EGT3
ENGINEERING TRIPOS PART IIB

Worked Solutions

Module 4F3

AN OPTIMISATION BASED APPROACH TO CONTROL

The mark allocations shown in the right margin are the same as on the question paper.

1 (a) Discuss the connection between Dynamic Programming and the Hamilton-Jacobi-Bellman (HJB) equation. Discuss also one advantage and one disadvantage of using the HJB equation to solve optimal control problems. [15%]

Solution. The Hamilton-Jacobi-Bellman equation is the continuous time version of the Dynamic Programming equation.
Advantage: It allows to obtain analytical solutions to optimal control problems.
Disadvantage: As a PDE it can be computationally intensive to solve numerically.

(b) Consider the system

$$x_{k+1} = \lambda x_k + u_k \tag{1}$$

where $x_k \in \mathbb{R}$, $u_k \in \mathbb{R}$ for $k = 0, \ldots, h-1$, $x_h \in \mathbb{R}$, and $\lambda \in \mathbb{R}$ is a constant. Consider also the cost

$$\gamma x_h^2 + \sum_{k=0}^{h-1} x_k^2 + u_k^2$$

where $\gamma > 0$ is a constant.

(i) Show using dynamic programming that the minimum value of this cost over all control inputs u_k, for a given initial condition x_0, is a quadratic function of the initial condition. [25%]

Solution. The Dynamic Programming (DP) Equation is

$$V(x,k) = \min_u \left\{ x^2 + u^2 + V(\lambda x + u, k+1) \right\}$$

We use the trial solution $V(x,k) = r_k x^2$. Substituting in the DP equation for $k = h-1$ we have

$$\begin{aligned} r_{h-1}x^2 &= \min_u \left\{ x^2 + u^2 + (\lambda x + u)^2 \gamma \right\} \\ &= \min_u \left\{ (1 + \lambda^2\gamma)x^2 + 2\lambda u \gamma x + (1+\gamma)u^2 \right\} \end{aligned}$$

Differentiating the RHS w.r.t. u and setting this equal to zero we get

$$(1+\gamma)u^* = -\lambda\gamma x \Rightarrow u^* = -\frac{\lambda\gamma}{1+\gamma}x$$

Therefore we have

$$\begin{aligned} r_{k-1}x^2 &= \left[1 + \lambda^2\gamma - 2\frac{\lambda^2\gamma^2}{1+\gamma} + \frac{\lambda^2\gamma^2}{1+\gamma} \right] x^2 \\ &= \underbrace{\left[1 + \lambda^2\gamma - \frac{\lambda^2\gamma^2}{1+\gamma} \right]}_{>0} x^2 \end{aligned} \tag{2}$$

 (cont.

Hence the DP equation is satisfied, and by induction the optimal cost is a quadratic function of the initial condition.

(ii) Consider system (1) with $\lambda = 1$. If the minimum cost for a horizon of length $h = 4$ is $2x_0^2$, find the minimum cost in terms of the initial condition when $h = 6$. [10%]

Solution. We apply the DP iteration (2) twice (with $\gamma = r_k$). Applying this once with $r_2 = 2$ we get $V(1, x) = (1+2-4/3)x^2 = (5/3)x^2$. Applying this again with $r_1 = 5/3$ we get $V(0, x) = [1 + 5/3 - (5/3)^2)/(1 + 5/3)]x^2 = (8/3 - 25/24)x^2 = (39/24)x^2$

(c) For the system (1) consider the problem of minimizing the cost

$$\sum_{k=0}^{\infty} x_k^2 + u_k^2 \tag{3}$$

over the control inputs $u_k \in \mathbb{R}$. Discuss whether control inputs such that x_k becomes zero in finite time, can lead to a lower cost from that obtained from the best linear controller of the form $u_k = Kx_k$ where K is a constant. Find also an expression for the minimum value of the cost in (3) for a given initial condition x_0. [25%]

Solution. The optimal controller is linear (even when arbitrary inputs are considered), hence x_k tends to zero asymptotically when an optimal controller is used. Therefore an input sequence and a corresponding trajectory where x_k is zero in finite time cannot have a lower cost.

