

AAE 590SC: Stochastic Control

Problem Set 3

Travis Hastreiter

2 October, 2025

Contents

Problem 1	3
2.5.i)	4
2.5.ii)	6
Problem 2	8
Problem 3	10
Problem 4	12
3.1.i)	12
3.1.ii)	12
Problem 5	13
3.2.i)	13
3.2.ii)	13

Code Listing

Problem 1

Derive Kalman Filter

For part i) I referenced Essential Aspects of Bayesian Data Imputation by Holt and Nguyen
Holt, William, and Duy Nguyen. "Essential Aspects of Bayesian Data Imputation." Marist College, 2022.

2.5.i)

$$\textcircled{1} \quad 2.5 - 2.7 \quad x \sim N(\bar{x}, P_{xx}), y \sim N(\bar{y}, P_{yy})$$

2.5.i Let x, y be jointly Gaussian random vectors s.t.

$$(x, y) \sim N\left(\begin{pmatrix} \bar{x} \\ \bar{y} \end{pmatrix}, \begin{pmatrix} P_{xx} & P_{xy} \\ P_{yx} & P_{yy} \end{pmatrix}\right), P_{yy} > 0$$

i) show that $p(x|y) \sim N(\hat{x}, P_{x|y})$

where $\hat{x} = \bar{x} + P_{xy}P_{yy}^{-1}(y - \bar{y})$ and $P_{x|y} = P_{xx} - P_{xy}P_{yy}^{-1}P_{yx}$

$$p(x|y) \sim N\left(\bar{x} + P_{xy}P_{yy}^{-1}(y - \bar{y}), P_{xx} - P_{xy}P_{yy}^{-1}P_{yx}\right)$$

$$\text{show } \mathbb{E}(p(x|y)) = \bar{x} + P_{xy}P_{yy}^{-1}(y - \bar{y}) = \hat{x}$$

$$= \mathbb{E}\left(\frac{p(x \cap y)}{p(y)}\right)$$

$$P_{xx} - P_{xy}P_{yy}^{-1}P_{yx} = \mathbb{E}((x - \hat{x})(x - \hat{x})^T) = P_{x|y}$$

$$\mathbb{E}((\underbrace{(x - \bar{x}) - P_{xy}P_{yy}^{-1}(y - \bar{y})}_{\text{let } x = \bar{x} + P_{xy}P_{yy}^{-1}(y - \bar{y})})(\underbrace{(x - \bar{x}) - P_{xy}P_{yy}^{-1}(y - \bar{y})}_{\text{let } x = \bar{x} + P_{xy}P_{yy}^{-1}(y - \bar{y})}^T))$$

$$P_{xx} = \mathbb{E}((x - \bar{x})(x - \bar{x})^T)$$

$$\text{Proof: } p(x|y | \bar{x}, P_{x|y}) = \frac{1}{(2\pi)^{(d+d_2)/2} |P_{x|y}|^{1/2}} \exp\left(-\frac{1}{2} \begin{pmatrix} x - \bar{x} \\ y - \bar{y} \end{pmatrix}^T \begin{pmatrix} P_{xx} & P_{xy} \\ P_{yx} & P_{yy} \end{pmatrix}^{-1} \begin{pmatrix} x - \bar{x} \\ y - \bar{y} \end{pmatrix}\right)$$

Note the following identity is directly obtained

$$\begin{pmatrix} I & -P_{xy}P_{yy}^{-1} \\ 0 & I \end{pmatrix} \begin{pmatrix} P_{xx} & P_{xy} \\ P_{yx} & P_{yy} \end{pmatrix} \begin{pmatrix} I & 0 \\ -P_{yy}^{-1}P_{yx} & I \end{pmatrix} = \begin{pmatrix} P_{xy}/P_{yy} & 0 \\ 0 & P_{yy} \end{pmatrix} \quad (1)$$

$$\text{Let } P_{x|y}/P_{yy} = P_{xx} - P_{xy}P_{yy}^{-1}P_{yx}$$

① Taking determinant of both sides of (1) we get

$$|P_{x|y}| = \begin{vmatrix} P_{xx} & P_{xy} \\ P_{yx} & P_{yy} \end{vmatrix} = |P_{x|y}/P_{yy}| |P_{yy}| \quad (2)$$

Inverting both sides of (1) .

