Homework assignment 2

Optimal control and dynamic programming (4SC000), TU/e, 2018-2019

Lecturer: Duarte Antunes

Submission: The solutions of Assignments 1, 2 should be submitted via Matlab grader and via canvas (.m files) and the solutions of Assignment 3 and 4 should be submitted only via canvas. For Assignments 3 and 4, use the solution template available on Canvas. Assignment 5 is open and you need to submit an .mpg4 video via canvas, with a maximum duration of 2 minutes, explaining the main ideas behind your solution as well as the source code (matlab function mintimeAV.m).

Grade: The overall assignment will be graded on a scale 0-10 according to the following distribution

Assignment	1	2	3	4	5	Total
Max score	2	1.5	1.5	2	3	10

There are 6 (hidden and visible) tests for each of the assignments 1,2. For each of these assignments, 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	0.4	0.4	0.5	0.6	1

Plagiarism: The matlab code will be checked with a plagiarism detection tool. In case of fraud, no points will be given for the entire assignment.

Deadline: January 10th, 23h45, 2019.

---

1. Trajectory generation using LQR

Suppose that we wish to find a smooth trajectory, in the sense that the derivatives exist up to a given order n, interpolating a set of N points specified in the 3D space $p_i = \begin{bmatrix} x_i & y_i & z_i \end{bmatrix}^\intercal$ and in time T_i, where $i \in \{1, \dots, N\}$ and T_i is an increasing sequence. In this exercise we formulate an LQR problem to provide a solution to such a problem.

We start by defining a piecewise linear trajectory interpolating the points

$$r(t) = p_i + \frac{(t - T_i)}{(T_{i+1} - T_i)}(p_{i+1} - p_i), \quad \text{for } t \in [T_i, T_{i+1}) \text{ and } i \in \{1, \dots, N-1\}$$

and a sampled version of this trajectory at times $k\tau_1$, where τ_1 is a given sampling period,

$$r_k = r(k\tau_1), \text{ for } k \in \{0, \dots, h\},$$

where $h = \lfloor \frac{T_N}{\tau_1} \rfloor$ is the floor of $\frac{T_N}{\tau_1}$. Define the nth derivative of the trajectory $p(t) = \begin{bmatrix} x(t) & y(t) & z(t) \end{bmatrix}^\intercal$ to be $u(t) = \begin{bmatrix} u_x(t) & u_y(t) & u_z(t) \end{bmatrix}^\intercal$

$$p^{(n)}(t) = u(t)$$

and suppose that it is piecewise constant

$$u(t) = u_k, \qquad t \in [k\tau_1, (k+1)\tau_1), \quad k \in \{0, \dots, h-1\}.$$

The initial position and derivatives of the trajectory up to $n-1$ are assumed to be given $p^{(j)}(0)$, $j \in \{0, 1, \dots, n-1\}$. Let $p_k := p(k\tau_1)$, for $k \in \{0, \dots, h\}$, and consider the following LQR problem

$$\min_{\{u_k | k \in \{0,\dots,h-1\}\}} \sum_{k=0}^{h-1} \|p_k - r_k\|^2 + \rho\|u_k\|^2 + \|p_h - r_h\|^2$$

where ρ is a given positive constant.

Assignment 1 Program a Matlab function providing a solution to this problem

```
[xix,xiy,xiz] = lqrtrajgeneration(P,T,X0,rho,tau1,N12)
```

where the input arguments are defined as follows

- P is an $3 \times N$ matrix such that $P = \begin{bmatrix} p_1 & p_2 & \dots & p_N \end{bmatrix}$
- T is a vector with times $T = \begin{bmatrix} T_1 & T_2 & \dots & T_N \end{bmatrix}$
- $X0$ is a $3 \times n$ matrix with the initial trajectory position and derivatives

$$X0 = \begin{bmatrix} x(0) & x'(0) & \dots & x^{(n-1)}(0) \\ y(0) & y'(0) & \dots & y^{(n-1)}(0) \\ z(0) & z'(0) & \dots & z^{(n-1)}(0) \end{bmatrix}$$

- rho $= \rho$
- τ_1 is the sampling period in the problem formulation determining the shape of the nth derivative of p.
- N_{12} is such that $\tau_2 = \frac{\tau_1}{N_{12}}$ is the output sampling period (this is used to specify the output arguments, see below).

