Homework assignment 1 (Draft version)

Optimal control and dynamic programming (4SC000), TU/e, 2020-2021

Lecturer: Duarte Antunes

Important note: For the visualization of the environment in Assignments 1,2,3 you can use the live script BuzwireChallenge.mlx available on canvas.
Submission: The assignments should be submitted via Matlab grader and via canvas (.m files). Due to computation limitations of Matlab Grader, Assignments 1, 2, 3 can eventually be hard-coded, i.e., the output can be determined by matrices that should be simply placed in the functions (you can use the built-in function writematrix.m if your Matlab version is not older than 2019a). If you decide to do this, the submitted code to canvas should generate the same matrix as the hard-coded versions submitted to Matlab grader. However, with efficient implementations this should not be necessary.
Grade: The overall assignment will be graded on a scale 0-10 according to the following distribution

Assignment	1	2	3	4	Total
Max score	2.5	2.5	2.5	2.5	10

There are 6 (hidden and visible) tests for each of the assignments and the grade will be given by

$$\text{grade assignment} = y(x) \times \text{Max score}$$

where $x \in \{0, 1, \ldots, 6\}$ denotes the number of tests (either hidden or visible) passed and $y(x)$ is specified in the next table

x	0	1	2	3	4	5	6
$y(x)$	0	0.2	0.2	0.4	0.5	0.6	1

Plagiarism: The Matlab code will be checked with a plagiarism tool. In case of fraud, no points will be given for the entire assignment.
Deadline: December 3rd, 23h45, 2020.

Buzz wire challenge

In the buzz wire game a metal loop is guided along a wire. The goal is to reach the end terminal of the wire without the loop touching the wire, in which case a buzz noise is heard (see, e.g., this video `https://www.youtube.com/watch?v=5NuuDHMMwVQ`).

Suppose that the game is played with a robot arm holding the metal loop. This assignment aims at finding a strategy for the robot arm to move as fast as possible from the start to the end terminal of the wire without touching the wire. For simplicity, the following assumptions are considered:

- The wire lies in a plane with x and y coordinates. Moreover, it can be represented by a function $y = f(x)$ as depicted in Figure 1, which implies that it does not curve back.
- The metal loop is represented by two small parallel bars, shown in blue in Figure 1, which form a rigid body with center point represented by a blue circle in Figure 1. Both the length of each bar and the space between the two bars are set to 4.1. The configuration of these parallel linear is characterized by the position of the center point, denoted by (p_x, p_y), and an orientation angle $\theta \in (-\frac{\pi}{2}, \frac{\pi}{2})$, which represents a rotation angle with respect to a frame with origin at the center point; $\theta = 0$ when the two blue bars are parallel to the x axis and $\theta > 0$ corresponds to an anti-clockwise rotation. The metal loop touches the wire if one of these parallel bars intersects the wire. This check is performed for a finite set of points of the bottom and top bars.
- The space is discretized such that each coordinate position of the center point (x, y) is a natural number, $x \in \{1, 2, \dots, N_x\}$, $y \in \{1, 2, \dots, N_y\}$ and the angle belongs to the set $\theta \in \{-\frac{\pi}{2} + k\frac{\pi}{N_\theta+1}\}$, for $k \in \{1, 2, \dots, N_\theta\}$.
- The game starts at time $k = 1$ with $p_{x,1} = 1$ and a feasible configuration where the metal loop does not touch the wire.
- The game is over when $p_{x,k} = N_x$ for a given $k = K$ and there has been no collisions for every $k \leq K$. K is the time to be minimized.

