1 Linear-Quadratic-Gaussian Optimal Control Problem

Imagine that we have a problem of the following form:

$$\max_{\{a_t,x_t\}_{t=0}^{\infty}} \sum_{t=0}^{\infty} \beta^t \left(x_t^T Q x_t + a_t^T R a_t + 2a_t^T W x_t\right) \tag{1}$$

where x_t is a vector of states and a_t is a vector of controls (actions). Assume that there are n states where the first one is the constant 1 and there are k controls – the matrices are then dimensioned as $Q_{n\times n}$, $R_{k\times k}$, and $W_{n\times k}$. States evolve according to the transition function

$$x_{t+1} = Ax_t + Ba_t. \tag{2}$$

Basically, we are maximizing a quadratic objective with linear constraints. The Bellman equation is

$$x_t^T P x_t = \max_{a_t} \left\{x_t^T Q x_t + a_t^T R a_t + 2a_t^T W x_t + \beta x_{t+1}^T P x_{t+1}\right\} \tag{3}$$

subject to

$$x_{t+1} = Ax_t + Ba_t. \tag{4}$$

I have used the fact that the value function will be quadratic; it is trivial to show this fact holds via guess and verify. The value function, $x^T Px$, is symmetric (the P matrix is symmetric; from now on I will abuse notation a bit and refer to P as the value function). Substituting the constraint into the right-hand-side of the Bellman equation yields

$$x^T Qx + a^T Ra + 2a^T Wx + \beta \left(x^T A^T + a^T B^T\right) P \left(Ax + Ba\right). \tag{5}$$

Maximizing with respect to a yields the solution[1]

$$a = -(R + \beta B^T PB)^{-1}(W + \beta B^T PA)x. \tag{6}$$

Substituting this into the Bellman equation and matching coefficients yields the recursive relationship known as the **matrix Ricatti equation**:

$$x^T Px = x^T(Q + \beta A^T PA - (W^T + \beta A^T PB)(R + \beta B^T PB)^{-1}(W + \beta B^T PA))x. \tag{7}$$

[1] This expression uses the matrix differentiation rules that

$$D_x \left[y^T Ax\right] = A^T y$$
$$D_y \left(y^T Ax\right) = Ax.$$

This equation can be used to solve for the value function by iteration. Think of the RHS as the result of the Bellman operator; it takes the function characterized by P and maps it into one characterized by

$$Q + \beta A^T PA - (W^T + \beta A^T PB)(R + \beta B^T PB)^{-1}(W + \beta B^T PA). \quad (8)$$

One way to solve this equation for P is to iterate: start with $P_0 = 0$, calculate a new P_1 by

$$P_1 = Q + \beta A^T P_0 A - (W^T + \beta A^T P_0 B)(R + \beta B^T P_0 B)^{-1}(W + \beta B^T P_0 A) \quad (9)$$

and repeat until the P are sufficiently close together. There are faster ways to solve this equation, but none which are so easily seen in light of our discussion of the value function – all we are doing here is applying the Bellman operator successively to our linear-quadratic problem. The Contraction Mapping Theorem guarantees that only one matrix P can solve the matrix Ricatti equation.

The optimal decision rules are then recovered by

$$\begin{aligned} a^* &= -Fx \\ &= -(R + \beta B^T PB)^{-1}(W + \beta B^T PA)x. \end{aligned}$$

In Matlab, the program would look like this:

1. Set initial P_0 (a good choice is either the zero matrix or a matrix which is a small negative number times an identity matrix);
2. Compute P_1 according to the Ricatti equation;
3. Compute the norm of $P_1 - P_0$ using the built-in 'norm' function;
4. Set up a 'while' loop: while norm$(P_1 - P_0) > \epsilon$ set $P_0 = P_1$ and recompute P_1.

The while loop will continue until the difference between the two matrices is small. After the loop terminates you can compute F.

Some minor points are of some importance here. We require that the value function be concave; this condition was needed to ensure differentiability and many other desirable features such as the transversality condition. In the quadratic case, we need P to be negative definite – it must possess only negative eigenvalues (except for one related to a constant, which typically will be large and positive).[2] We also want our system to be saddle-stable; this will require that the matrix $A - BF$ have eigenvalues within the unit circle since

$$\begin{aligned} x_{t+1} &= Ax_t + Ba_t^* \\ &= (A - BF)x_t. \end{aligned}$$

[2] A matrix A is called **negative definite** if and only if $x^T Ax < 0 \; \forall x \neq 0$. The eigenvalue condition is necessary and sufficient for this condition.