To find the optimal cost we solve the Riccati equation

$$r = 1 + \lambda^2 r - \frac{\lambda^2 r^2}{1 + r}$$

which gives

$$r^2 - \lambda^2 r - 1 = 0$$

This has positive solution

$$r = \frac{1}{2}\left(\lambda^2 + \sqrt{\lambda^4 + 4}\right)$$

The minimum cost is

$$r = \frac{1}{2}x_0\left(\lambda^2 + \sqrt{\lambda^4 + 4}\right)$$

(d) Consider the optimal control problem in part (c), but the control input u_k can now take only discrete values in the set $\{-2, -1, 0, 1, 2\}$. Find the optimal control inputs and the minimum cost for initial condition $x_0 = 2$, with $\lambda = 1$ in system (1). [25%]

Solution. If the state reaches $x_k = 0$ for some k it is optimal to stay at zero. If $x_k \neq 0$ for all k then the cost is infinite. Therefore under an optimal policy the system satisfies $x_k = 0$ for $k \geq h$ for some h.

Trajectories that satisfy this property include ones generated by the input sequences $(-1,-1), (-2,0), (-1,-2,1), (1,-2,-1)$.

All other trajectories that reach zero have higher cost as they include additional stages k with $x_k \neq 0$. The minimum cost is achieved with the inputs $u_0 = -1, u_1 = -1$ and $u_k = 0$ for $k \geq 2$. The value of the minimum cost is $(2^2 + 1) + (1 + 1) = 7$.

2 (a) Explain what is meant by the $\mathcal{H}_2$ optimal control problem. Also discuss in what ways $\mathcal{H}_2$ optimal control provides a generalization to the Linear Quadratic Regulator problem. [10%]

Solution. An $\mathcal{H}_2$ optimal control problem is a problem where we aim to find a controller that minimizes the $\mathcal{H}_2$ norm of a prescribed closed loop transfer function. It extends LQR control to the output feedback case, where not all states are directly available to the controller. LQR control is a special case of an $\mathcal{H}_2$ optimal control problem, via an appropriate choice of the B matrix in terms of the initial conditions when an $\mathcal{H}_2$ optimal control problem is formulated.

(b) Consider a stable linear system with state space representation $\dot{x} = Ax+Bw,\ y = Cx$. We denote the transfer function from w to y by G.

(i) Show that if there exists a constant matrix $M > 0$ such that $V(x) = x^T Mx$ satisfies

$$\frac{dV}{dt} \leq \gamma^2 w^T w - y^T y$$

for all signals w and corresponding outputs y such that $\int_0^\infty w^T w dt < \infty$, where $\gamma > 0$ is a constant, then $\|G\|_\infty \leq \gamma$. [15%]

Solution.

$$\int_0^\infty \frac{dV}{dt} dt = V(\infty) - V(0) = 0 \leq \gamma^2 \|w\|_2^2 - \|y\|_2^2 \tag{4}$$

which implies $\|G\|_\infty \leq \gamma$.

(ii) Use this to derive a Riccati equation through which one can verify if $\|G\|_\infty \leq \gamma$. [25%]

Solution.

$$\max_w \left\{ \dot{V} + y^T y - \gamma^2 w^T w \right\} \leq 0$$

and if equality holds it ensures $\|G\|_\infty \leq \gamma$. Substituting the relation

$$\frac{dV}{dt} = \dot{x}^T Mx + xM\dot{x}^T$$

we have

$$(Ax + Bw)^T Mx + x^T M(Ax + Bw) + (Cx)^T Cx - \gamma^2 w^T w = 0$$

Differentiating w.r.t. w and setting the derivative equal to zero we get $w = \gamma^{-2} B^T Mx$. Substituting in the equation we get

$$x^T (A+\gamma^{-2} BB^T M)^T Mx + x^T M(A+\gamma^{-2} BB^T M)x + x^T C^T Cx - \gamma^{-2} x^T MBB^T Mx = 0$$

which implies

$$A^T M + MA + C^T C + \gamma^{-2} MBB^T M = 0$$

Hence, if there exists $M > 0$ such that the equation above holds then this implies $\|G\|_\infty \leq \gamma$.

(c) Consider the system

$$\begin{aligned} \dot{x} &= -0.2x + u + w_1 \\ y &= x + w_2 \\ z_1 &= x, \quad z_2 = u \end{aligned}$$

where $x(t) \in \mathbb{R}$ is the state of the system, $w_1(t), w_2(t)$ are external disturbances, and $u(t)$ is the control input. We also denote $w = \begin{bmatrix} w_1 & w_2 \end{bmatrix}^T$, and $z = \begin{bmatrix} z_1 & z_2 \end{bmatrix}^T$. A controller with transfer function $K(s)$ is implemented such that $\bar{u}(s) = K(s)\bar{y}(s)$, where $\bar{u}(s), \bar{y}(s)$ are the Laplace transforms of the signals u, y respectively.