Deterministic model

A purely kinematic model for the robot holding the metal loop is considered and it is assumed that the joint angles are such that the robot can reach all the configurations of interest, therefore (using the robot kinematics) we can assume that x, y, and θ can be controlled directly. The following discrete-time model is considered

$$\begin{aligned} p_{x,k+1} &= \min\{p_{x,k} + u_{x,k}, N_x\} \\ p_{y,k+1} &= \min\{\max\{p_{y,k} + u_{y,k}, 1\}, N_y\} \\ \theta_{k+1} &= \min\{\max\{\theta_k + u_{\theta,k}, -\pi\frac{N_\theta}{N_\theta+1}\}, \pi\frac{N_\theta}{N_\theta+1}\} \end{aligned}$$

with

$$\begin{aligned} &u_{x,k} \in \{0, 1, \dots, M_x - 1\}, \\ &u_{y,k} \in \{-(M_y - 1)/2, -(M_y - 1)/2 + 1, \dots, (M_y - 1)/2\}, \\ &u_{\theta,k} \in \{-(M_\theta - 1)/2, -(M_\theta - 1)/2 + 1, \dots, (M_\theta - 1)/2\} \end{aligned}$$

for odd M_y and M_θ.

Assignment 1 (Dynamic programming) Program a matlab function

```
[mu,J] = buzzwiredp(Nx,Ny,Ntheta,Mx,My,Mtheta,wirefunction);
```

which provides the optimal policy and the optimal cost to go for each possible configuration of the metal loop x, y, θ to reach the end of the wire with the least number of time steps using the dynamic programming algorithm. The input arguments are N_x, N_y, N_θ, M_x, M_y, M_θ and wirefunction which is a row vector with dimensions $\frac{N_x-1}{\delta}+1$ where $\delta = 0.1$, with `wirefunction(i)` $= f((i-1)\delta + 1)$. The output arguments are:

- a cell array mu{px}{py}{itheta}, with px$\in \{1,\dots,Nx\}$, py$\in \{1,\dots,Ny\}$, and itheta$\in \{1,\dots,N_\theta\}$, such that $\theta = itheta\frac{\pi}{N_\theta+1}$ where each entry is a 3 dimensional vector mu{px}{py}{itheta}$= [u_x, u_y, u_\theta]$ corresponding to an optimal decision for this state;

- a cost-to-go multidimensional vector J(px,py,itheta) where px,py,itheta takes values in the sets mentioned above and whose value is the optimal time to complete the game if the game starts at (px,py,itheta) in case this is a feasible state (the wire lies inside the area encompassed by the two blue bars and there is no collision) and J(px,py,itheta)$=\infty$ otherwise (mu{px}{py}{itheta} can be any value in this latter case).

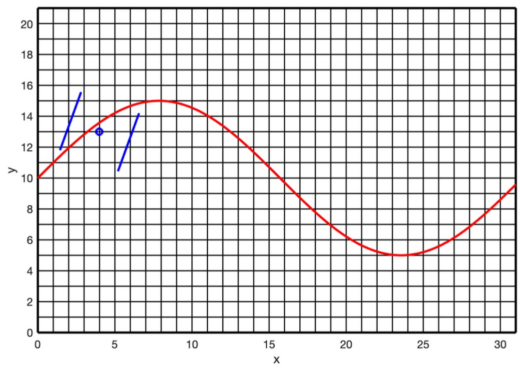

Figure 1: Buzzwire environment. The wire, depicted in red, lies on a plane and it is represented by a function $y(x)$. The two blue parallel bars represent the metal loop and form a rigid body with center at the center point represented by a blue circle. This rigid body rotates by an angle θ about a frame centered at the center point; $\theta = 0$ when both blue bars are parallel to the x axis, and $\theta > 0$ represents an anticlockwise rotation.

Assignment 2 (Dijktra) Program a matlab function

```
[optpath] =
buzzwiredijkstra(Nx,Ny,Ntheta,Mx,My,Mtheta,wirefunction,x0);
```