$$\begin{pmatrix} P_{xx} & P_{xy} \\ P_{yx} & P_{yy} \end{pmatrix}^{-1} = \begin{pmatrix} I & 0 \\ -P_{yy}^{-1}P_{yx} & I \end{pmatrix} \begin{pmatrix} (P_{x|y}/P_{yy})^{-1} & 0 \\ 0 & P_{yy}^{-1} \end{pmatrix} \begin{pmatrix} I & -P_{xy}P_{yy}^{-1} \\ 0 & I \end{pmatrix}$$

Resulting from this

$$\begin{aligned} p(x, y | \hat{x}, P_{x|y}) &\propto \exp\left(-\frac{1}{2}(x-\bar{x})^T \begin{pmatrix} P_{xx} & P_{xy} \\ P_{yx} & P_{yy} \end{pmatrix}^{-1} \begin{pmatrix} x-\bar{x} \\ y-\bar{y} \end{pmatrix}\right) \\ &= \exp\left(-\frac{1}{2}(x-\bar{x} - P_{xy}P_{yy}^{-1}(y-\bar{y}))^T (P_{x|y}/P_{yy})(x-\bar{x} - P_{xy}P_{yy}^{-1}(y-\bar{y}))\right) \\ &\quad \cdot \exp\left(-\frac{1}{2}(y-\bar{y})^T P_{yy}^{-1}(y-\bar{y})\right) \end{aligned}$$

From (2) it can be seen

$$\begin{aligned} (2\pi)^{(d_1+d_2)/2} |P_{x|y}|^{1/2} &= (2\pi)^{(d_1+d_2)/2} \left(|P_{x|y}/P_{yy}| |P_{yy}| \right)^{1/2} \\ &= (2\pi)^{d_1/2} |P_{x|y}/P_{yy}|^{1/2} (2\pi)^{d_2/2} |P_{yy}|^{1/2} \end{aligned}$$

Hence

$$\begin{aligned} p(x, y | \hat{x}, P_{x|y}) &= \frac{1}{(2\pi)^{d_1/2} |P_{x|y}/P_{yy}|^{1/2}} \\ &\quad \cdot \exp\left(-\frac{1}{2}(x-\bar{x} - P_{xy}P_{yy}^{-1}(y-\bar{y}))^T (P_{x|y}/P_{yy})^{-1} (x-\hat{x} - P_{xy}P_{yy}^{-1}(y-\bar{y}))\right) \\ &\quad \cdot \frac{1}{(2\pi)^{d_2/2} |P_{yy}|^{1/2}} \exp\left(-\frac{1}{2}(y-\bar{y})^T P_{yy}^{-1}(y-\bar{y})\right) \\ &= p(x|y) p(y) \end{aligned}$$

Therefore

$$\begin{aligned} p(x|y) &= \frac{1}{(2\pi)^{d_1/2} |P_{x|y}/P_{yy}|^{1/2}} \\ &\quad \cdot \exp\left(-\frac{1}{2}(x-\bar{x} - P_{xy}P_{yy}^{-1}(y-\bar{y}))^T (P_{x|y}/P_{yy})^{-1} (x-\bar{x} - P_{xy}P_{yy}^{-1}(y-\bar{y}))\right) \\ &\sim \mathcal{N}(\hat{x}, P_{x|y}) \end{aligned}$$

where

$$\hat{x} = x + P_{xy}P_{yy}^{-1}(y-\bar{y})$$

$$\underline{P_{x|y} = P_{x|y}/P_{yy} = P_{xx} - P_{xy}P_{yy}^{-1}P_{yx}}$$

2.5.ii)

()

i) Derive Kalman filter formula (2.52) and (2.53)

Specifically, show if $p(x_{t+1}|y_{0:t-1}) \sim N(\hat{x}_{t+1|t-1}, P_{t+1|t-1})$
then $p(x_t|y_{0:t-1}) \sim N(\hat{x}_{t|t-1}, P_{t|t-1})$ where $\hat{x}_{t|t-1}$ and $P_{t|t-1}$
are given by (2.52). Then using part i) show
that $p(x_t|y_{0:t}) \sim N(\hat{x}_{t|t}, P_{t|t})$ where $\hat{x}_{t|t}$ and $P_{t|t}$ are given by (2.53)