(i) Find a controller $K(s)$ that will ensure the $\mathcal{H}_\infty$ norm of the transfer function from w to z is less than or equal to 2. [30%]

Solution. Comparing with the expressions in the datasheet we have $A = -0.2$, $B_1 = B_2 = C_1 = C_2 = 1$, $\gamma = 2$. The Riccati equation for X is

$$2AX + 1 - (1 - \gamma^{-2})X^2 = 0$$

and has solution

$$X = \frac{A + \sqrt{A^2 + 1 - \gamma^{-2}}}{1 - \gamma^{-2}} = 0.9184$$

The Riccati equation for Y is

$$2\hat{A}Y + 1 - (1 - \gamma^{-2}F^2)Y^2 = 0$$

where $\hat{A} = A + \gamma^{-2}X$, $F = X$, and has solution

$$Y = \frac{\hat{A} + \sqrt{\hat{A}^2 + 1 - \gamma^{-2}F^2}}{1 - \gamma^{-2}F^2} = 1.1639$$

The controller has state spare realization given by $A_K = \hat{A} - F - H$, $B_K = -Y$, $C_K = X$, $D_K = 0$ and transfer function $K(s) = -1.069/(s + 2.053)$.

(ii) Consider a controller of the form $u = ky$ where $k \leq 0$ is a constant. Find an expression for the $\mathcal{H}_\infty$ norm of the transfer function from w to x. Discuss also if this $\mathcal{H}_\infty$ norm can be made arbitrarily small via appropriate choice of k. [20%]

Solution. With this control policy the new A matrix is $A + k$. To find the $\mathcal{H}_\infty$ norm we use the Riccatti equation derived in part (b) with $B = \begin{bmatrix} 1 & k \end{bmatrix}$, $C = 1$.
This becomes

$$2(A + k)M + 1 + \gamma^{-2}(1 + k^2)M^2 = 0$$

and has solutions

$$M = \frac{-2(A + k) \pm \sqrt{4(A + k)^2 - 4\gamma^{-2}(1 + k^2)}}{2\gamma^{-2}(1 + k^2)}$$

A positive solution exists for $4(A + k)^2 - 4\gamma^{-2}(1 + k^2) \geq 0$ which implies

$$\gamma^2 \geq \frac{1 + k^2}{(A + k)^2}$$

So the $\mathcal{H}_\infty$ norm is

$$\frac{\sqrt{1 + k^2}}{|-0.2 + k|}$$

The expression cannot be made arbitrarily small by varying k.

3 **Predictive control.**

Consider the discrete-time linear system

$$x(k+1) = Ax(k) + Bu(k),$$

with $x(k) \in \mathbb{R}^n$, $u(k) \in \mathbb{R}^m$, and assume full state measurement is available.

At time k, a predictive controller computes an input sequence

$$u = \{u_0, u_1, \ldots, u_{N-1}\}$$

and associated predicted states $\{x_i\}_{i=0}^{N}$ satisfying

$$x_0 = x(k), \qquad x_{i+1} = Ax_i + Bu_i, \quad i = 0, 1, \ldots, N-1,$$

to minimise the finite-horizon cost

$$V(x(k), u) = x_N^\top P x_N + \sum_{i=0}^{N-1} \left(x_i^\top Q x_i + u_i^\top R u_i \right),$$

where $Q > 0$, $R > 0$, and $P \geq 0$ are given.

The *receding horizon* control law applies

$$u(k) = \kappa(x(k)) := u_0^\star(x(k)),$$

where $u^\star(x(k))$ minimises $V(x(k), u)$.

(a) State clearly the receding horizon principle. Assume that K is any feedback gain such that $\rho(A + BK) < 1$ and that $P > 0$ satisfies

$$(A + BK)^\top P (A + BK) - P \;\leq\; -Q - K^\top R K.$$

Using the "shifted sequence + terminal tail" argument, show that the optimal value function $V^\star(x) = \min_u V(x, u)$ satisfies

$$V^\star(x(k+1)) \leq V^\star(x(k)) - x(k)^\top Q x(k).$$

Hence explain why the origin is an asymptotically stable equilibrium point of the *unconstrained* receding-horizon closed loop system. [25%]

Solution. The *receding horizon principle* is:

> At each time k, measure $x(k)$, solve the finite-horizon optimisation to obtain an optimal sequence $\{u_0^\star, \ldots, u_{N-1}^\star\}$, apply only the first input $u(k) = u_0^\star$, discard the remaining inputs, increment $k \leftarrow k+1$, and repeat.

