= \frac{1}{\Delta t} v(x, t+\Delta t) - \frac{1}{\Delta t} \phi(x)' w(t) \tag{118}$$

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

$$
\begin{aligned}
& g(x) + \frac{1}{\Delta t} v(x, t + \Delta t) \\
& + \Big(\dot\phi(x)' a(x) + h(x) - (\frac{1}{\tau} + \frac{1}{\Delta t})\phi(x)\Big)' w(t) \\
& + \frac{1}{2} w'(t)\dot\phi(x)' b(x) q^{-1} b(x)' \dot\phi(x) w(t) = 0
\end{aligned} \tag{119}
$$

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

$$
h_i(x) = \frac{1}{2}\mathrm{Tr}[c'(x)c(x)\ddot\phi_i(x)] \tag{120}
$$

This gives us as the key for an algorithm to find $v(x,t)$: If we knew $v(x, t + \Delta t)$, $\dot\phi(x)$, $\ddot\phi(x)$ we could search for values of $w(t)$ that satisfy (119).

If we have a explicit form for $g_T(x)$ then we just let $v(x,T) = g_T(x)$. Otherwise we just need to find a $w(T)$ such that

$$
\phi(x)' w(T) \approx g_T(x) \tag{121}
$$

from a sample of states $\{x^1, x^2, \cdots x^{n_s}\}$ where $x^i \in \Re^{n_x}$. These states can be chosen in any way we want. $w(T)$ can be found by solving the following linear regression problem

$$
\rho(w(T)) = \sum_{i=1}^{n_s} \Big[g_T(x) - \phi(x^i)' w(T)\Big]^2 \tag{122}
$$

For $t < T$ we choose either the same sample or a different of sample states in any way we want. We let the error at time t defined as follows

$$
\rho(w(t)) = \sum_{i=1}^{n_s} \Big[\mathbf{a}_i(t) + \mathbf{b}_i'(t) w(t) + w(t)' \mathbf{c}_i(t) w(t)\Big]^2 \tag{123}
$$

where

$$
\mathbf{a}_i(t) = g(x^i) + \frac{1}{\Delta t} v(x^i, t + \Delta t) \tag{124}
$$

$$
\mathbf{b}_i(t) = \dot\phi(x^i)' a(x^i) + h(x^i) - (\frac{1}{\tau} + \frac{1}{\Delta t})\phi(x^i) \tag{125}
$$

$$
\mathbf{c}_i(t) = \frac{1}{4}\dot\phi(x^i)' b(x^i) q^{-1} b(x^i)' \dot\phi(x^i) \tag{126}
$$

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 Δ_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 $\mathbf{a}_i(t)$ 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 s 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.

Requirements:

- $a(x), b(x)$ can be learned from examples using non-linear regression with error

$$e(x) = \Delta x - a(x)\Delta_t + b(x)u\Delta_t \tag{127}$$

- $c(x)$ can be obtained from model's error

$$c(x)c(x)' = \mathrm{Cov}(\Delta X/\Delta_t - a(X) - b(X)U) \tag{128}$$

- q the matrix for the quadratic error of the action.
- A way to sample from $g(x)$ and $g_T(x)$, the cost of the state.

0.9.1 Using Gaussian Radial Basis Functions

Gaussian functions centered at a fixed set of states $\mu^1 \cdots \mu^{n_f}$, and with fixed precision matrices ν_i can be used as feature functions, i.e.,

$$\phi_i(x) = \exp\big(-\frac{1}{2}(x^i - \mu^i)'\nu_i(x^i - \mu^i)\big) \tag{129}$$

where μ^i is a fixed n_x dimensional vector and ν_i is an $n_x \times n_x$ symmetric positive definite matrix. Thus in this case

$$\dot{\phi}_i(x) = \phi_i(x)\ \nu_i\ (\mu_i - x) \tag{130}$$

$$\ddot{\phi}(x) = \phi_i(x)\Big(\nu_i\ (x - \mu_i)\ (x - \mu_i)'\ \nu_i - \nu_i\Big) \tag{131}$$

1 Appendix

Lemma 1.1. *If $w_i \geq 0$ and $\hat{\beta}$ maximizes $f(i, \beta)$ for all i then*

$$\max_\beta \sum_i w_i f(i, \beta) = \sum_i w_i \max_\beta f(i, \beta) \tag{132}$$

Proof.