which, given an initial configuration, provides the optimal path to reach the end of the wire with the least number of time steps using the Dijkstra's algorithim. The first seven input arguments are as in Assignment 1 and x0 = [px py itheta] is the initial configuration (with px = 1 and which can be assumed to be feasible) with py$\in \{1, \dots, Ny\}$, and itheta$\in \{1, \dots, N_\theta\}$. The output argument is a vector with $T+1$ rows and 3 collums specifying one optimal path which needs T time steps to reach the end; each row $k \in \{1, \dots, T+1\}$ contains the configuration $\begin{bmatrix} p_{x,k-1} & p_{y,k-1} & \text{itheta}_{k-1} \end{bmatrix}$ after k time steps,$p_{x,k-1} \in \{1, \dots, Nx\}$, $p_{y,k-1} \in \{1, \dots, Ny\}$, and $\text{itheta}_{k-1} \in \{1, \dots, N_\theta\}$. Note that $\begin{bmatrix} p_{x,0} & p_{y,0} & \text{itheta}_0 \end{bmatrix}$ is the initial configuration ($p_{x,0} = 1$) and $\begin{bmatrix} p_{x,T} & p_{y,T} & \text{itheta}_T \end{bmatrix}$ is the final configuration ($p_{x,T} = N_x$).

Stochastic model

For assignment 3, a stochastic model is assumed instead of the deterministic model considered above. This means that even if an action tries to enforce that the metal loop arrives at a given configuration, this might not be the case due to disturbances. Disturbances are assumed to only change the position and not the angle. The exact discrete time model considered is the following

$$\begin{aligned} p_{x,k+1} &= \min\{\max\{p_{x,k} + u_{x,k} + d_{x,k}, 1\}, N_x\} \\ p_{y,k+1} &= \min\{\max\{p_{y,k} + u_{y,k} + d_{y,k}, 1\}, N_y\} \\ \theta_{k+1} &= \min\{\max\{\theta_k + u_{\theta,k}, -\pi\frac{N_\theta}{N_\theta+1}\}, \pi\frac{N_\theta}{N_\theta+1}\} \end{aligned}$$

where $d_{x,k} \in \{-1, 0, 1\}$ and $d_{y,k} \in \{-1, 0, 1\}$. The probabilities of all the possible events are summarized in a vector $p = \begin{bmatrix} p_1 & p_2 & p_3 & \dots p_9 \end{bmatrix}$

$$\text{Prob}[(d_{x,k}, d_{y,k}) = c_i] = p_i, i \in \{1, \dots, 9\},$$

with $c_1 = (-1, 1)$, $c_2 = (0, 1)$, $c_3 = (1, 1)$, $c_4 = (-1, 0)$, $c_5 = (0, 0)$, $c_6 = (1, 0)$, $c_7 = (-1, -1)$, $c_8 = (0, -1)$,$c_8 = (1, -1)$. Since the model is stochastic it might be the case that collisions do occur, even if unintended (i.e., these would not occur if $(d_{x,k}, d_{y,k}) = (0, 0)$). If a collision occurs the game stops and a typically large positive penalty C_W is incurred. Note that if the metal wire is far from the end goal it might be optimal to hit the wire (or risk to hit it) and incur a penalty CW than incurring in a penalty coinciding with the time to reach the end.

Assignment 3 (stochastic) Program a matlab function

```
[mu,J] =
buzzwiredpstoch(Nx,Ny,Ntheta,Mx,My,Mtheta,wirefunction,CW,p);
```

which provides the optimal policy and the optimal cost to go for each possible configuration of the metal loop x, y, θ. The first seven input arguments are the same as for the function buzzwiredpstoch; CW is the cost of hitting the wall and p is the vector containting the probability distribution. The output arguments are the same, but note that now the cost-to-go does not necessarily coincide with the optimal time to reach the wire end terminal, since the game might stop with a penaly C_W before the end terminal is reached.

IMU estimation with Bayes filter

Background

An Inertial measurement unit (IMU) comprises accelerometers, gyroscopes and magnetometers and allows to estimate the pose of a rigid body. The gyroscope provides measurements of the angular velocity and from the measurements of the accelerometer and of the magnetometer one can extract measurements of the three angles (Euler angles according to a given convention) characterizing the attitude. In a simple linear model the kinematic equations for the three angles are decoupled. In particular, one can simply consider the following model

$$\dot{\lambda} = \omega$$

where $\lambda \in \mathbb{R}$ denotes any of the three angles describing the attitude (pitch, roll, or yaw), and $\omega \in \mathbb{R}$, m, I_ω denote the angular velocity, the torque and the moment of inertial about the corresponding axis. Using a simple Euler discretization, we obtain

$$\lambda_{k+1} = \lambda_k + \tau\omega_k$$

where for a given function of time $x(t)$ we use the notation $x_k := x(k\tau)$ and τ is the sampling time.