Let $x = x(k)$ and let $u^\star(x) = \{u_0^\star, \dots, u_{N-1}^\star\}$ be optimal at time k with corresponding predicted state sequence $\{x_i^\star\}_{i=0}^N$ (so $x_0^\star = x$ and $x_{i+1}^\star = Ax_i^\star + Bu_i^\star$).
Define a candidate input sequence for the optimisation at time $k+1$ by shifting the previously optimal inputs and appending the stabilising tail law $u = Kx$:

$$\tilde{u}(x) = \{u_1^\star(x), u_2^\star(x), \dots, u_{N-1}^\star(x), Kx_N^\star(x)\}.$$

Because the prediction model matches the plant (unconstrained, no disturbances), the state at the next sampling instant is

$$x(k+1) = Ax + Bu_0^\star(x) = x_1^\star(x).$$

By optimality at time $k+1$,

$$V^\star(x(k+1)) \le V\big(x(k+1), \tilde{u}(x)\big).$$

Now compare $V\big(x(k+1), \tilde{u}(x)\big)$ with $V^\star(x) = V(x, u^\star(x))$. All stage terms for indices $i = 1, \dots, N-1$ cancel, leaving

$$\begin{aligned} V\big(x(k+1), \tilde{u}(x)\big) - V^\star(x) = &\Big(x_{N+1}^\top P x_{N+1} + x_N^{\star\top} Q x_N^\star + (Kx_N^\star)^\top R (Kx_N^\star)\Big) - x_N^{\star\top} P x_N^\star \\ &- \Big(x^\top Q x + (u_0^\star)^\top R u_0^\star\Big), \end{aligned}$$

where $x_{N+1} = (A + BK)x_N^\star$ under the tail law. Hence

$$\begin{aligned} V\big(x(k+1), \tilde{u}(x)\big) - V^\star(x) &= x_N^{\star\top}\Big((A+BK)^\top P (A+BK) - P + Q + K^\top R K\Big) x_N^\star - x^\top Q x - (u_0^\star)^\top R u_0^\star \\ &\le -x^\top Q x, \end{aligned}$$

using the assumed matrix inequality for P and the fact that $-(u_0^\star)^\top R u_0^\star \le 0$. Therefore

$$V^\star(x(k+1)) \le V^\star(x(k)) - x(k)^\top Q x(k).$$

Since $Q > 0$ and $R > 0$, the optimal value $V^\star(x)$ is positive definite (and $V^\star(0) = 0$). The strict decrease above shows $V^\star$ is a Lyapunov function for the closed loop. Hence $x(k) \to 0$ as $k \to \infty$, i.e. the origin is an asymptotically stable equilibrium.

(b) Now consider the *constrained* MPC problem with mixed linear constraints

$$Mx_i + Eu_i \le b, \quad i = 0, 1, \dots, N-1,$$

and a terminal constraint

$$M_N x_N \le b_N.$$

Show that the constrained MPC problem at time k can be written as a standard quadratic program (QP)

$$\min_{\theta} \ \frac{1}{2}\theta^\top H\theta + f^\top\theta \quad \text{s.t.} \quad A_{\text{eq}}\theta = b_{\text{eq}}, \ \ G\theta \le h,$$

using the stacked decision vector

$$\theta = \begin{bmatrix} u_0 \\ x_1 \\ u_1 \\ x_2 \\ \vdots \\ u_{N-1} \\ x_N \end{bmatrix}.$$

Give the *block structure* of H, A_{eq}, b_{eq}, G, and h in terms of $A, B, Q, R, P, M, E, b, M_N, b_N$ and the measured state $x(k)$. (You do *not* need to write every block entry explicitly, but you must show the matrix patterns clearly.) [35%]