$$\max_\beta \sum_i w_i f(i, \beta) \leq \sum_i \max_\beta f(i, \beta) = \sum_i w_i f(i, \hat{\beta}) \tag{133}$$

moreover

$$\max_\beta \sum_i w_i f(i,\beta) \geq \sum_i f(i,\hat{\beta}) = \sum_i w_i \max_\beta f(i\beta) \tag{134}$$

□

Lemma 1.2. *If* $w_i \geq 0$ *and*

$$\max_\beta \sum_i w_i f(i,\beta) = \sum_i w_i \max_\beta f(i,\beta) \tag{135}$$

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

$$f(i,\hat{\beta}) = \max_\beta f(i,\beta) \tag{136}$$

Proof. Let

$$f(i,\hat{\beta}_i) = \max_\beta f(i,\beta) \tag{137}$$

and

$$f(i,\hat{\beta}) = \max_\beta \sum_i w_i f(i,\beta) \tag{138}$$

then

$$\sum_i w_i(f(i,\hat{\beta}_i) - f(i,\hat{\beta})) = 0 \tag{139}$$

Thus, since

$$f(i,\hat{\beta}_i) - f(i,\hat{\beta}) \geq 0 \tag{140}$$

it follows that

$$f(i,\hat{\beta}) = f(i,\hat{\beta}_i) = \max_\beta f(i,\beta) \tag{141}$$

for all i such that $w_i > 0$. □

Lemma 1.3 (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*

$$\rho(x) = E[(bx - C)'a(bx - C)] + x'd\,x \tag{142}$$

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*

$$\begin{aligned} J_x\rho &= E[J_{bx-C}(bx - C)'a(bx - C)\,J_x(bx - C)] + J_x x'd\,x && (143)\\ &= 2E[(bx - C)'ab] + 2x'd && (144)\\ \nabla_x\rho &= (J_x)' = 2b'a(bx - \mu) + 2d\,x && (145) \end{aligned}$$

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

$$(b'ab + d)x = b'a\mu \tag{146}$$

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

$$\hat{x} = h\mu \tag{147}$$

where

$$h = (b'ab + d)^{-1}b'a \tag{148}$$

Moreover

$$\rho(\hat{x}) = (bh\mu - C)'a(bh\mu - C) + \mu'h'dh\mu \tag{149}$$

$$= \mu'h'b'abh\mu - 2\mu'h'b'a\mu + E[C'aC] + \mu'h'dh\mu \tag{150}$$

Now note

$$\mu'h'b'abh\mu + \mu'h'dh\mu = \mu'h'(b'ab + d)h\mu \tag{151}$$

$$= \mu'a'b(b'ab + d)^{-1}(b'ab + d)(b'ab + d)^{-1}b'a\mu \tag{152}$$

$$= \mu'a'b(b'ab + d)^{-1}b'a\mu \tag{153}$$

$$= \mu'h'b'a\mu \tag{154}$$

Thus

$$\rho(\hat{x}) = E[C'aC] - \mu'h'b'a\mu \tag{155}$$

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

$$\rho(\hat{x}) = c'ac - c'h'b'ac = c'kc \tag{156}$$

where

$$k = a - h'b'a = a - a'b(b'ab + d)^{-1}b'a \tag{157}$$

For the more general case it is sometimes useful to express (155) *as follows*

$$\rho(\hat{x}) = E[C'aC] - \mu'h'b'a\mu = E[C'(a - h'b'a)C] + E[(C - \mu)'h'b'a(C - \mu)] \tag{158}$$

Lemma 1.4 (Quadratic Regression)**.** *We want to minimize*

$$\rho(w) = \sum_i \Big(a_i + b_i'w + w'c_iw\Big)^2 \tag{159}$$

where a_i is a scalar, b_i, w are n-dimensional vectors and c_i an $n \times n$ symetric matrix[1]*. We solve the problem iteratively starting at a weight vector w_k linearizing the quadratic part of the function and iterating.*

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

Linearizing about w_k we get

$$\begin{aligned} w'c_iw &\approx w_k'c_iw_k + 2w_k'c_i(w - w_k) \\ &= -w_k'c_iw_k + 2w_k'c_iw \end{aligned} \tag{160}$$

Thus

$$a_i + b_i'w + w'c_iw \approx a_i - w_k'c_iw_k + (b_i + 2c_iw_k)'w \tag{161}$$

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

$$y_i = -a_i + w_k'c_iw_k \tag{162}$$

and predicting variables into a matrix x with rows

$$x_i = (b_i + 2c_iw_k)' \tag{163}$$

with

$$w_{k+1} = (x'x)^{-1}x'y \tag{164}$$