Linear - Quadratic Dynamic Programming

Problem:

$$\underset{u_t}{Max} \sum_{t=0}^{\infty} \beta^t [x_t' Q x_t + u_t' R u_t + 2x_t' W u_t]$$

subject to: $x_{t+1} = Ax_t + Bu_t$

$x_t \ : \ (n \times 1)$

$u_t \ : \ (k \times 1)$

$R \ : \ (n \times n)$, negative semidefinite, symmetric.

$Q \ : \ (k \times k)$, negative definite, symmetric.

$W \ : \ (n \times k)$

$A \ : \ (n \times n)$

$B \ : \ (n \times k)$

Vector - Matrix Transposition rules:

$$(GH)' = H'G'$$
$$[(GH)^{-1}]' = [H'G']^{-1}$$

For scalars, $k' = k$

Vector - Matrix differentiation rules:

$$\frac{\partial x' G x}{\partial x} = (G + G')x = 2Gx, \text{ if } G \text{ is symmetric.}$$
$$\frac{\partial y' H z}{\partial y} = Hz$$
$$\frac{\partial y' H z}{\partial z} = H' y$$

Bellman Equation:

$$V(x_t) = \underset{u_t}{Max} \ ([x_t' Q x_t + u_t' R u_t + 2x_t' W u_t] + \beta V(x_{t+1}))$$

Guess:

$$V(x_t) = x_t' P x_t, P \text{ symmetric.}$$

Plug in for $V(x_{t+1})$ with $x_{t+1}' P x_{t+1}$:

$$\begin{aligned} RHS &= x_t' Q x_t + u_t' R u_t + 2x_t' W u_t \\ &\quad + \beta (Ax_t + Bu_t)' P (Ax_t + Bu_t) \\ &= x_t' Q x_t + u_t' R u_t + 2x_t' W u_t \\ &\quad + \beta [x_t' A' P A x_t + x_t' A' P B u_t + u_t' B' P A x_t + u_t' B' P B u_t] \end{aligned}$$

As scalars, $x_t'A'PBu_t = u_t'B'P'Ax_t = u_t'B'PAx_t$.
Therefore:

$$\begin{aligned} RHS &= x_t'Qx_t + u_t'Ru_t + 2x_t'Wu_t \\ &\quad + \beta[x_t'A'PAx_t + 2u_t'B'PAx_t + u_t'B'PBu_t] \end{aligned}$$

Note that $R = R'$, symmetric.
Also $(B'PB)' = B'P'B = B'PB$, as P is symmetric.
Now take derivative of RHS with respect to u_t.

$$\begin{aligned} \frac{\partial RHS}{\partial u_t} &= 2Ru_t + 2W'x_t \\ &\quad + 2\beta B'PAx_t + 2\beta B'PBu_t \end{aligned}$$

Set equal to zero and rearrange:

$$(R + \beta B'PB)u_t = -(W' + \beta B'PA)x_t$$

$$u_t^* = -(R + \beta B'PB)^{-1}(W' + \beta B'PA)x_t$$

Equivalently:

$$\begin{aligned} (u_t^*)' &= -x_t'(W' + \beta B'PA)'[(R + \beta B'PB)^{-1}]' \\ &= -x_t'(W + \beta A'P'B)(R' + \beta B'P'B)^{-1} \end{aligned}$$

Again, $P = P'$ and $R = R'$, and so:

$$(u_t^*)' = -x_t'(W + \beta A'PB)(R + \beta B'PB)^{-1}$$

Return to Bellman equation:

$$\begin{aligned} V(x_t) &= x_t'Qx_t + (u_t^*)'Ru_t^* + 2x_t'Wu_t^* \\ &\quad + \beta\left[x_t'A'PAx_t + 2(u_t^*)'B'PAx_t + (u_t^*)'B'PBu_t^*\right] \end{aligned}$$

The term, $x_t'Wu_t^* = (u_t^*)'W'x_t$ is a scalar, and so:

$$\begin{aligned} V(x_t) &= x_t'[Q + \beta A'PA]x_t + (u_t^*)'[R + \beta B'PB]u_t^* \\ &\quad + 2(u_t^*)'[W' + \beta B'PA]x_t \end{aligned}$$

Now plug in for u_t^*, $(u_t^*)'$:

$$\begin{aligned} &(u_t^*)'[R + \beta B'PB]u_t^* \\ &= -x_t'(W + \beta A'PB)(R + \beta B'PB)^{-1}(R + \beta B'PB) \\ &\quad \times (-1)(R + \beta B'PB)^{-1}(W' + \beta B'PA)x_t \end{aligned}$$

Simplify to:

$$\begin{aligned} &(u_t^*)'[R + \beta B'PB]u_t^* \\ &= x_t'\left[(W + \beta A'PB)(R + \beta B'PB)^{-1}(W' + \beta B'PA)\right]x_t \end{aligned}$$

Next evaluate:

$$2(u_t^*)'[W' + \beta B'PA]x_t$$
$$= -2x_t'(W + \beta A'PB)(R + \beta B'PB)^{-1}(W' + \beta B'PA)x_t$$