Solution. With decision vector $\theta = [u_0^\top \ x_1^\top \ u_1^\top \ x_2^\top \ \dots \ u_{N-1}^\top \ x_N^\top]^\top$, the cost can be written as

$$V(x(k), u) = x(k)^\top Q x(k) + \theta^\top \Omega\, \theta, \qquad \Omega = \text{blkdiag}(R, Q, R, Q, \dots, R, P).$$

The constant $x(k)^\top Q x(k)$ may be dropped, so a standard QP objective is obtained with

$$H = 2\Omega, \qquad f = 0.$$

Dynamics as equalities. The constraints $x_{i+1} = Ax_i + Bu_i$ become $A_{\text{eq}}\theta = b_{\text{eq}}$. There are N block rows, each contributing n scalar equations. The block-sparse pattern is

$$A_{\text{eq}} = \begin{bmatrix} -B & I & 0 & 0 & \cdots & 0 & 0 \\ 0 & -A & -B & I & \cdots & 0 & 0 \\ \vdots & & & \ddots & \ddots & & \vdots \\ 0 & 0 & 0 & \cdots & -A & -B & I \end{bmatrix}, \qquad b_{\text{eq}} = \begin{bmatrix} Ax(k) \\ 0 \\ \vdots \\ 0 \end{bmatrix}.$$

The first block row enforces $x_1 - Ax(k) - Bu_0 = 0$; subsequent rows enforce $x_{i+1} - Ax_i - Bu_i = 0$ for $i = 1, \dots, N-1$.

Inequalities. For stage $i = 0$, the constraint is $Mx(k) + Eu_0 \le b$, i.e.

$$Eu_0 \le b - Mx(k).$$

 (cont.

For stages $i = 1, \dots, N-1$, $Mx_i + Eu_i \le b$ couples (x_i, u_i). Finally the terminal constraint is $M_N x_N \le b_N$. All can be stacked as $G\theta \le h$ with the block pattern

$$G = \begin{bmatrix} E & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & M & E & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & M & \cdots & 0 & 0 \\ \vdots & & & & \ddots & & \vdots \\ 0 & 0 & 0 & 0 & \cdots & M & E \\ 0 & 0 & 0 & 0 & \cdots & 0 & M_N \end{bmatrix}, \qquad h = \begin{bmatrix} b - Mx(k) \\ b \\ b \\ \vdots \\ b \\ b_N \end{bmatrix}.$$

Any equivalent formulation (for example, eliminating x_i to obtain an input-only QP) is acceptable.

(c) Even though the plant is linear, $\kappa(x) = u_0^\star(x)$ is generally *nonlinear* when constraints are present. Explain why. Describe briefly what is meant by an *explicit MPC* controller, and state what functional form $\kappa(x)$ takes when the optimisation is a strictly convex QP with linear constraints. [15%]

Solution. With constraints, the optimiser depends on which inequality constraints are *active* at the optimum. As the measured state x changes, the set of active constraints can change, so the mapping $x \mapsto u_0^\star(x)$ switches between different affine expressions. Hence the overall feedback law $\kappa(x)$ is generally nonlinear even though the plant and constraints are linear.

An *explicit MPC* controller is obtained by solving the (multi-)parametric QP offline with x as the parameter, yielding a closed-form piecewise-defined feedback law. For a strictly convex QP (e.g. $H > 0$) with linear constraints, one obtains a polyhedral partition of the feasible set into regions $\{\mathcal{R}_i\}$ on each of which

$$\kappa(x) = u_0^\star(x) = K_i x + v_i, \qquad x \in \mathcal{R}_i,$$

i.e. a *piecewise affine* state feedback.

(d) Define:

(i) an *invariant set* for a closed-loop system $x^+ = f(x)$;

(ii) a *constraint-admissible* set for a feedback law $u = \kappa(x)$ with constraint set $\mathcal{Z} = \{(x, u) : Mx + Eu \le b\}$.

Explain how one can choose the terminal set $\{x : M_N x \le b_N\}$ to be a constraint-admissible invariant set for the linear tail law $u = Kx$, and how this choice guarantees

recursive feasibility of the constrained receding-horizon controller. Outline (no detailed calculations required) an algorithm to construct such a terminal set. [25%]

Solution.

(i) A set $\mathcal{S} \subset \mathbb{R}^n$ is *(positively) invariant* for $x^+ = f(x)$ if $x \in \mathcal{S} \implies f(x) \in \mathcal{S}$.

