# 4F3 cribs

## Question 1

(a) Let us denote $x_0, x_1, \ldots, x_N$ the sequence of nodes along the optimal path from node $x_0$ to node $x_N$. By Bellman's Principle of Optimality, the truncated sequence $x_0, x_1, \ldots, x_{N-1}$ is the optimal path from $x_0$ to $x_{N-1}$ as well (truncated problem). This principle allows to take advantage of recursive algorithms in optimization.

(b) The sequence of graphs in Figure 1 represents the main steps of the dynamic programming computation. At each step we update the cost to go, whose values are represented within each node.

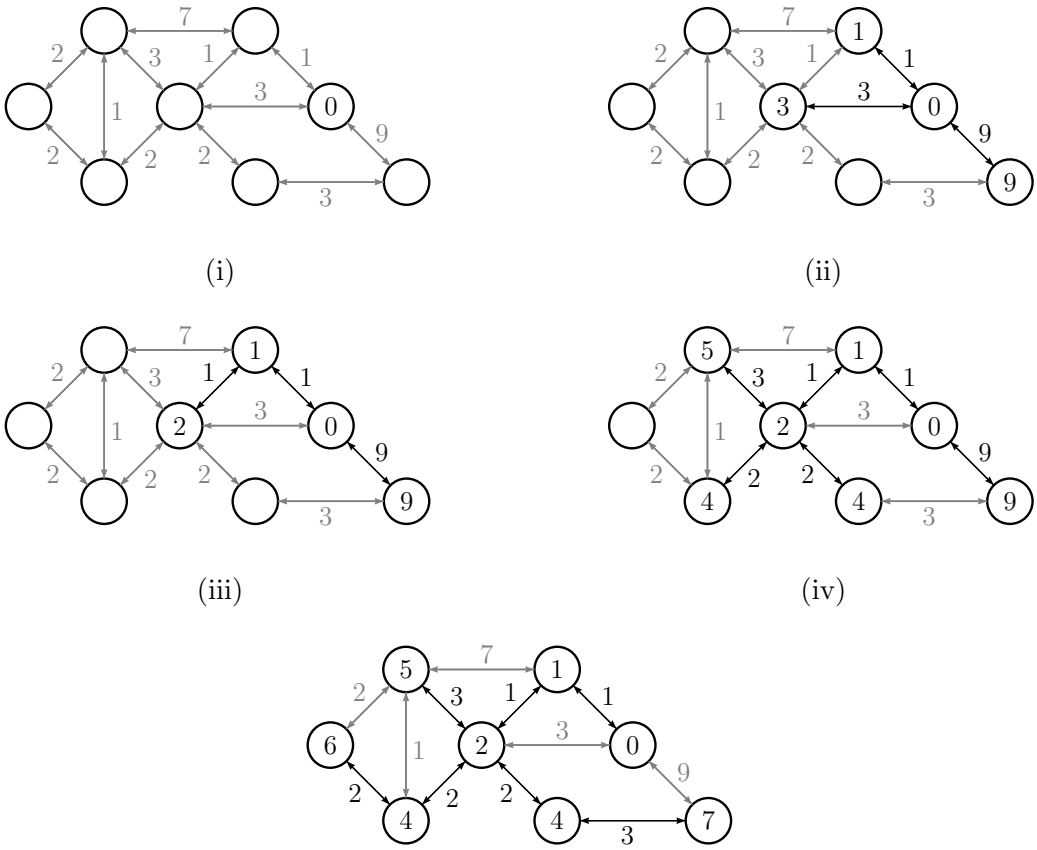



Figure 1

(c) The minimal-time routing paths needs to be recalculated, with the delay per edge increased by 1, as paths with more hops (intermediate nodes) will be penalized more. After recalculating there are some nodes with two paths with the same minimal cost.

(d) Running back and forth on the edge with negative weight makes the cost to go smaller and smaller. That is, with a negative edge the path length cost has no minimum.