The output arguments are

- xix is a matrix with entries

$$\begin{bmatrix} x(0) & x(\tau_2) & \dots & x(H\tau_2) \\ x'(0) & x'(\tau_2) & \dots & x'(H\tau_2) \\ & & & \\ x^{(n)}(0) & x^{(n)}(\tau_2) & \dots & x^{(n)}(H\tau_2) \end{bmatrix}$$

  where $H = N_{12}h$. Note that $x^{(n)}(H\tau_2)$ is undefined: make $x^{(n)}(H\tau_2) = x^{(n)}((H-1)\tau_2)$.
- xiy and xiz, which are defined similarly as xix but for the y and z components of p.

2. Multi-rate LQG

Suppose that, as in the standard LQG framework, we wish to control a linear plant

$$x_{k+1} = Ax_k + Bu_k + w_k$$
$$y_k = Cx_k + v_k$$

for $k \in \mathbb{N}_0 := \mathbb{N} \cup \{0\}$, where both $\{w_k | k \in \mathbb{N}_0\}$ and $\{v_k | k \in \mathbb{N}_0\}$ are sequences of independent and identically distributed Gaussian random variables, with $\mathbb{E}[w_k w_k^\intercal] = W$ and $\mathbb{E}[w_k w_k^\intercal] = V$. The performance of the system is measured by an average quadratic cost

$$\lim_{T\to\infty} \frac{1}{T}\mathbb{E}[\sum_{k=0}^{T-1} x_k^\intercal Q x_k + u_k^\intercal R u_k] \tag{1}$$

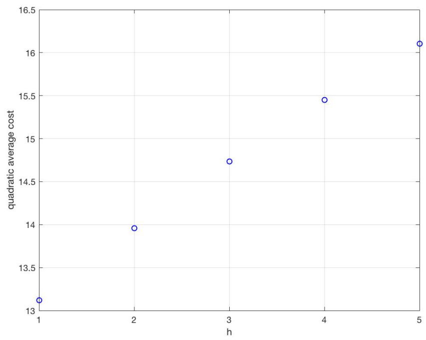
for positive definite matrices Q and R. However, suppose that the sensor information y_k is only available at times

$$t_\ell = \ell h, \quad \ell \in \mathbb{N}_0, \tag{2}$$

where h is a positive integer ($h = 1$ boilds down to the standard LQG problem). The figure below plots the average cost corresponding to such an optimal controller for several values of h and for the parameters $\tau = 0.1$,

$$A = e^{A_c s}, B = \int_0^\tau e^{A_c s} ds \begin{bmatrix} 0 \\ 1 \end{bmatrix}, C = [1\,0], \;\; W = I, \;\; V = 0.01, \;\; Q = I, \;\; R = 0.01, \;\; A_c = \begin{bmatrix} 0 & 1 \\ -1 & -1 \end{bmatrix}.$$

Note that, as expected, the performance degrades with increasing h.



Assignment 2
Program a Matlab function

```
[cost] = lqgmultirate(A,B,C,Q,R,W,V,h)
```

that computes the cost when the optimal LQG controller is applied for a given h, considering the parameters (A, B, C, Q, R, W, V) and h.

Suggestion: solve first the case $h = 1$, some tests on Matlab grader consider just this case.

3. MPC trajectory tracking

Consider the linearized model of the longitudinal movement of a quadcopter

$$m\ddot{p} = mg\theta,$$
$$I_\theta \ddot{\theta} = \tau$$

where p denotes the x coordinate with respect to a world fixed frame, θ is the pitch angle, τ is the torque, and m, g, and I_θ denote the mass, gravitation constant and the moment of inertia. By scaling the pitch angle and the torque we can write the model as a fourth order integrator

$$\dot{x} = A_c x + Bu$$

where $x = \begin{bmatrix} p & \dot{p} & \ddot{p} & \dddot{p} \end{bmatrix}$, $u = \ddddot{p}$, and

$$A_c = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad B_c = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}.$$

In fact, note that $\ddot{p} = g\theta$ and $u = \frac{g}{I_\theta}\tau$. After zero order hold with sampling period τ and considering $x_k := x(k\tau)$ and

$$u(t) = u_k, \qquad t \in [k\tau, (k+1)\tau), \quad k \in \{0, \dots, h-1\},$$

we can write

$$x_{k+1} = Ax_k + Bu_k,$$

where $A = e^{A_c\tau}$, $B = \int_0^\tau e^{A_c s} ds B_c$.