(ii) A set $\mathcal{S} \subset \mathbb{R}^n$ is *constraint-admissible* for the feedback $u = \kappa(x)$ and constraint set $\mathcal{Z} = \{(x, u) : Mx + Eu \leq b\}$ if

$$(x, \kappa(x)) \in \mathcal{Z} \quad \text{for all } x \in \mathcal{S}.$$

To guarantee recursive feasibility, choose a stabilising tail gain K and a terminal set $\mathcal{S}_f = \{x : M_N x \leq b_N\}$ such that

(i) *tail admissibility:* $Mx + E(Kx) \leq b$ for all $x \in \mathcal{S}_f$ (so applying the tail law respects constraints);

(ii) *tail invariance:* $(A + BK)x \in \mathcal{S}_f$ for all $x \in \mathcal{S}_f$.

If the MPC problem is feasible at time k, then shifting the optimal sequence $(u_1^\star, \ldots, u_{N-1}^\star)$ and appending $Kx_N^\star$ produces a *feasible candidate* at time $k + 1$: stage constraints remain satisfied by construction, and invariance ensures the new terminal state remains in $\mathcal{S}_f$. Thus feasibility at k implies feasibility at $k + 1$ (recursive feasibility).

Construction (typical polyhedral algorithm). Let

$$\mathcal{S}_0 = \{x : (M + EK)x \leq b\}$$

be the set of states for which the tail input $u = Kx$ satisfies the mixed constraints at the current step. Then iterate

$$\mathcal{S}_{i+1} = \{x \in \mathcal{S}_i : (A + BK)x \in \mathcal{S}_i\} = \mathcal{S}_i \cap (A + BK)^{-1}\mathcal{S}_i,$$

until convergence ($\mathcal{S}_{i+1} = \mathcal{S}_i$), yielding the maximal invariant set contained in $\mathcal{S}_0$. Represent the resulting polytope as $M_N x \leq b_N$ and use it as the terminal set.

4 **Reinforcement learning.**

Consider the discounted Markov decision process (MDP) with states

$$\mathcal{S} = \{s_0, s_1, s_2, s_T\},$$

where s_T is an absorbing terminal state. The action set is $\mathcal{A} = \{a, b\}$. The per-step *cost* is $c(s, a)$, and the discount factor is $\lambda = 0.9$.

The transition model is:

•From s_0:

–action a: cost 1, next state s_1 with probability 1;

–action b: cost 1, next state s_2 with probability 1.

•From s_1:

–action a: cost 1, next state s_T with probability 0.9, and s_1 with probability 0.1;

–action b: cost 2, next state s_2 with probability 1.

•From s_2:

–action a: cost 1, next state s_T with probability 1;

–action b: cost 2, next state s_0 with probability 1.

•From s_T: cost 0, remains in s_T.

(a) Define the value function $V^\pi(s)$ and action-value function $Q^\pi(s, a)$ for a deterministic policy π. Write down the Bellman equation satisfied by V^π. Write down the Bellman *optimality* equation for the optimal value function $V(s) = \min_\pi V^\pi(s)$. [15%]

Solution. For a deterministic policy $\pi : \mathcal{S} \to \mathcal{A}$,

$$V^\pi(s) = \mathbb{E}\left[\sum_{t=0}^{\infty} \lambda^t \, c\big(s_t, \pi(s_t)\big) \;\middle|\; s_0 = s\right],$$

and the action-value function is

$$Q^\pi(s, a) = \mathbb{E}\big[c(s, a) + \lambda V^\pi(s_1) \;\big|\; s_0 = s,\ a_0 = a\big].$$

For non-terminal s, the Bellman equation for V^π is

$$V^\pi(s) = c\big(s, \pi(s)\big) + \lambda \sum_{s' \in \mathcal{S}} P(s'|s, \pi(s))\, V^\pi(s'), \qquad V^\pi(s_T) = 0.$$

The Bellman optimality equation (cost-minimisation form) is

$$V(s) = \min_{a \in \mathcal{A}} \left(c(s,a) + \lambda \sum_{s' \in \mathcal{S}} P(s'|s,a)\, V(s') \right), \qquad V(s_T) = 0.$$

(b) *Value iteration:* starting from $V_0(s) = 0$ for all states (and keeping $V_k(s_T) = 0$), compute V_1 and V_2 for s_0, s_1, s_2. From V_2, state the greedy policy at each non-terminal state. [25%]

