A Graph to Illuminate Robust Control

1 Note to Chase and Spencer

I recommend that you start by reading section 2 and then just jump to 8 that describe some Matlab programs that could form the heart of the calculations. The other sections are background.

2 The graph

Think of a graph with values on the coordinate (y) axis and discounted entropy on the ordinate (x) axis. By value, we mean a value function evaluated at a particular state vector at time 0, y_0. We want to compute *sets* of values attained by two decision rules, one non robust or 'optimal' decision rule, another 'robust'. View the discounted entropy as a constraint that either a minimizing or a maximizing second player faces in choosing a perturbation of the approximating model. (We'll use negative values of θ below to induce our computer programs solve max-max problems.) For a given decision rule, the maximized value function at a given entropy level defines the upper bound of the set, while the minimized value function at a given entropy level defines the lower bound of the set. For a given decision rule, the set of possible values depicts the range of values that are possible at different entropy levels. At zero entropy, only one value is attained because there is only one probability measure or model.

What I expect and hope the graphs to show is

- The non-robust rule has a set that is higher than a robust rule for small entropy levels.
- A robust rule has a skinnier set and eventually overlaps and overtakes big parts of the set of values attained by the non-robust rule. I hope

that the 'levels' and 'fatnesses' of the sets are revealing. But who knows?

The remainder of this note describes how to compute the graph for an LQ model.

3 Boilerplate linear regulator

Let $r(y,a) = -(y'Qy + a'Ra)$. A decision maker wants to choose a plan for a control vector a_t to maximize

$$E_0 \sum_{t=0}^{\infty} \beta^t r(y_t, a_t)$$

subject to $y_{t+1} = Ay_t + Ba_t + C\epsilon_{t+1}$ where A is a square matrix, B and C are matrices, y_t is a vector of states, a_t is a vector of controls, and ϵ_{t+1} is an i.i.d. sequence of random vectors with $\epsilon_{t+1} \sim \mathcal{N}(0, I)$. We want to consider the situation in which the decision maker does not completely trust the law of motion for y and therefore wants decision rules that are robust to misspecifications of it.

A robust LQ control problem is pinned down by the list $(A, B, C, Q, R, \beta, \theta)$ where $\beta \in (0,1)$ and $\theta \in (\underline{\theta}, +\infty]$ in the case of a robustness problem or $\theta \in [-\infty, \underline{\theta})$ in the case of an 'optimistic control problem' that we'll want to solve below.

4 Some operators and formulas

4.1 Ordinary non-robust control problem

$$\mathcal{B}(P) = Q + \beta A'PA - \beta^2 A'PB(R + \beta B'PB)^{-1}B'PA \tag{1}$$

is the operator associated with the right side of the *ordinary* Bellman equation $-\frac{1}{2}y'Py = \max_a \left\{ r(y,a) - \beta\frac{1}{2}(Ay + Ba)'P(Ay + Ba) \right\}$. The optimal value function is pinned down by the fixed point

$$P = \mathcal{B}(B).$$

The maximizing decision rule is $a = -Fy$ where

$$F = \beta(R + \beta B'PB)^{-1}B'PA. \tag{2}$$

4.2 Finding a worst-case mean distortion

Let $A_F = A - BF$ for a fixed decision rule F. Another key operator

$$\mathcal{D}(P) = P + PC(\theta I - C'PC)^{-1}C'P$$

is associated with the problem

$$\min_w \left\{ -\frac{1}{2}(A_F y + Cw)'P(A_F y + Cw) + \frac{\theta}{2}w'w \right\}.$$

In this deterministic problem, we penalize the choice of the shock mean distortion w using only the contribution to relative entropy that comes from what in the stochastic problem is the mean distortion w. The minimizing w is

$$w^* = (\theta I - C'PC)^{-1} C'PA_F y.$$

This coincides with the mean distortion of the worst-case normal distribution for the stochastic problem. The minimized objective function is

$$-\frac{1}{2}(A_F y)'\mathcal{D}(P)(A_F y),$$

which agrees with the contribution to the stochastic robust adjustment to the value function coming from the quadratic form in $A_F y$. (All that is missing relative to the stochastic problem is the distorted covariance matrix for the worst-case normal distribution and the constant term in the adjusted value function. The worst-case shock's covariance matrix is $\left(I - \frac{1}{\theta}C'PC\right)^{-1}$ rather than I.)