Thus, iterating on the decision rule produces a sequence that converges to the steady state, so transversality is satisfied.

In general the growth model is not quadratic. Therefore, now we will examine how to make it so – this is the approach used by Kydland and Prescott in their time-to-build paper and first discussed in the appendix to their rules vs. discretion paper. The growth model has a Bellman equation like this:

$$v\left(k\right)=\max_{k'}\{u(f(k)+(1-\delta)k-k')+\beta v(k')\}. \tag{10}$$

Let's take the steady state of this model to be $\overline{k}$. We know that the model is saddle-stable, at least locally, around this point – it will tend to this value over time. Therefore, what if we choose to approximate the utility function around this point by a quadratic? We know from before that this steady state is given by the solution to

$$f'(\overline{k})+1-\delta=1/\beta. \tag{11}$$

From Taylor's theorem we have

$$\begin{aligned} u(f(k)+(1-\delta)k-k') &= u(f(\overline{k})+(1-\delta)\overline{k}-\overline{k})+ \\ & u'(f(\overline{k})+(1-\delta)\overline{k}-\overline{k})(f'(\overline{k})+1-\delta)(k-\overline{k})+ \\ & u'(f(\overline{k})+(1-\delta)\overline{k}-\overline{k})(k'-\overline{k})+ \\ & \text{second order terms.} \end{aligned}$$

We will make the compute calculate this for us; otherwise, the higher-order terms become quite tedious. To do so, we will use the definition of a derivative:

$$f'(x)=\lim_{h\to 0}\frac{f(x+h)-f(x)}{h}. \tag{12}$$

This is called a "one-sided derivative" and to calculate it we would choose a small h – somewhere around $10^{-6}x$. But in general this is not very accurate – so we will employ the two-sided derivative

$$f'(x)=\lim_{h\to 0}\frac{f(x+h)-f(x-h)}{2h}. \tag{13}$$

This will be much more accurate. The second derivative is then

$$\begin{aligned} f''(x) &= \lim_{h\to 0}\frac{f'(x+h)-f'(x-h)}{2h} \\ &= \frac{\frac{f(x+2h)-f(x)}{2h}-\frac{f(x)-f(x-2h)}{2h}}{2h} \\ &= \frac{f(x+2h)-2f(x)+f(x-2h)}{4h^2} \end{aligned}$$

where h should be around $10^{-3}x$. For a function of more than one variable, we calculate partial derivatives by

$$\frac{\partial f(x,y)}{\partial x}=\frac{f(x+h_1,y)-f(x-h_1,y)}{2h_1} \tag{14}$$

$$\frac{\partial f(x,y)}{\partial y} = \frac{f(x,y+h_2) - f(x,y-h_2)}{2h_2} \tag{15}$$

$$\begin{aligned}
\frac{\partial^2 f(x,y)}{\partial x \partial y} &= \frac{1}{2h_1}\left[\frac{f(x+h_1,y+h_2) - f(x+h_1,y-h_2)}{2h_2} - \frac{f(x-h_1,y+h_2) - f(x-h_1,y-h_2)}{2h_2}\right] \\
&= \frac{f(x+h_1,y+h_2) - f(x+h_1,y-h_2) - f(x-h_1,y+h_2) + f(x-h_1,y-h_2)}{4h_1h_2}
\end{aligned}$$

where h_1, h_2 should be around $10^{-3}x$ or $10^{-3}y$. In general one should avoid computing higher-order derivatives in this manner, but we don't need those terms here.

Our goal is to approximate the return function

$$u(f(k) + (1-\delta)k - k') \equiv H(y) \tag{16}$$

where

$$y = \begin{bmatrix} k \\ k' \end{bmatrix} = \begin{bmatrix} x \\ a \end{bmatrix} \tag{17}$$

where we do not augment x with a constant. Let the steady state be denoted $\overline{y}$. Then the approximation around the steady state yields

$$H(y) = H(\overline{y}) + f^T(y-\overline{y}) + (y-\overline{y})^T\frac{1}{2}V(y-\overline{y}). \tag{18}$$