(2.52) prediction;

$$\hat{x}_{t+1} = A_t \hat{x}_{t-1|t-1} \quad (2.52a)$$

$$P_{t+1|t-1} = A_t P_{t-1|t-1} A_t^T + W_t \quad (2.52b)$$

(2.53) update:

$$\hat{x}_{t|t} = \hat{x}_{t-1|t-1} + P_{t-1|t-1} C_t^T (C_t P_{t-1|t-1} C_t^T + V_t)^{-1} (y_t - C_t \hat{x}_{t-1|t-1}) \quad (2.53a)$$

$$P_{t|t} = P_{t-1|t-1} - P_{t-1|t-1} C_t^T (C_t P_{t-1|t-1} C_t^T + V_t)^{-1} C_t P_{t-1|t-1} \quad (2.53b)$$

Show if $p(x_{t+1}|y_{0:t-1}) \sim N(\hat{x}_{t+1|t-1}, P_{t+1|t-1})$ thus $p(x_t|y_{0:t-1}) \sim N(\hat{x}_{t|t-1}, P_{t|t-1})$ where $\hat{x}_{t|t-1}$ from (2.52a)Linear model $x_{t+1} = A_t x_t + w_t$ $w_t \sim N(0, W_t)$ $P_{t+1|t-1}$ from (2.52b)so $p(A_t x_{t-1|t-1}) \sim N(A_t x_{t-1|t-1}, A_t P_{t-1|t-1} A_t^T + W_t)$

$$x_{t|t-1} = A_t x_{t-1|t-1} + v_t$$

$$\Rightarrow p(x_t|y_{0:t-1}) \sim N(A_t \hat{x}_{t-1|t-1}, A_t P_{t-1|t-1} A_t^T + W_t)$$

variance adds for Gaussian

- Now using part i) show
that $p(x_t | y_{0:t}) \sim N(\hat{x}_{t|t}, P_{t|t})$ where $\hat{x}_{t|t}$ & $P_{t|t}$ given by (2.53)

$$\text{From i) } \hat{x} = \bar{x} + P_{xy} P_{yy}^{-1} (y - \bar{y})$$

$$P_{xly} = P_{xly}/P_{yy} = P_{xx} - P_{xy} P_{yy}^{-1} P_{yx}$$

$$\begin{aligned}
 & \text{Want } \hat{x}_{t+1} = \hat{x}_{t|t-1} + P_{t|t-1}^{-\frac{1}{2}} C_t^T (C_t P_{t|t-1} C_t^T + V_t)^{-1} (y_t - C_t \hat{x}_{t|t-1}) \\
 & P_{t|t} = P_{t|t-1} - P_{t|t-1}^{-\frac{1}{2}} C_t^T (C_t P_{t|t-1} C_t^T + V_t)^{-1} C_t P_{t|t-1} \\
 & \downarrow \text{directly} \\
 & P_{xx}^{-1} y_{t|t} \sim N(C_t x_{t|t}, P_{t|t})
 \end{aligned}$$

Expected value of y is $c_t \hat{x}_{t+1}$

$$\text{because } Y_t = (\epsilon X_t + v_t, v_t \sim N(0, v_t))$$

$$\Rightarrow E(Y_t) = E(f_t x) \equiv C E(w) + 1$$

$$\mathbb{E}((y_t - \hat{c}_t \hat{x}_{t+1|t}) (y_t - \hat{c}_t \hat{x}_{t+1|t})^T) = c_t P_{t+1|t} (c_t^T + v_t)$$

How to solve $P_{YX} = C_6 P_{H+1}$?

$$= E((y_t - (\epsilon \hat{x}_{t+1|t})) (x_t - \hat{x}_{t+1|t})^T)$$

$$= \mathbb{E}((C_t x_t - C_t \hat{x}_{t+1}) (x_t - \hat{x}_{t+1})^\top)$$

$$= \mathbb{E}(c_e (x_e - \hat{x}_{e|e-1})(x_e \hat{x}_{e|e-1})^T)$$

$$= C \mathbb{E}((x_t - \hat{x}_{t+1})(x_t \hat{x}_{t+1})^T)$$

$$= c_e \rho_{e1e-1}$$

WAN F

KALMAN FILTER DERIVED!!

Problem 2

(2.6) POMDP

Suppose \mathcal{X}, \mathcal{U} , and \mathcal{Y} are finite and consider the finite-horizon POMDP. Show that $J_t(\pi_t)$ is piecewise linear and concave in π_t for each $t = 0, 1, \dots, T$.