Solution. Value iteration is

$$V_{k+1}(s) = \min_{a \in \mathcal{A}} \left(c(s,a) + \lambda \sum_{s'} P(s'|s,a) V_k(s') \right), \qquad V_k(s_T) = 0.$$

With $V_0 \equiv 0$:
First iteration.

$$\begin{aligned}
V_1(s_0) &= \min\{1 + 0.9V_0(s_1),\ 1 + 0.9V_0(s_2)\} = \min\{1, 1\} = 1,\\
V_1(s_1) &= \min\{1 + 0.9(0.9V_0(s_T) + 0.1V_0(s_1)),\ 2 + 0.9V_0(s_2)\} = \min\{1, 2\} = 1,\\
V_1(s_2) &= \min\{1 + 0.9V_0(s_T),\ 2 + 0.9V_0(s_0)\} = \min\{1, 2\} = 1.
\end{aligned}$$

Second iteration.

$$\begin{aligned}
V_2(s_0) &= \min\{1 + 0.9V_1(s_1),\ 1 + 0.9V_1(s_2)\} = \min\{1.9, 1.9\} = 1.9,\\
V_2(s_1) &= \min\{1 + 0.9(0.9V_1(s_T) + 0.1V_1(s_1)),\ 2 + 0.9V_1(s_2)\} = \min\{1.09, 2.9\} = 1.09,\\
V_2(s_2) &= \min\{1 + 0.9V_1(s_T),\ 2 + 0.9V_1(s_0)\} = \min\{1, 2.9\} = 1.
\end{aligned}$$

Greedy policy w.r.t. V_2. Choose $\pi(s) \in \arg\min_a \{c(s,a) + \lambda \mathbb{E}[V_2(s')]\}$:

$$\begin{aligned}
s_0:&\quad a \mapsto 1 + 0.9V_2(s_1) = 1 + 0.9 \cdot 1.09 = 1.981, \quad b \mapsto 1 + 0.9V_2(s_2) = 1 + 0.9 \cdot 1 = 1.9,\\
s_1:&\quad a \mapsto 1 + 0.9(0.9 \cdot 0 + 0.1V_2(s_1)) = 1 + 0.9 \cdot 0.109 = 1.0981, \quad b \mapsto 2 + 0.9V_2(s_2) = 2.9,\\
s_2:&\quad a \mapsto 1 + 0.9 \cdot 0 = 1, \quad b \mapsto 2 + 0.9V_2(s_0) = 2 + 0.9 \cdot 1.9 = 3.71.
\end{aligned}$$

Thus one greedy choice is

$$\pi(s_0) = b, \qquad \pi(s_1) = a, \qquad \pi(s_2) = a.$$

(cont.

(c) *Q-learning vs. SARSA (off-policy vs. on-policy):* two episodes are observed (each terminates upon reaching s_T):

•Episode 1: $(s_0, a, c = 1, s_1)$, $(s_1, a, c = 1, s_T)$;

•Episode 2: $(s_0, a, c = 1, s_1)$, $(s_1, a, c = 1, s_T)$.

Assume initial values $Q_0(s, a) = 0$ for all (s, a), learning rate $\alpha = 0.5$, discount $\lambda = 0.9$. Process the data in the order shown, updating after each transition.

(i) Using *Q-learning*,

$$Q(s, a) \leftarrow (1 - \alpha)Q(s, a) + \alpha\Big(c + \lambda \min_{b \in \mathcal{A}} Q(s', b)\Big),$$

compute the final Q-values after all transitions have been processed.

(ii) Using *SARSA*,

$$Q(s, a) \leftarrow (1 - \alpha)Q(s, a) + \alpha\Big(c + \lambda Q(s', a')\Big),$$

where a' is the next action actually taken in the episode (as listed), compute the final Q-values.

(iii) Briefly explain why the two algorithms give different values here, and what this illustrates about off-policy vs. on-policy learning.

[30%]

Solution. Initially $Q_0(s, a) = 0$ for all pairs, with $Q(s_T, \cdot) = 0$.