Therefore:

$$x_t'Px_t = x_t'[Q + \beta A'PA]x_t$$
$$+ x_t'(W + \beta A'PB)(R + \beta B'PB)^{-1}(W' + \beta B'PA)x_t$$
$$- 2x_t'(W + \beta A'PB)(R + \beta B'PB)^{-1}(W' + \beta B'PA)x_t$$
$$= x_t'[Q + \beta A'PA]x_t$$
$$- x_t'(W + \beta A'PB)(R + \beta B'PB)^{-1}(W' + \beta B'PA)x_t$$

Finally:

$$x_t'Px_t = x_t'[Q + \beta A'PA - (W + \beta A'PB)$$
$$(R + \beta B'PB)^{-1}(W' + \beta B'PA)]x_t$$

Rearrange as:

$$P = [Q + \beta A'PA - (W + \beta A'PB)$$
$$(R + \beta B'PB)^{-1}(W' + \beta B'PA)]$$

A Riccati equation in P.
Solve iteratively:

- Guess RHS P.
- Calculate LHS P.
- Use this P as next guess, etc.
- Should converge to the correct P.

The program olrp.m converges more quickly.
Method is comparable.
Original notation (A, B, Q, R, W) conforms to olrp.m
Syntax: $[F, P] =$olrp(β, A, B, Q, R, W)
P is the matrix for the value function.
F gives the feedback rule.
Note that solution is of the form:

$$u_t^* = -Fx_t, \text{ where}$$
$$F = (R + \beta B'PB)^{-1}(W' + \beta B'PA)$$

The solution is therefore linear in x_t.
Behavior of x_t following optimal rule:

$$x_{t+1} = Ax_t + Bu_t$$
$$= (A - BF)x_t$$

Therefore:

$$x_{t+j} = (A - BF)^j x_t$$

For given initial condition, x_0 :

$$x_t = (A - BF)^t x_0$$

Steady State:

$$\bar{x} = \lim_{t \to +\infty} (A - BF)^t x_0$$

Typically $\bar{x}$ is independent of x_0.

Stochastic Linear - Quadratic Programming

Now amend the original problem as follows:

$$\underset{u_t}{Max} \sum_{t=0}^{\infty} \beta^t [x_t' Q x_t + u_t' R u_t + 2x_t' W u_t]$$

subject to: $x_{t+1} = Ax_t + Bu_t + C\varepsilon_{t+1}$

$$E(\varepsilon_t) = 0,\ E(\varepsilon_t \varepsilon_{t+i}') = 0 \forall i \neq 0,\ E(\varepsilon_t \varepsilon_t') = I$$

The individual ε_t's are normally distributed.
Now guess:

$V(x_t) = x_t' P x_t + d$, d a scalar constant.

Redo derivation to confirm that:

$$\begin{aligned} P = [&Q + \beta A' P A - (W + \beta A' P B) \\ &(R + \beta B' P B)^{-1} (W' + \beta B' P A)] \end{aligned}$$

Certainty Equivalence: same P as above.
Therefore policy rule: $u_t = -Fx_t$ unchanged.
We may also confirm that:

$$d = \beta(1 - \beta)^{-1} tr(PCC')$$

Uncertainty matters for V, but not for F.

Linear - Quadratic Approximations

Consider the problem:

$$\underset{u_t}{Max} \sum_{t=0}^{\infty} \beta^t r(x_t, u_t)$$

$$\text{subject to: } x_{t+1} = Ax_t + Bu_t + C\varepsilon_{t+1}$$

Set up x_t such that $x_{1t} = 1$.

$$x_t : (n \times 1)$$
$$u_t : (k \times 1)$$

All nonlinearities subsumed in $r(x_t, u_t)$.
Now define:

$$z_t \equiv \begin{bmatrix} x_t \\ u_t \end{bmatrix}$$

Approximation:

$$\hat{r}(z_t) \approx r(\bar{z}) + (z_t - \bar{z})' \frac{\partial r}{\partial z} + \frac{1}{2}(z_t - \bar{z})' \frac{\partial^2 r}{\partial z \partial z'}(z_t - \bar{z})$$

Now find M such that:

$$r(z_t) \approx (z_t' M z_t)$$

Define the vector:

$$e = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} : (n+k, 1)$$

We may now confirm that:

$$\begin{aligned} M = {} & e\left[r(\bar{z}) - \left(\frac{\partial r}{\partial z} \right)' \bar{z} + \frac{1}{2} \bar{z}' \frac{\partial^2 r}{\partial z \partial z'} \bar{z} \right] e' \\ & + \frac{1}{2}\left[\frac{\partial r}{\partial z} e' - e\bar{z}' \frac{\partial^2 r}{\partial z \partial z'} - \frac{\partial^2 r}{\partial z \partial z'} \bar{z} e' + e\left(\frac{\partial r}{\partial z} \right)' \right] \\ & + \frac{1}{2}\left(\frac{\partial^2 r}{\partial z \partial z'} \right) \end{aligned}$$

All derivatives evaluated at $\bar{z}$.
Finally, partition M to conform with original notation:

$$\begin{aligned} M &= \begin{bmatrix} M_{11}(n \times n) & M_{21}(n \times k) \\ M_{21}(k \times n) & M_{22}(k \times k) \end{bmatrix} \\ &\equiv \begin{bmatrix} Q & W \\ W' & R \end{bmatrix} \end{aligned}$$