Relevant properties of affine functions

- affine functions are concave and convex
- minimum of a sum of affine functions is concave

Relevant properties of concave functions

- sum of concave functions is concave
- minimum of a set of concave functions is concave

We are dealing with discrete \mathcal{X} so the cost functions can be written as

$$\tilde{C}_t(\pi_t, u_t) = \int_{\mathcal{X}_T} C_t^{\text{exit}}(x_t) \pi_t(x_t) dx_t = \begin{pmatrix} C_t(x_1, u_t) \\ C_t(x_2, u_t) \\ \vdots \\ C_t(x_{n_x}, u_t) \end{pmatrix}^T \begin{pmatrix} \pi_t(x_1) \\ \pi_t(x_2) \\ \vdots \\ \pi_t(x_{n_x}) \end{pmatrix}$$

$$\tilde{C}_T^{\text{exit}}(\pi_T) = \int_{\mathcal{X}_T} C_T^{\text{exit}}(x_T) \pi_T(x_T) dx_T = \begin{pmatrix} C_T^{\text{exit}}(x_1) \\ C_T^{\text{exit}}(x_2) \\ \vdots \\ C_T^{\text{exit}}(x_{n_x}) \end{pmatrix}^T \begin{pmatrix} \pi_T(x_1) \\ \pi_T(x_2) \\ \vdots \\ \pi_T(x_{n_x}) \end{pmatrix}$$

Now it is obvious that $\tilde{C}_t(\pi_t, u_t)$ and $\tilde{C}_T^{\text{exit}}(\pi_T)$ are both affine in π_t .

$J_T(\pi_T) = \tilde{C}_T^{\text{exit}}(\pi_T)$ so $J_T(\pi_T)$ is also affine in π_t .

$$J_t(\pi_t) = \min_{u_t \in \mathcal{U}} \tilde{C}_t(\pi_t, u_t) + \int_{\pi_{t+1} \in \Delta(\mathcal{X})} J_{t+1}(\pi_{t+1}) \tau(\pi_{t+1} | \pi_t, u_t) d\pi_{t+1}$$

$$\text{Let } \tau(u_t) = \begin{pmatrix} \tau_{u_t}(1|1) & \tau_{u_t}(1|2) & \dots & \tau_{u_t}(1|n_x) \\ \tau_{u_t}(2|1) & \tau_{u_t}(2|2) & \dots & \tau_{u_t}(2|n_x) \\ \vdots & \vdots & \ddots & \\ \tau_{u_t}(n_x|1) & \tau_{u_t}(n_x|2) & \dots & \tau_{u_t}(n_x|n_x) \end{pmatrix} = \tau(\pi_{t+1} | \pi_t, u_t) \text{ where } n_x \text{ is the number of states.}$$

Now, it can be seen that

$$\pi_{t+1} = \tau(u_t) \pi_t$$

Focusing on $t = T - 1$

$$\int_{\pi_T \in \Delta(\mathcal{X})} J_T(\pi_T) \tau(\pi_T | \pi_{T-1}, u_{T-1}) d\pi_T = \begin{pmatrix} C_T^{\text{exit}}(x_1) \\ C_T^{\text{exit}}(x_2) \\ \vdots \\ C_T^{\text{exit}}(x_{n_x}) \end{pmatrix}^T \tau(u_{T-1}) \begin{pmatrix} \pi_{T-1}(x_1) \\ \pi_{T-1}(x_2) \\ \vdots \\ \pi_{T-1}(x_{n_x}) \end{pmatrix}$$

$$J_{T-1}(\pi_{T-1}) = \min_{u_{T-1} \in \mathcal{U}} \begin{pmatrix} C_{T-1}(x_1, u_{T-1}) \\ C_{T-1}(x_2, u_{T-1}) \\ \vdots \\ C_{T-1}(x_{n_x}, u_{T-1}) \end{pmatrix}^T \begin{pmatrix} \pi_{T-1}(x_1) \\ \pi_{T-1}(x_2) \\ \vdots \\ \pi_{T-1}(x_{n_x}) \end{pmatrix} + \begin{pmatrix} C_T^{\text{exit}}(x_1) \\ C_T^{\text{exit}}(x_2) \\ \vdots \\ C_T^{\text{exit}}(x_{n_x}) \end{pmatrix}^T \tau(u_{T-1}) \begin{pmatrix} \pi_{T-1}(x_1) \\ \pi_{T-1}(x_2) \\ \vdots \\ \pi_{T-1}(x_{n_x}) \end{pmatrix}$$