(i) *Q-learning.*
Episode 1.

$$Q(s_0, a) \leftarrow 0.5\Big(1 + 0.9 \min\{Q(s_1, a), Q(s_1, b)\}\Big) = 0.5(1 + 0) = 0.5,$$
$$Q(s_1, a) \leftarrow 0.5\Big(1 + 0.9 \min\{Q(s_T, a), Q(s_T, b)\}\Big) = 0.5(1 + 0) = 0.5.$$

Episode 2. At this point $Q(s_1, a) = 0.5$ but $Q(s_1, b) = 0$, so $\min_b Q(s_1, b) = 0$.

$$Q(s_0, a) \leftarrow (1 - 0.5)\, 0.5 + 0.5\Big(1 + 0.9 \cdot 0\Big) = 0.25 + 0.5 = 0.75,$$
$$Q(s_1, a) \leftarrow (1 - 0.5)\, 0.5 + 0.5\Big(1 + 0\Big) = 0.25 + 0.5 = 0.75.$$

All other $Q(s, \cdot)$ remain 0.

(ii) *SARSA.*

Episode 1. The first transition $(s_0, a, 1, s_1)$ is followed in the episode by action a at s_1.

$$Q(s_0, a) \leftarrow 0.5\Big(1 + 0.9\, Q(s_1, a)\Big) = 0.5(1 + 0) = 0.5,$$
$$Q(s_1, a) \leftarrow 0.5\Big(1 + 0.9\, Q(s_T, \cdot)\Big) = 0.5(1 + 0) = 0.5.$$

Episode 2. Now $Q(s_1, a) = 0.5$, and again the next action is a.

$$Q(s_0, a) \leftarrow (1 - 0.5)\, 0.5 + 0.5\Big(1 + 0.9 \cdot 0.5\Big) = 0.25 + 0.5 \cdot 1.45 = 0.975,$$
$$Q(s_1, a) \leftarrow (1 - 0.5)\, 0.5 + 0.5(1 + 0) = 0.75.$$

Again, all other pairs remain 0.

(iii) *Explanation.* Q-learning is *off-policy*: it updates towards the greedy continuation value $\min_b Q(s', b)$ regardless of what action is actually taken next. SARSA is *on-policy*: it updates using $Q(s', a')$ for the action a' actually taken. Here $Q(s_1, b)$ is never updated and remains at 0, so Q-learning uses $\min\{Q(s_1, a), Q(s_1, b)\} = 0$ in the update for $Q(s_0, a)$ and therefore propagates the cost-to-go from s_1 back to s_0 less strongly than SARSA.

(d) *Function approximation and "deep" Q-learning:*

(i) Give two reasons why tabular $Q(s, a)$ becomes infeasible as the state dimension grows (or when s is continuous). [10%]

(ii) Consider a parameterised approximation $Q_\theta(x, a)$ and a target network with parameters θ^- held fixed for multiple updates. For a single sample (x, a, c, x'), define a squared TD loss

$$L(\theta) = \big(y - Q_\theta(x, a)\big)^2, \qquad y = c + \lambda \min_{b \in \mathcal{A}} Q_{\theta^-}(x', b).$$

Derive one stochastic gradient descent update for θ. [10%]

(iii) Explain the purpose of (A) a replay buffer and (B) a target network in stabilising learning. [10%]

[30%]

Solution.

(i) Two standard reasons:

- *Curse of dimensionality / storage*: the number of state(-action) entries grows exponentially with dimension, so memory and data requirements become prohibitive.

•*Continuous spaces / lack of generalisation*: if s (or a) is continuous, a table would require infinitely many entries, and nearby states do not generalise to each other.

(ii) With target y treated as constant w.r.t. θ (since θ^- is held fixed),

$$\nabla_\theta L(\theta) = -2\big(y - Q_\theta(x,a)\big)\,\nabla_\theta Q_\theta(x,a).$$

A single stochastic gradient step with step size $\eta > 0$ is

$$\theta \leftarrow \theta - \eta\nabla_\theta L(\theta) = \theta + 2\eta\big(y - Q_\theta(x,a)\big)\,\nabla_\theta Q_\theta(x,a),$$

often written (absorbing the factor 2 into the step size) as

$$\theta \leftarrow \theta + \alpha\big(y - Q_\theta(x,a)\big)\,\nabla_\theta Q_\theta(x,a).$$

(iii) (A) A *replay buffer* stores experience tuples and allows random minibatch sampling. This breaks temporal correlations in successive samples and improves data efficiency by reusing past experience.
(B) A *target network* holds θ^- fixed for multiple updates so that the bootstrapping target y changes slowly. This reduces the "moving target" effect and improves numerical stability (helping to prevent divergence).

END OF PAPER

THIS PAGE IS BLANK