Exercises for Lecture Course on Optimal Control and Estimation (OCE)
Albert-Ludwigs-Universität Freiburg – Summer Term 2015

Exercise 3 - Dynamic Programming (deadline: 18.5.2014, 2:15pm)

Prof. Dr. Moritz Diehl, Jonas Koenemann and Greg Horn

1 Discrete State Dynamic Programming on Paper

Consider a very simple system with state $x \in \{1, 2, \dots, 10\}$ and controls $u \in \{-1, 0, 1\}$ and time invariant dynamics $f(x, u) = x + u$ and stage cost $L(x, u) = |u|$ on a horizon of length $N = 3$. The terminal cost $E(x)$ is given by zero if $x = 5$ and by 100 otherwise. Take pen and paper and compute and sketch the cost to go functions J_3, J_2, J_1, J_0.

(3 points)

2 Linear Quadratic Dynamic Programming on Paper

Use dynamic programming to solve the following simple discrete time OCP with one state and one control by hand. On the way towards the solution, explicitly state the cost to go functions $J_2(x), J_1(x), J_0(x)$ and feedback control laws $u_0^*(x)$ and $u_1^*(x)$.

$$\begin{aligned}
\underset{x_0, x_1, x_2, u_0, u_1}{\text{minimize}} \quad & \sum_{k=0}^{1} u_k^2 \; + \; 10 x_2^2 \\
\text{subject to} \quad & \\
x_0 &= 5 \\
x_{k+1} &= x_k + u_k, \quad k = 0, 1.
\end{aligned}$$

(3 points)

3 Discrete State Dynamic Programming for an Inverted Pendulum

In this task we are using Dynamic Programming to find optimal controls to swing up a pendulum. The state of the system is $x = [\phi, \omega]^\top$ where ϕ is the angle and ω the angular velocity of the pendulum. When $\phi = 0$, the pendulum is in its inverse position, e.g. the mass is at the top. The system dynamics are given by

$$\begin{aligned}
\dot{\phi} &= \omega \\
\dot{\omega} &= 2\sin(\phi) + u
\end{aligned}$$

where $\omega \in [\omega_{\min}, \omega_{\max}]$, and $u \in [u_{\min}, u_{\max}]$.
The find the controls for the pendulum swing-up, we are solving the following optimal control problem:

$$\begin{aligned}
\underset{x_0, \dots, x_N, u_0, \dots, u_{N-1}}{\text{minimize}} \quad & \sum_{k=0}^{N-1} (\phi_k^2 + u_k^2) \\
\text{subject to} \quad & \\
f(x_k, u_k) - x_{k+1} &= 0, \quad \text{for} \quad k = 0, \dots, N-1 \\
\bar{x}_0 &= x_0 \\
u_{\min} &\leq u_k \leq u_{\max}, \quad \text{for} \quad k = 0, \dots, N-1 \\
\omega_{\min} &\leq \omega_k \leq \omega_{\max}, \quad \text{for} \quad k = 0, \dots, N
\end{aligned}$$

Dynamic Programming requires a system which is discrete in space and in time. We already prepared the discretization of the continuous system for you in the file `pendulum_template.m` on the course webpage. The discretization is done in the following way:

The discrete versions of ϕ, ω and u live in the integer space $\mathbb{Z}$ and thus are denoted by ϕ_Z, ω_Z and u_Z. The conversion from real space to integer space is done by projection of the variables into N_ϕ, N_ω, N_u equally spaced bins in the range of the variables. In the template file you find predefined functions to convert the variables between integer and real numbers, e.g. `phiZ_to_phi` and `phi_to_phiZ`.

To complete the tasks, fill in the missing parts of the template file `pendulum_template.m`.

Tasks

1. Use the function `integrate_Z` to simulate the system in discrete space with $x_0 = [0.4, 0]^\top$ and $u = 0$. Do $N = 60$ integration steps and use a timestep of $h = 0.12$. Plot the evolution of ϕ and ω in time (in continuous state space). Make a plot (animation) that shows the motion of the pendulum. Assume a rod length of $1m$. You don't need to submit the animation in the pdf, it's just for you to see if the system behaves well.

2. In the following section of the template file you see the precalculation of all integrations in the discrete state space to avoid unnecessary computations during Dynamic Programming. For a given combination of ϕ_Z, ω_Z, and u_Z, the resulting state from integration $[\phi_Z^+, \omega_Z^+]$ is stored in the lookup tables `PhiNext` and `WNext` and the costs are stored in the table `L`.

   Use the lookup tables to do the same simulation as in Task 1. Plot the evolution of ϕ and ω in time (in continuous state space).

3. Implement the backward pass (recursion) of Dynamic Programming, i.e. calculate the cost-to-go function $J_k(x_k)$ going from $k = N$ to $k = 1$. For $k = N$, the cost-to-go is initialized zero for all states (no terminal cost). Fill in the missing lines in the template file for this task. Use $N = 60$ and $h = 0.12$.

4. Simulate the system using the optimal controls which are given by

$$u_k^*(x_k) = \arg\min_u \phi_k^2 + u^2 + J_{k+1}(f(x_k, u))$$

   starting from $x_0 = [-\pi, 0]^\top$. Plot the evolution of ϕ and ω in time (in continuous state space). Make a plot (animation) that shows the motion of the pendulum to see if the pendulum swings up.

(4 points)