(e.i) Define the index set $I = \{1, 2, \ldots, 8\}$ associated with each node. For a path from node 1 to node 7, the indicator variable $x_{ij}$ is defined as $x_{ij} = 1$ if there exists an edge from $i$ to $j$ that belongs to the path and 0 otherwise, i.e. the indicator set identifies the path. Every path from node 1 to node 7 satisfies the following three conditions

$\mathbf{C_1}$: there is a single edge entering the final node:

$$\sum_{i \in I} x_{i7} = 1 \ ;$$

$\mathbf{C_2}$: there is a single edge leaving the initial node:

$$\sum_{i \in I} x_{1i} = 1 \ ;$$

$\mathbf{C_3}$: the number of edges entering the node $j$ and the number of edges leaving the node $j$ are the same: for all $j \in I$ such that $j \neq 0$ and $j \neq 7$,

$$\sum_{i \in I} x_{ij} - \sum_{i \in I} x_{ji} = 0 \ .$$

Subject to these constraints, the suggested cost associated to the path from node 1 to node 7 is given by

$$J = \sum_{i \in I, j \in I} w_{ij} x_{ij} \ .$$

Summarizing, the mathematical optimisation problem reads

$$\min \sum_{i \in I, j \in I} w_{ij} x_{ij} \text{ subject to } x_{ij} \in \{0, 1\} \text{ and } \mathbf{C_1}, \mathbf{C_2}, \mathbf{C_3} \ .$$

(e.ii) For $x_{i,j} \geq 0$ and continuous, this is a linear program, since cost function and constraints are all linear.

# Question 2

(a) The system can be written as follows

$$\dot{x} = Ax + B_1 w_1 + B_2 u \qquad y = Cx + w_2$$

where

$$A = \begin{bmatrix} -k_{12} - d & k_{21} \\ k_{12} & -k_{21} - d \end{bmatrix} = \begin{bmatrix} -11 & 20 \\ 10 & -21 \end{bmatrix}$$

and

$$B_1 = \begin{bmatrix} 0 \\ 1 \end{bmatrix} , \quad B_2 = \begin{bmatrix} 1 \\ 0 \end{bmatrix} , \quad C = \begin{bmatrix} 0 & 1 \end{bmatrix} .$$

We can now derive the transfer function as

$$T_{w_1 \to y} = C(sI - A)^{-1} B_1 = \frac{s + 11}{s^2 + 32s + 31}$$

(b) The $\mathcal{H}_2$ norm is defined as follows

$$\|T_{w_1 \to y}\|_2^2 = \int_{-\infty}^{\infty} \text{trace} \left\{ \overline{T}_{w_1 \to y}(j\omega)^T T_{w_1 \to y}(j\omega) \right\} \, \mathrm{d}\omega$$

Its physical meaning in terms of system performance is given by the expression

$$\|y\|_\infty \leq \frac{1}{\sqrt{2\pi}} \|T_{w_1 \to y}\|_2 \|w_1\|_2 \, .$$

Thus, $\|T_{w_1 \to y}\|_2$ provides a bound on the largest amplification between the energy of the disturbance $w_1$ and the magnitude of the output $y$.

The $\mathcal{H}_\infty$ norm satisfies

$$\|T_{w_1 \to y}\|_\infty = \sup_\omega \bar{\sigma}(T_{w_1 \to y}(j\omega)).$$

Its physical meaning is well described by the expression

$$\|T_{w_1 \to y}\|_\infty = \sup_{w_1} \frac{\|T_{w_1 \to y} w_1\|_2}{\|w_1\|_2} = \sup_{w_1} \frac{\|y\|_2}{\|w_1\|_2} \, .$$

Thus, $\|T_{w_1 \to y}\|_\infty$ provides a bound on the largest amplification between the energies of the disturbance $w_1$ and the output $y$.

