

Optimal control in a simple system

1. A simple model system for optimal control.

We'll be looking at how to deduce optimal control strategies for a very simple model system (see Fig. 1). Obtain a copy of Chapter 1 of Kappen's book ('Optimal control theory and the linear Bellman equation'). Ensure that you are comfortable with his Eqns. 1.1-1.4. We will be implementing the algorithm on page 4.

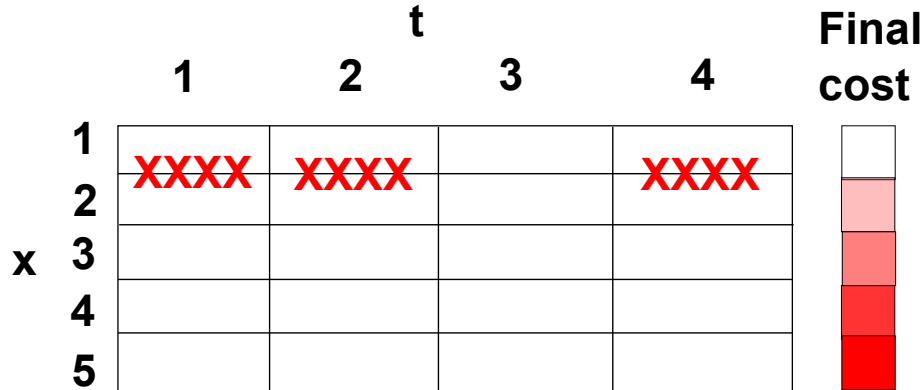


Figure 1: A simple model system for control. We have 4 discrete time points and 5 discrete space points. Control strategies at each point can shift the current space state up or down, or leave it unperturbed. Costs are low but nonzero for shifts, except when such a shift would cross the 'X' barriers, which incur a high cost. Final cost is an increasing function of distance from $x = 1$.

2. Initialisation. Implement MATLAB code to do the following:

- (a) Initialise a matrix J of size $(n_t + 1) \times n_x$ with all zeroes, except for the elements $J_{(n_t+1),x}$, which should take values $2(x - 1) + (x - 1)^2$.
- (b) Initialise two matrices c^+ , c^- , of size $n_t \times n_x$, which store the costs of making a control move at state (t, x) that increases or decreases x by 1 respectively. Use cost 100 for crossing the 'X' barriers, 1000 for leaving the grid (this ensures we never do so), and 1 otherwise.
- (c) Initialise a matrix u' of size $n_t \times n_x$ with all zeroes.

3. Dynamic programming – backwards cost-to-go. You have done step 1 in Kappen's algorithm on page 4. Now do step 2. Implement code to loop t backwards from T to 1 (we're indexing from 1, rather than 0, because of MATLAB's idiosyncrasy). For each t , loop over all x values, and compute the costs of employing a positive, negative, or zero control move. In our model, these are respectively:

$$R_{t,x}^+ = c_{t,x}^+ + J_{t+1,x+1} \tag{1}$$

$$R_{t,x}^- = c_{t,x}^- + J_{t+1,x-1} \tag{2}$$

$$R_{t,x}^0 = 0 + J_{t+1,x}. \tag{3}$$

Find the lowest-cost strategy, and store the relevant strategy in your u' matrix (this should store 1, -1, or 0 depending on the winner) and the resulting cost-to-go in J . Notice how looping back in time allows us to use the elements of J that we have previously computed.

4. Forward trajectories. You now have a matrix u' of optimal control strategies at each point in spacetime. Now do Kappen's step 3. Loop over all initial x values. For each initial condition, loop over t from 1 to T , and at each timestep, update the current state x with the appropriate control strategy from u' .

Controlling a simple stochastic system

1. Sampling stochastic control strategies.

In this practical, we'll be looking at how we can construct optimal control strategies by sampling random trajectories of a stochastic dynamic system. Obtain a copy of *Path integrals and symmetry breaking for optimal control theory*, H. J. Kappen, J. Stat. Mech. **11** p11011 (2005) (also available on the arXiv). Read Section 5.1. In particular, take note of Eqn. 30 and the text beneath.

2. Deriving a controlling strategy for a stochastic path through a single slit.

- (a) The file `SingleSlit.m` simulates stochastic paths through a slit. Observe lines 8 through 23. Write down each variable and a sentence describing its meaning. What is w ?
- (b) The matrix `randomstore` stores the value of each random step taken by the system, referenced by a trajectory label and the time at which the step was taken. We will use these steps to derive a control strategy u (see Fig. 2). Observe lines 57 to 73. Write down a description of the function of each line.
- (c) Run the code. Use `imagesc` to plot the resulting `statem`, which shows the number of runs making it to each point in spacetime. An intuitive plot can be obtained using `colormap('gray'); imagesc(flipud(1-statem'/10000))`;

Plot the resulting u . How can this be interpreted as a control strategy?

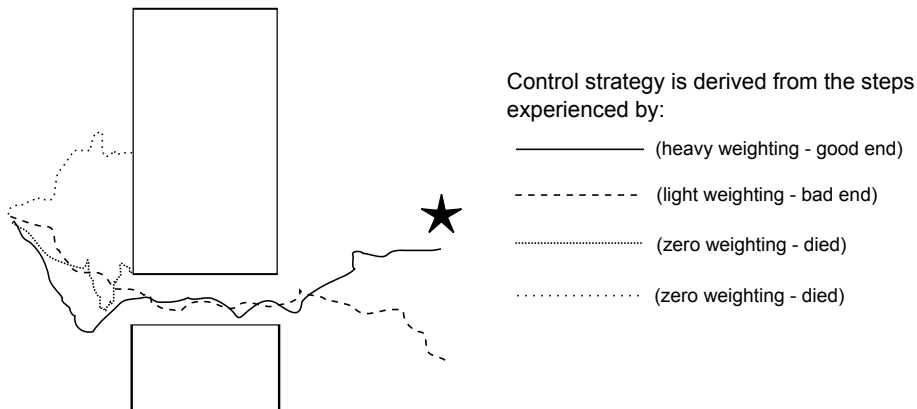


Figure 2: Deriving a control strategy by sampling trajectories.

3. Controlling a stochastic system.

- (a) The file `SingleSlitControl.m` allows you to provide a control strategy to the system. Compare the number of surviving runs using a zero strategy, and using the control strategy derived in the previous question. Use `imagesc` to plot the `statem` density recorded using control. Compare the uncontrolled case.
- (b) The single-slit model illustrates stochastic control but is a bit abstract. How may it map to problems in biology? Consider what the variable under control would be (x); whether time or another variable is the ordinate; and how a control strategy (u) may be manifest. How may a control strategy be 'derived' in biology?