Now define

$$\frac{1}{2}V = \begin{bmatrix} S & T^T \\ T & R \end{bmatrix} \tag{19}$$

$$l_k^T = \frac{1}{2}[f_k^T - 2\overline{x}^T T^T - 2\overline{a}^T R] \tag{20}$$

$$l_n^T = \frac{1}{2}[f_n^T - 2\overline{x}^T S - 2\overline{a}^T T] \tag{21}$$

$$G = H(\overline{y}) + \overline{y}^T\frac{1}{2}V\overline{y} - f^T\overline{y} \tag{22}$$

where

$$f = \begin{bmatrix} f_n \\ f_k \end{bmatrix} \tag{23}$$

where f_n refers to the first derivatives with respect to states and f_k to controls. Now define

$$W = [l_k \quad T] \tag{24}$$

and

$$Q = \begin{bmatrix} G & l_n^T \\ l_n & S \end{bmatrix}. \tag{25}$$

From here we can use the Ricatti equation above to find P and F. Q now embeds the constant term G as well as the linear terms associated with states times the constant.

We can produce a log-quadratic-linear approximation by replacing each variable by its "exponential" and then computing the same expressions – thus, instead of x we have $\exp(x)$ and instead of a we have $\exp(a)$. Then x and a are interpreted to be logs. The value of making these substitutions is that for some problems the approximations are better in the sense of reducing sup-norm approximation errors.

Note: if you are using a programming environment that manipulates symbolic expressions – such as Mathematica or Maple, which comes embedded in Matlab – you can actually create the derivative terms symbolically. To do so, you define all the expressions as 'syms' and then use the 'diff' command. For most problems the numerical approach is sufficiently accurate.

Adding uncertainty in this environment is very simple, because it satisfies something called certainty equivalence. Imagine the following two problems:

$$\max_a E\left[f(x,a)\right] \tag{1}$$

where x is a random variable and

$$\max_a f(E(x),a) \tag{2}$$

where all the randomness is gone. Under what conditions are these two problems equivalent? By equivalent, I mean that they generate the same α; the value functions for the two problems will differ by a constant. We state and prove the following theorem due to Simon (1948):

Theorem 1 ***(Certainty Equivalence Theorem)*** *If f is quadratic in x and a, these two problems will have the same a^* where*

$$a^* = \operatorname*{argmax}_a \left\{E\left[f\left(x,a\right)\right]\right\} = \operatorname*{argmax}_a \left\{f\left(E\left(x\right),a\right)\right\}.$$

Proof. For problem (1) the first-order conditions are

$$\frac{\partial}{\partial\alpha}E\left[f(x,a)\right] = 0. \tag{26}$$

Assume that

$$f(x,a) = A + Bx + Ca + Dx^2 + Fa^2 + Gxa. \tag{27}$$

Then the first-order condition implies

$$E\left[c + 2Fa + Gx\right] = 0 \Rightarrow c + 2Fa + GE(x) = 0. \tag{28}$$

For problem (2) the first-order conditions are

$$\frac{\partial}{\partial x}f(E(x),a) = 0 \Rightarrow c + 2Fa + GE(x) = 0. \tag{29}$$

Thus, the problems are equivalent. ■

We can use the certainty equivalence principle to derive the **Ricatti equation** for a quadratic problem with uncertainty. The Bellman equation is

$$v(x_t) = \max_{a_t} \left\{ x_t^T Q x_t + a_t^T R a_t + 2a_t^T W x_t + \beta E_t[v(x_{t+1})|x_t] \right\} \tag{30}$$

subject to the transition equation

$$x_{t+1} = Ax_t + Ba_t + \epsilon_{t+1} \tag{31}$$

where $\epsilon_t\tilde{}N(0,\Sigma)$ is a vector of random variables. The form of the value function is now

$$x_t^T P x_t + d \tag{32}$$

where d is a constant. Substituting into the right-hand-side of the Bellman equation and taking expectations yields

$$\begin{aligned} E_t(x_{t+1}^T P x_{t+1} + d) &= E_t\left[(Ax_t + Ba_t + \epsilon_{t+1})^T P (Ax_t + Ba_t + \epsilon_{t+1}) + d\right] \\ &= E_t\left[\cdot\right] + +E_t\left[a_t B^T P \epsilon_{t+1}\right] + E_t\left[\epsilon_t^T P A x_t\right] + E_t\left[\epsilon_t^T P B a_t\right] + \\ &\quad E_t\left[x_{t+1}^T A^T P \epsilon_{t+1}\right] + E_t\left[\epsilon_{t+1}^T P \epsilon_{t+1}\right] + d \end{aligned}$$

where $[\cdot]$ consists of terms that do not involve ϵ_{t+1}. This implies

$$E_t(x_{t+1}^T P x_{t+1}) = [\cdot] + E_t\left[\epsilon_{t+1}^T P \epsilon_{t+1}\right]. \tag{33}$$