4.3 Robust control formulas

Consider the fixed point

$$P = \mathcal{B} \circ \mathcal{D}(P). \tag{3}$$

The fixed point P can be computed by iterating to convergence on the composite operator $\mathcal{B} \circ \mathcal{D}$ and the robust decision rule is $a = -Fy$, where

$$F = \beta(R + \beta B'\mathcal{D}(P)B)^{-1}B'\mathcal{D}(P)A. \tag{4}$$

The worst-case shock obeys the decision rule $w = Ky$, where

$$K = \theta^{-1}(I - \theta^{-1}C'PC)^{-1}C'P(A - BF). \tag{5}$$

For the stochastic version of the model, the worst case distribution of y_{t+1} conditional on y_t is normal with mean $(A_F + CK)y_t$ and covariance matrix

$$\hat{C}\hat{C}' = C(I - \theta^{-1}C'PC)^{-1}C'. \tag{6}$$

We use these objects when we compute continuation entropy below.

5 Worst case mean distortion for fixed F

For a fixed F, define

$$Q_F = Q + F'RF$$

and

$$A_F = A - BF.$$

For a fixed F, compute a worst-case shock mean by finding a fixed point of

$$P = Q_F + \beta A_F' \mathcal{D}(P) A_F \tag{7}$$

and then apply formula (5) for K. (Using techniques in Hansen-Sargent (2008), it is easy to compute these objects using an ordinary optimal linear regulator. I'll add details.) The optimal value under the worst-case model (or, when $\theta < 0$, the best case model) associated with a fixed F is then

$$-y_0' P y_0 - p \tag{8}$$

where

$$p = \frac{\beta}{1-\beta} \text{trace} P\hat{C}\hat{C}'$$

where $\hat{C}\hat{C}'$ satisfies (6). Here, F is understood to be a function of the θ parameter in the robust control or optimism control problem.

5.1 Computing continuation entropy

For the stochastic version of the discounted linear quadratic problem, it is useful to compute continuation entropy, which satisfies the difference equation

$$R_t = \beta E_t m_{t+1}(\log m_{t+1} + R_{t+1}). \tag{9}$$

For the present problem, the increment to entropy $E_t m_{t+1} \log m_{t+1}$ equals

$$y_t' K' K y_t + h_o, \tag{10}$$

where

$$h_o = \frac{1}{2}\text{trace}[(I - \theta^{-1} C' P C)^{-1} - I] - \frac{1}{2} \log \det (I - \theta^{-1} C' P C)^{-1}$$

and where P and K are to be computed using formulas(7) and (5). To compute continuation entropy R_t that solves (9), guess that

$$R_t = y_t' O y_t + o$$

and represent (9) as[1]

$$\begin{aligned} & \beta y_t' K' K y_t + h_o + \beta E((A_F + CK) y_t + C\epsilon_{t+1})' O ((A_F + CK) y_t + C\epsilon_{t+1}) + \beta o \\ = \; & x_t' O x_t + o. \end{aligned} \tag{11}$$

Equation (11) implies that the matrix O satisfies the discrete Lyapunov (or Sylvester) equation

$$O = \beta K' K + \beta (A_F + CK)' O (A_F + CK), \tag{12}$$

and that the constant o then satisfies

$$o = h_o + \beta \text{trace}(O \hat{C} \hat{C}') + \beta o, \tag{13}$$

where $\hat{C}\hat{C}'$ satisfies (6). Equation (12) can be solved using the Matlab program `dlyap.m` or `doublej2.m`.

6 The killer graph

Here is how to construct *the graph*. To get started, I'll give you a particular economic model that pins down the matrices (A, B, C, Q, R) and the discount factor β and some initial state y_0. We'll figure out a way to choose an interesting initial state y_0 by first computing the stationary distribution under the approximating model and a non-robust rule or something like that. We'll experiment with some different θ's.

[1]Note that we are evaluating expectations under the worst-case model as required by formula (9).

1. Solve an ordinary (non-robust) problem (i.e., set $\theta = +\infty$). Compute the associated F, P. Call it F_o to denote the nonrobust rule.

