Numerical Optimal Control, August 2014

Exercise 5: Dynamic programming

Joel Andersson Joris Gillis Greg Horn Rien Quirynen Moritz Diehl

University of Freiburg – IMTEK, August 5th, 2014

Dynamic programming for a two-state OCP

Dynamic programming and its continuous time counterpart – the Hamilton-Jacobi-Bellman equation – can be used to calculate the global solution of an optimal control problem. Unfortunately they suffer from Bellman's so-called "curse-of-dimensionality", meaning that they get exponentionally expensive with the number of states and control. In practice, they can be used for systems with 3-4 differential states or systems that have special properties.

Here we shall consider a simple OCP with two states (x_1, x_2) and one control (u):

$$\begin{aligned} \underset{x,u}{\text{minimize}} \quad & \int_0^T x_1(t)^2 + x_2(t)^2 + u(t)^2 \, dt \\ \text{subject to} \quad & \dot{x}_1 = (1 - x_2^2)\, x_1 - x_2 + u, \qquad x_1(0) = 0 \\ & \dot{x}_2 = x_1, \qquad x_2(0) = 1 \\ & -1 \le x_1(t) \le 1, \quad -1 \le x_2(t) \le 1, \quad -1 \le u(t) \le 1, \end{aligned} \tag{1}$$

with $T = 10$.

To be able to solve the problem using dynamic programming, we parameterize the control trajectory into $N = 20$ piecewise constant intervals. On each interval, we then take N_K steps of a RK4 integrator in order to get a discrete-time OCP of the form:

$$\begin{aligned} \underset{x,u}{\text{minimize}} \quad & \sum_{k=0}^{N-1} F_0(x_1^{(k)}, x_2^{(k)}, u^{(k)}) \\ \text{subject to} \quad & x_1^{(k+1)} = F_1(x_1^{(k)}, x_2^{(k)}, u^{(k)}), \quad k = 0, \dots, N-1, \quad x_1^{(0)} = 0 \\ & x_2^{(k+1)} = F_2(x_1^{(k)}, x_2^{(k)}, u^{(k)}), \quad k = 0, \dots, N-1, \quad x_2^{(0)} = 1 \\ & -1 \le x_1^{(k)} \le 1, \quad -1 \le x_2^{(k)} \le 1, \quad -1 \le u^{(k)} \le 1 \qquad \forall k. \end{aligned} \tag{2}$$

Tasks:

5.1 On the course webpage and on Gist[1], you will find an incomplete implementation of dynamic programming for problem (2). Add the missing calculation of the cost-to-go function to get the script working.

5.2 Add the additional end-point constraint $x_1(T) = -0.5$ and $x_2(T) = -0.5$. How does the solution change?

[1] `https://gist.github.com/jaeandersson/e37e796e094b3c6cad9e`