Clearly we will have the same decision rules as before. The matrix Ricatti equation is then identical to the one before. Matching up constants yields

$$d = \beta d + \beta E_t\left[\epsilon_{t+1}^T P \epsilon_{t+1}\right] \Rightarrow d = \frac{\beta}{1-\beta}\,\text{trace}(P\Sigma). \tag{34}$$

Since P is negative definite this term is negative; uncertainty about the future lowers lifetime utility. What we have done here is shown that the problem "separates" forecasting – the forming of expectations – and control – the optimal decisions; that is the essence of certainty equivalence. Forecast first and then make decisions based on the conditional expected value of future variables without regard to the risk associated with realizations differing from those conditional expectations. Note: because we have added a constant to the state vector, d is not the only constant in the value function.

It is possible to extend the linear quadratic regulator problem to break the certainty equivalence property. Hansen and Sargent (1995) formulate a model with "risk sensitivity" which alters the expectations of the household in a way that lets Σ, the variance-covariance matrix of the stochastic terms, affect decisions but keeps the problem quadratic. The distortion takes the form of an operator D that alters V:

$$D(V) = V - \sigma V \Sigma \left(I + \sigma \Sigma' V \Sigma\right)^{-1} \Sigma' V \tag{35}$$

where σ is a preference parameter with the property that if σ increases the risk aversion of the agent with respect to gambles over future wealth also increases.

Note that if $\sigma = 0$ (agents obey certainty equivalence) or $\Sigma = 0$ (the model is deterministic) this expression collapses to V. Assuming that the return function is quadratic the Bellman equation is

$$V(x) = \max_a \left\{ x^T Q x + a^T R a + 2x^T W a + \beta \left(\frac{2}{\sigma}\right) \log \left(E_t \left[\exp\left(-\frac{\sigma}{2} V(x')\right) \Big| x \right] \right) \right\};$$

this equation is a particular form of Epstein-Zin preferences – the critical part is that the elasticity of intertemporal substitution is fixed at 1 while risk aversion can varies with σ.

The Bellman equation can be computed by evaluating the expectation under the assumption of normal errors and linear state transition equations:

$$x^T P x + d = \max_a \left\{ x^T Q x + a^T R a + 2x^T W a + \beta \left(x^T A^T + a^T B^T \right) D(P) (Ax + Ba) + U(P, d) \right\} \quad (36)$$

where U is some function that does not involve a. The optimal control is

$$a = -(R + \beta B^T D(P) B)^{-1} (W + \beta B^T D(P) A) x. \quad (37)$$

Substitution back into the Bellman equation yields the Ricatti equation

$$x^T P x + d = x^T (Q + \beta A^T D(P) A - (W^T + \beta A^T D(P) B)(R + \beta B^T D(P) B)^{-1} (W + \beta B^T D(P) A)) x + U(P, d). \quad (38)$$

This expression again defines a Contraction Mapping so iterations converge to a unique P. Since Σ appears in the expression for $D(P)$, agents in this economy change the coefficients of their linear decision rules in response to different variance-covariance matrices for the shocks.

Since the economy no longer displays certainty equivalence, the deterministic and stochastic steady states are no longer the same. As a result, the decision rules, if centered at the deterministic steady state, may be inaccurate around the average values, particularly for large σ and Σ. To center the approximation around the stochastic steady state involves solving the system of equations

$$x^* = (A - BF(\overline{x})) x^*$$

for x^*, where $F(\overline{x})$ is the decision rule obtained when the quadratic approximation is taken around $\overline{x}$. This equation can be solved using eigenvalue methods, since x^* is just the eigenvector associated with the unit eigenvalue of $A - BF(\overline{x})$; if the solution yields $x^* = \overline{x}$ then we have located the stochastic steady state. An efficient way to solve for $\overline{x}$ is to embed the entire program inside an fsolve driver file, where the function value is the difference between $\overline{x}$ and x^*.

Interpreting the deviation from certainty equivalence seems straightforward, but it seems there are many ways to do so. One interpretation is the one used above – sensitivity to risk. But apparently it can also be interpreted as a concern for model misspecification. For those interested, Hansen and Sargent have a whole book – called Robustness – dedicated to interpreting this problem.