2. Solve a robust control problem with a particular $\theta_r \in (\underline{\theta}, +\infty)$. (Here $\underline{\theta}$ is the breakdown point for this particular problem.) Compute the associated F_r and value function matrix P_r using the standard formulas (3) and (4) for the robust control problem.

3. Pick a grid of θ's. Fix $F = F_o$. For each θ in the grid, use formulas (7) and (5) to compute an associated K and P. Do the same thing for each θ in *minus* the *same* grid. Here, for each θ we are evidently computing worst and best case shocks and associated values. Call these objects $P_o(\theta), K_o(\theta)$ for the respective values of θ.

4. Repeat the procedure in the preceding bullet point, but this time for $F = F_r$ computed in the second bullet point.

5. For $F = F_o$ and the same grid of θs, both positive and negative ones, compute discounted entropy using formulas (12) and (13).

6. For $F = F_r$, and the same grid of θs, both positive and negative ones, compute discounted entropy using formulas (12) and (13).

7. Now for both F_o and F_r plot the associated best and worst values on the vertical axis against discounted entropy on the horizontal axis. Please do this by sweeping out values $-\frac{1}{2}y_0 P y_0$ and discounted entropy by varying θ within the grid. Both values and discounted entropies are functions (described above) of θ.

7 Matlab programs

Here are some programs described in *Robustness* (2008) that will do much of the work.

- Perhaps the simplest program is `olrprobust.m` described on page 43. It tricks a robust control problem into an ordinary optimal linear regulator problem.

- `doublex9.m` uses a doubling algorithm to solve a robust control problem. See pages 368-369 of *Robustness*, but this accomplishes more than we need here.

- I advocate relying on `olrprobust.m` to do most of the work, then using `dlyap.m` to solve Lyapunov equations where needed.

8 For Chase and Spencer

I recommend that you start reading here.

In the *.zip file are several Matlab files that you could either use to make the graph (or graphs), or better yet, you could pythonize and improve the programs. Here is a description of the Matlab files:

- `olrp.m` solves an optimal linear regulator without robustness.

- `doublex9.m` computes a robust decision rule for an LQ problem. It uses a fancy doubling algorithm to accelerate the computations.

- `olrprobust.m` uses a trick to induce `olrp.m` to solve a robust control problem. I don't use this program here, but I could have used it instead of `doublex9.m`.

- `doublej.m` solves a discrete Lyapanov equation in a way that I trust.

- `Kworst.m` computes key objects that we need for the graph. Takes an arbitrary decision rule F and then for a given parameter sig computes a worst-case (if sig is negative) or best case (if sig is positive) shock process. It also computes a bunch of other stuff I want you to graph.

- `monopolist_robust_mock2.m` sets up the example for which I want you to compute the graph. It is an LQ model of a monopolist facing adjustment costs and a linear inverse demand curve. `monopolist_robust_mock2.m` shows how to compute everything that I want you to graph. Especially look at the last lines of `monopolist_robust_mock2.m`.

Here is what I'd like you to do to create the 'killer graph.'

- For the given parameter values, use `olrp.m` to compute an optimal rule without robustness.

- For a given $\sigma = -\frac{1}{\theta}$, compute a robust rule.
- Save both of the preceding rules. We'll graph how both behave under different probability models that we'll obtain by using `Kworst.m`.
- For two lists of values of the σ_c in intervals $[-\underline{\sigma}_c, .0000001]$ and $[.00000001, \overline{\sigma}_c]$, I want you to sweep out two sets delineated by upper and lower bounds, one set for each of the two decision rules constructed above. I'll describe these sets in the next two bullet points. The lower bound is affiliated with negative σ_c's, the upper bound with positive σ_c's.
- One set is associated with the optimal rule without robustness. To get the lower boundary of this set, take the negative interval of σ_c's and for a given value of the state vector (see what I did in the final lines of `monopolist_robust_mock2.m` please) sweep out the values of the value function and entropy. Plot the value function on the vertical axis against entropy on the negative axis. Then do the same thing for the positive intervals of σ_c.
- Do the same calculations except now for the other decision rule, the robust one calculated earlier.
- Plot both sets on the same graph. The optimal set should start out higher for zero entropy but be fatter as entropy grows.