Suppose that we wish to follow a given reference $r(t)$, which takes the values $r_k := r(k\tau)$ at sampling times and that the input is constraint to

$$|u_k| < c.$$

To this effect, we use MPC with cost function at each time k,

$$\sum_{\ell=k}^{k+H-1} (p_\ell - r_\ell)^2 + \rho u_\ell^2$$

where $p_k = p(k\tau)$ and H is the prediction horizon. The MPC algorithm runs for every $k \in \mathbb{N}_0$ (consider $h = \infty$). Consider $r(t) = 3\sin(0.3t)$, $\rho = 0.01$, $\tau = 0.1$, $H = 80$, and $c = 0.4$.

Assignment 3

Using Matlab, plot the optimal $\{u_k | k \in \{0, \dots, N\}\}$ and the corresponding functions $\{p_k | k \in \{0, \dots, N\}\}$ and $\{v_k | k \in \{0, \dots, N\}\}$, against $k\tau, |k \in \{0, \dots, N\}$ where $v_k := \dot{p}(k\tau)$, for $N = 400$ and for the following initial conditions

(i) $x_0 = \begin{bmatrix} 10 & 0 & 0 & 0 \end{bmatrix}$

(ii) $x_0 = \begin{bmatrix} -20 & -10 & 0 & 0 \end{bmatrix}$

(iii) $x_0 = \begin{bmatrix} -1 & -1 & 0.1 & 0.4 \end{bmatrix}$

4. Duality

Suppose that we wish to estimate the state, at a given time $k = h$, of a linear system:

$$x_{k+1} = Ax_k + w_k, \quad k \in \mathbb{N}_0 := \{0, 1, 2, \ldots h-1\},$$

where each w_k denotes the state disturbance at time k. Assuming that A is invertible, we have:

$$x_k = A^{-1}x_{k+1} - A^{-1}\bar{w}_{k+1} \tag{3}$$

where $\bar{w}_{k+1} = w_k$. The following output measurements

$$y_k = Cx_k + v_k$$

are available at times $k \in \{0, 1, 2, \ldots, h\}$, where v_k denotes the output noise at time k. The initial state is unknown but an initial estimate, denoted by $\bar{x}_0$, is available. In order to obtain the state estimate at time h, denoted by $\bar{x}_h$, one can: (i) first solve the optimal control problem

$$\begin{aligned} J_h(x_h) = \min_{\bar{w}_1,\ldots,\bar{w}_h} & (x_0 - \bar{x}_0)^\intercal \bar{\Theta}_0^{-1}(x_0 - \bar{x}_0) + (y_0 - Cx_0)^\intercal V^{-1}(y_0 - Cx_0) \\ & + \sum_{k=1}^{h} \bar{w}_k^\intercal W^{-1}\bar{w}_k + (y_k - Cx_k)^\intercal V^{-1}(y_k - Cx_k) \end{aligned}$$

subject to (3), for given positive definite weighting matrices W, V, $\bar{\Theta}$; and then (ii) find

$$\bar{x}_h = \text{argmin } J_h(x_h).$$

This can be interpreted as follows: estimate the states x_0, $\ldots$, x_h by taking those which correspond to ('can be explained by') small values of state disturbances, output noise, and deviation from the initial condition estimate according to the penalties in this cost function and take the last state estimate of x_h.

Assignment 4 Solve the given optimal control problem and show that $\bar{x}_h$ coincides with the solution of the Kalman filter, $\hat{x}_{h|h}$ obtained by iterating, for $k \in \{0, 1, \dots, h\}$,

$$\begin{aligned}
\hat{x}_{k|k} &= \hat{x}_{k|k-1} + L_k(y_k - C\hat{x}_{k|k-1}) \\
\hat{x}_{k+1|k} &= A\hat{x}_{k|k},
\end{aligned}$$

with initial condition $\hat{x}_{0|-1}$, where, for $k \in \{0, 1, \dots, h\}$,

$$\begin{aligned}
L_k &= \Theta_{k|k-1}C^\intercal(C\Theta_{k|k-1}C^\intercal + V)^{-1} \\
\Theta_{k|k} &= \Theta_{k|k-1} - \Theta_{k|k-1}C^\intercal(C\Theta_{k|k-1}C^\intercal + V)^{-1}C\Theta_{k|k-1} \\
\Theta_{k+1|k} &= A\Theta_{k|k}A^\intercal + W
\end{aligned}$$

with initial condition $\Theta_{0|-1} = \bar{\Theta}_0$ (in this sense this control problem is the dual of the estimation problem addressed in class).