The following measurements are available

$$y_{\lambda,k} = \lambda_k + n_k$$
$$y_{\omega,k} = \omega_k + v_k$$

where $\{n_k | k \in \mathbb{N}_0\}$ and $\{v_k | k \in \mathbb{N}_0\}$ are independent and identically distributed sequences of zero mean Gaussian white noise with variance at time k, N and V, respectively.

The goal is to find an estimator providing estimates $\hat{\lambda}_k$ of the angle λ_k, for $k \in \mathbb{N}_0$, such that the following asymptotic variance is minimized

$$\lim_{k\to\infty} \mathbb{E}[(\lambda_k - \hat{\lambda}_k)^2]. \tag{1}$$

One of the most known methods is the complementary filter. The complementary filter can be written, in continuous-time, as

$$\dot{\hat{\lambda}} = y_\omega + \alpha(y_\lambda - \hat{\lambda}),$$

where α is a given constant; in discrete-time it takes the form

$$\hat{\lambda}_{k+1} = \hat{\lambda}_k + \tau y_{\omega,k} + \tau\alpha(y_{\lambda,k} - \hat{\lambda}_k).$$

It is possible to show (see Live Script KalmanIMU.mlx) that when $\alpha\tau$ take a special form (coincides with the Kalman gain) the complementary filter is simply an implementation of the optimal Kalman filter providing estimates $\hat{\lambda}_k = \hat{\lambda}_{k|k-1} = \mathbb{E}[\lambda_k | \{(y_{\lambda,0}, y_{\omega,0}), \dots, (y_{\lambda,k-1}, y_{\omega,k-1}), \}]$. It is optimal in the sense that the asymptotic variance

$$\lim_{k\to\infty} \mathbb{E}[\lambda_k | \{(y_{\lambda,0}, y_{\omega,0}), \dots, (y_{\lambda,k-1}, y_{\omega,k-1}), \}]$$

is the smallest among all possible filters that rely on past measurements.

The first goal of this exercise is to use the Bayes' filter and a discretization technique to show that the Bayes' filter can also provide approximate optimal estimates with a very close variance to the one of the Kalman filter. In fact, as we shall see later in the course, the Kalman filter is a special case of the Bayes' filter in continuous spaces when the disturbances and noise are Gaussian (the approximation comes from the fact that here we use the Bayes filter in discrete spaces). The second goal is to convey the message that the Bayes' filter is more general and can also be used when the disturbances and noise are not Gaussian distributed.

Assignment

We can abstract from the application and simply consider the following model

$$x_{k+1} = x_k + u_k + w_k$$

with measurements

$$y_k = x_k + v_k$$

by making $x_k = \lambda_k$, $u_k = \tau y_{\omega,k}$, $w_k = \tau y_{\omega,k}$, $y_k = y_{\lambda,k}$. Since we will consider the Bayes filter in discrete state, control and output spaces we need to discretize the variables x_k, y_k, u_k, w_k and v_k. To this effect, for a given discretization step, we consider that these variables belong to the following sets for odd natural numbers n_x, n_u, n_w, n_y, n_v

$$\xi_k \in \{\tau\bar{\xi}_k\}, \quad \bar{\xi}_k = \underline{\xi}_k - \frac{(n_\xi + 1)}{2} \quad \underline{\xi}_k \in \{1, 2, \dots, n\},$$

where ξ should be replaced by x, y, u, w and v. Then we consider the following model

$$\bar{x}_{k+1} = \min\{\max\{\bar{x}_k + \bar{u}_k + \bar{w}_k, -\frac{(n_\xi - 1)}{2}\}, \frac{(n_\xi - 1)}{2}\}$$