(c) We have that $\frac{1}{\sqrt{2\pi}} \|T_{w_1 \to y}\|_2 = \sqrt{B_1^T L B_1}$ where $L = L^T$ solves

$$A^T L + LA + C^T C = 0 \quad (L \text{ is the observability Gramian}).$$

From the latter, taking $L = \left[ \begin{smallmatrix} L_1 & L_2 \\ L_2 & L_3 \end{smallmatrix} \right]$ we have

$$\begin{bmatrix} -11 & 10 \\ 20 & -21 \end{bmatrix} \begin{bmatrix} L_1 & L_2 \\ L_2 & L_3 \end{bmatrix} + \begin{bmatrix} L_1 & L_2 \\ L_2 & L_3 \end{bmatrix} \begin{bmatrix} -11 & 20 \\ 10 & -21 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} = 0$$

which gives

$$L = \begin{bmatrix} 0.0504 & 0.0554 \\ 0.0554 & 0.0766 \end{bmatrix} \, .$$

Thus,

$$\sqrt{B_1^T L B_1} = \sqrt{0.0766} \simeq 0.2768$$

(d.i) Let $P(s)$ be a generalized plant with performance input/output pair $w$ and $z$ and control input/output pair $u$ and $y$. Tthe $\mathcal{H}_2$ optimal control problem corresponds to finding the stabilizing controller $u = K(s)y$ that minimizes $\|T_{w\to z}(s)\|_2$ in closed loop. Using linear fractional transformations, $T_{w\to z}(s) = \mathcal{F}_l(P(s), K(s))$ therefore the $\mathcal{H}_2$ optimal control problem corresponds to

$$\min_{K(s) \text{ stabilising}} \|\mathcal{F}_l(P(s), K(s))\|_2$$

(d.ii) Given the output performance $z$, the generalized plant is given by

$$\left[\begin{array}{c} \dot{x} \\ \hline z \\ y \end{array}\right] = \left[\begin{array}{c|cc} A & \begin{bmatrix} B_1 & 0 \end{bmatrix} & B_2 \\ \hline \begin{bmatrix} C_1 \\ 0 \end{bmatrix} & 0 & \begin{bmatrix} 0 \\ I \end{bmatrix} \\ C_2 & \begin{bmatrix} 0 & I \end{bmatrix} & 0 \end{array}\right] \left[\begin{array}{c} x \\ \hline w \\ u \end{array}\right]$$

where $C_2 = C_1 = \begin{bmatrix} 0 & 1 \end{bmatrix}$. All other matrices have been already defined above.

(d.ii) From the datasheet, we need to compute

$$A_k = A - B_2F - HC_2 \qquad B_k = -H = YC_2^T \qquad C_k = F = B_2^TX$$

Thus,

$$A_k = \begin{bmatrix} -11.0491 & 19.8387 \\ 10.0000 & -21.0747 \end{bmatrix} \qquad B_k = \begin{bmatrix} -0.1072 \\ -0.0747 \end{bmatrix} \qquad C_k = \begin{bmatrix} 0.0491 & 0.0542 \end{bmatrix}$$

# Question 3

(a) $\min_\theta \frac{1}{2}\theta^T Q\theta + c^T\theta$ subject to $A\theta \leq b$

(b) (i) Let $P \geq 0$ be steady state solution to the Ricatti equation in datasheet, ie

$$P = Q + A^T PA - A^T PB(R + B^T PB)^{-1}B^T PA$$

and put $K = -(R + B^T PB)^{-1}B^T PA$, which is the optimal solution, a state feedback gain, for the unconstrained infinite horizon case.

Now find $M, b$ s.t. the set $\{x : Mx \leq b\}$ is *constraint admissible* and *invariant* with respect to the control $u = Kx$ (i.e. $Mx \leq b \implies |Kx| \leq U$ and $Mx \leq b \implies M(Ax + BKx) \leq b$).

Now solve the MPC problem

$$\min_{u_0,u_1} \sum_{k=0}^{1}(x_k^T Qx_k + u_k^T Ru_k) + x_2^T Px_2$$

subject to $x_1 = Ax_0 + Bu_0$, $x_2 = Ax_1 + Bu_1 = A^2x_0 + ABu_0 + Bu_1$

i.e.