Suggestion: Consider the DP algorithm going forward in time (instead of backwards) with dynamics (3) and the initial 'cost-to-come' (instead of the cost-to-go)

$$J_0(x_0) = (x_0 - \bar{x}_0)^\intercal \bar{\Theta}_0^{-1}(x_0 - \bar{x}_0) + (y_0 - Cx_0)^\intercal V^{-1}(y_0 - Cx_0)$$

and use (multiple times!) the matrix inversion lemma

$$(A + UCV)^{-1} = A^{-1} - A^{-1}U(C^{-1} + VA^{-1}U)^{-1}VA^{-1}$$

The following identities may also be useful

$$\begin{aligned}
\Theta_{0|0}^{-1} &= \bar{\Theta}_0^{-1} + C^\intercal VC \\
(I - L_0C)^\intercal\Theta_{0|0}^{-1} &= \bar{\Theta}_0^{-1} \\
L_0^\intercal\Theta_{0|0}^{-1} &= V^{-1}C
\end{aligned}$$

5. Minimum time control of a simple 2D double integrator model [1]

Consider an omni-directional autonomous vehicle moving in the 2D track depicted in the figure below and modeled by

$$\ddot{x} = u_x$$
$$\ddot{y} = u_y$$

where (x, y) is the position and u_x and u_y are the force inputs normalized by the mass. The sampling period is $\tau = 0.1$ and a zero order hold is considered

$$u_x(t) = u_{x,k}, \quad u_y(t) = u_{y,k}, \quad t \in [k\tau, (k+1)\tau)$$

Assume that there is an input constraint taking the form

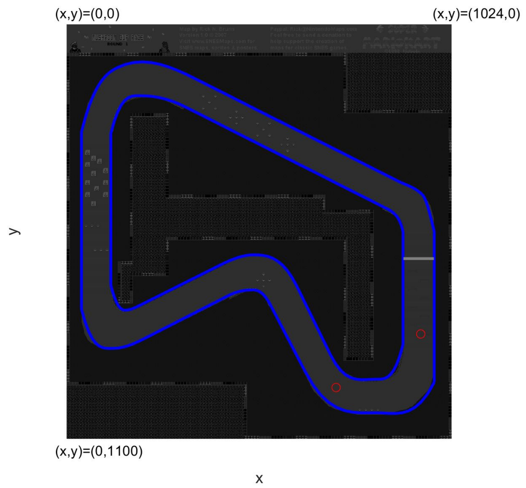
$$\|(u_{x,k}, u_{y,k})\| \leq L$$

with $L = 300$, where $\|.\|$ is the Euclidean norm. Position is measured in decimeters. The initial velocity is zero and consider two initial positions

$$x_0 \in \{\begin{bmatrix} 942 \\ 822 \end{bmatrix}, \begin{bmatrix} 717 \\ 964 \end{bmatrix}\},$$

which are indicated by the red circles in the figure. The finish line is indicated in grey in the figure. The boundaries of the track are defined by a set of straight lines and 3rd order splines. The parameters of these straight lines and splines can be found in the Matlab files provided for this assignment in canvas (see cell vectors gamma and gamma theta and control points p, q in file drawtrack.m).

The goal of this assignment is to compute a control input which allows the robot to complete one lap (anti-clockwise direction) starting from the initial position and crossing the finish line in minimum time, without stepping out of the track boundaries and satisfying the input constraints. Run checksolution.m to see an expected outcome.



[1] A version of this exercise was proposed in 2017/2018 and the videos made by the students can be seen here: `https://surfdrive.surf.nl/files/index.php/s/fZlr5GvDSo7ph6q`

Assignment 5 Program a Matlab function

```
[u] = mintimeAV(indx0)
```

which computes the $2 \times h$ control input that satisfies the specifications above, with u(1,k)$=u_{x,k-1}$ in the first row and u(2,k)$=u_{y,k-1}$ in the second row for $k \in \{1, \ldots, h\}$. The input argument indxo is 1 if the initial condition is $x_0 = \begin{bmatrix} 942 & 822 \end{bmatrix}^\mathsf{T}$ or 2 if the initial condition is $x_0 = \begin{bmatrix} 717 & 964 \end{bmatrix}^\mathsf{T}$. Note that the time to complete one lap is $h \times \tau$. Your function will be tested with checksolution.m.