and

$$\bar{y}_k = \min\{\max\{\bar{x}_k + \bar{v}_k, -\frac{(n_y - 1)}{2}\}, \frac{(n_y - 1)}{2}\}$$

where the min and max operations ensure that the state and output lie on the specified sets. This is just an artefact in order to make sure that these variables live in finite sets and it does not play a role provided that the probability distribution of the state is sufficiently small close to the boundaries of these sets. Suppose that the following probability distributions are given

$$P_{w,i} = \text{Prob}[\bar{w}_k = i - \frac{(n_w + 1)}{2}], \quad i \in \{1, 2, \dots, n_w\}$$
$$P_{v,i} = \text{Prob}[\bar{v}_k = i - \frac{(n_v + 1)}{2}], \quad i \in \{1, 2, \dots, n_v\}$$
$$P_{x_0,i} = \text{Prob}[\bar{x}_0 = i - \frac{(n_x + 1)}{2}, \quad i \in \{1, 2, \dots, n_x\}],$$

Then the Bayes' filter allows to compute

$$P_{x_k|k,i} = \text{Prob}[\bar{x}_k = i - \frac{(n_x + 1)}{2}|I_k], \quad P_{x_k|k-1,i} = \text{Prob}[\bar{x}_k = i - \frac{(n_x + 1)}{2}|I_{k-1}]$$

where $i \in \{1, 2, \dots, n_x\}$ and $I_k = I_{k_1} \cup \{y_k, u_{k-1}\}$, $I_0 = \{y_0\}$ from which we can compute the expected value and the variance of x_k conditioned on I_k and I_{k-1}. If $P_{w,i}$, $P_{w,i}$ and $P_{x_0,i}$ corresponds to discretization of Gaussian distributions, i.e., ,

$$P_{\xi,i} = \frac{1}{\sqrt{2\pi}\sigma_\xi} e^{-(\frac{i - \frac{(n_w+1)}{2}}{2\sigma_\xi})^2}, \quad P_{x_0,i} = \frac{1}{\sqrt{2\pi}\sigma_{x_0}} e^{-(\frac{i - \frac{(n_w+1)}{2} + \mu_0}{2\sigma_{x_0}})^2},$$

for ξ replaced by w, v, for a given integer μ_0 and positive real numbers σ_ξ and σ_{x_0} then we can approximate the value of the optimal asymptotic covariance obtained in the live script Kalman-IMU.mlx. However the function provided in this assignment should consider arbitrary probability distributions. Since u_k plays no role in computing the asymptotic variance for simplicity assume $u_k = 0$ for every time step.

Assignment 4

Program a Matlab function

```
[p,p1,varp,varp1] = IMUBayes(n,tau,pdistw,pdistv,pdistx0,y)
```

which provides as outputs matrix $[p_{i,k+1}]_{1\leq i\leq n, 1\leq k\leq K}$, $[p^1_{i,k+1}]_{1\leq i\leq n, 1\leq k\leq K}$ with $p_{i,k+1} = [P_{x_k|k,1} \quad P_{x_k|k,2} \quad \dots \quad P_{x_k|k,n_x}]$, $p^1_{i,k+1} = [P_{x_k|k-1,1} \quad P_{x_k|k-1,2} \quad \dots \quad P_{x_k|k-1,n_x}]$ and varp, varp1 are the estimated variances of x given I_k and I_{k-1} respectively. The input arguments are n, tau=τ, row vectors pdistw= $\begin{bmatrix}P_{w,1} & P_{w,2} & \dots P_{w,n_w}\end{bmatrix}$, pdistv= $\begin{bmatrix}P_{v,1} & P_{v,2} & \dots P_{v,n_v}\end{bmatrix}$, pdistx0= $\begin{bmatrix}P_{x_0,1} & P_{x_0,2} & \dots P_{x_0,n_x}\end{bmatrix}$ and a vector of measurements y=$\begin{bmatrix}y_0 & \dots & y_{K-1}\end{bmatrix}$