$$\min_{u_0,u_1} x_0^T Qx_0 + (Ax_0 + Bu_0)^T Q(Ax_0 + Bu_0) +$$
$$(A^2x_0 + ABu_0 + Bu_1)^T P(A^2x_0 + ABu_0 + Bu_1) + u_0Ru_0 + u_1Ru_1$$

$$= \text{ const } + \min_{u_0,u_1} \begin{bmatrix} u_0 \\ u_1 \end{bmatrix}^T \begin{bmatrix} B^T QB + B^T A^T PAB + R & B^T A^T PB \\ B^T PAB & B^T PB + R \end{bmatrix} \begin{bmatrix} u_0 \\ u_1 \end{bmatrix}$$
$$+2 \begin{bmatrix} x_0^T A^T QB + x_0^T A^{2^T} PAB & x_0^T A^{2^T} PB \end{bmatrix} \begin{bmatrix} u_0 \\ u_1 \end{bmatrix}$$

s.t

$$\begin{bmatrix} I & \\ -I & \\ MAB & B \end{bmatrix} \begin{bmatrix} u_0 \\ u_1 \end{bmatrix} \leq \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ b - A^2x_0 \end{bmatrix}$$

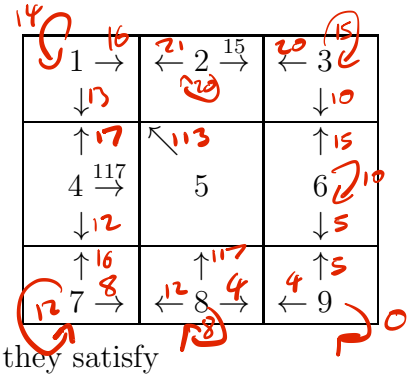
(ii) The control above is suboptimal because of the added constaint that $Mx \leq b$ at step 2. Increasing $N$ pushes this constraint further into the future, and the resulting controller wii be closer to optimal, and eventually optimal.

(iii) For large $x_0$ it may be impossible to get inside the control invariant set in 2 time steps.

# Question 4

(a) $Q(s, a)$ is the cost of taking action $a$ from state $s$ and taking optimal actions thereafter. It enables you to find the optimal action, by enumeration, without knowing the model or reward.

(b) (i)




These are optimal, since they satisfy

$$Q(s, a) = c(s) + \min_{a'} Q(s', a')$$

in each case (where $s'$ is that state reached by taking action $a$ at state $s$.)

(ii) The optimal cost is 13, associated with the path $1 \to 4 \to 7 \to 8 \to 9$ (and remain at 9 thereafter).

(iii) After each state action pair has been visited a sufficiently large number of times $Q$ will have converged to the optimal value. There is then a 5% chance of taking the suboptimal action $4 \to 5$, and the same for $8 \to 5$. There is thus a 10% chance of an extra cost of <u>at least</u> 100. Hence the cost $> 13 + 0.1 * 100 = 23$.

(iv) For SARSA the iteration is instead

$$Q(s, a) \leftarrow c(s) + Q(s', a')$$

where $a'$ is selected according to the current policy. If this policy is $\epsilon$-greedy, then actions to states 4 and 8 will be marked with a higher cost and the path found will be $1 \to 2 \to 3 \to 6 \to 9$ with an optimal cost of slightly greater than 16.

Q1. Dynamic programming.

A quite popular question that was attempted by most candidates. Parts (a)-(d) associated with finding the minimal-time routing path was generally well answered. Part (e) was more challenging and many candidates had difficulties formulating the constraints associated with the optimization problem.

Q2. H2 optimal control.

The question was attempted by about two third of the candidates. Part (a) associated with providing a state space representation and finding a transfer function was generally well answered. Many candidates had difficulties in part (c) associated with the computation of the H2 norm. There were also good attempts for part (d) where candidates were asked to formulate and solve the H2 optimal control problem.

Q3. Model Predictive Control.

A popular question on using MPC to solve the constrained LQR problem. Attempted by most candidates and with many excellent answers. Most marks were lost by a lack of precision in answering a).

Q4 Reinforcement learning

A less popular question, attempted by about half the candidates, comparing Q-learning and SARSA on a cliff edge walking problem. The basics were well done, but ost marks were lost on b) parts iii) and iv), with many candidates not grasping the effect of using epsilon-greedy action selection.