$$J_{T-1}(\pi_{T-1}) = \min_{u_{T-1} \in \mathcal{U}} \left(\begin{pmatrix} C_{T-1}(x_1, u_{T-1}) \\ C_{T-1}(x_2, u_{T-1}) \\ \vdots \\ C_{T-1}(x_{n_x}, u_{T-1}) \end{pmatrix}^T + \begin{pmatrix} C_T^{\text{exit}}(x_1) \\ C_T^{\text{exit}}(x_2) \\ \vdots \\ C_T^{\text{exit}}(x_{n_x}) \end{pmatrix}^T \tau(u_{T-1})) \begin{pmatrix} \pi_{T-1}(x_1) \\ \pi_{T-1}(x_2) \\ \vdots \\ \pi_{T-1}(x_{n_x}) \end{pmatrix} \right)$$

It is obvious that $J_{T-1}(\pi_{T-1})$ is the minimum of a set of affine functions of π_{T-1} (one for each control input). This results in an a convex piecewise affine function of π_{T-1} with up to n_u affine regions.

For $t = T-2, \dots, 1, 0$ the integration must be done by summing over the integrals of each affine region in $J_t(\pi_t)$ resulting in a sum of concave piecewise affine functions which when minimized is a concave piecewise affine function. Therefore, by induction J_t is piecewise affine and concave in $\pi_t \forall t = 0, 1, \dots, T$.

Problem 3

(2.7) Stochastic LQG

Define the value function

$$J_t(\pi_t) = \min_{\mu_{t:T-1}} \sum_{k=t}^{T-1} \mathbb{E}(\|x_k\|_{Q_k}^2 + \|u_k\|_{R_k}^2) + \mathbb{E}\|x_T\|_{Q_T}^2$$

Prove (2.69) by showing that

$$J_t(\pi_t) = Tr(S_t P_{t|t}) + \sum_{k=t}^{T-1} (Tr(W_k S_{k+1}) + Tr(\Theta_k P_{k|k}))$$

for each $t = 0, 1, \dots, T$

Note: (2.68) says the optimal policy of the value function is

$$\begin{aligned} \hat{x}_{t|t} &= \hat{x}_{t|t-1} + L_t(y_t - C_t \hat{x}_{t|t-1}) \\ u_t &= K_t \hat{x}_{t|t} \\ \hat{x}_{t+1|t} &= A_t \hat{x}_{t|t} + B_t u_t \end{aligned}$$

So Kalman filter + LQR

Trace property: For a gaussian random variable a with mean \bar{a} and covariance A

$$\mathbb{E}(a^T M a) = \bar{a}^T M \bar{a} + Tr(MA)$$

Looking at $t = T - 1$

$$\begin{aligned} J_t(\pi_t) &= \min_{\mu_{t:T-1}} \sum_{k=t}^{T-1} \mathbb{E}(\|x_k\|_{Q_k}^2 + \|u_k\|_{R_k}^2) + \mathbb{E}\|x_T\|_{Q_T}^2 \\ J_{T-1}(\pi_{T-1}) &= \mathbb{E}(\|x_k\|_{Q_k}^2 + \|u_k\|_{R_k}^2) + \mathbb{E}\|x_T\|_{Q_T}^2 \\ Q_T = S_T \implies J_{T-1}(\pi_{T-1}) &= \mathbb{E}(\|x_k\|_{Q_k}^2 + \|u_k\|_{R_k}^2) + \mathbb{E}(x_T^T S_T x_T) \\ x_{t+1} = A_t x_t + B_t u_t + w_t \implies \mathbb{E}(x_T^T S_T x_T) &= \mathbb{E}((A_{T-1} x_{T-1} + B_{T-1} u_{T-1} + w_{T-1})^T S_T (A_{T-1} x_{T-1} + B_{T-1} u_{T-1} + w_{T-1})) \\ \mathbb{E}(w_t) = 0 \implies \mathbb{E}(x_T^T S_T x_T) &= \mathbb{E}((A_{T-1} x_{T-1} + B_{T-1} u_{T-1})^T S_T (A_{T-1} x_{T-1} + B_{T-1} u_{T-1})) + Tr(S_T W_{T-1}) \\ \implies J_{T-1}(\pi_{T-1}) &= \mathbb{E}(x_{T-1}^T Q_{T-1} x_{T-1} + u_{T-1}^T R_{T-1} u_{T-1} + (A_{T-1} x_{T-1})^T S_T (A_{T-1} x_{T-1}) + 2(B_{T-1} u_{T-1})^T S_T (A_{T-1} x_{T-1}) \\ &\quad + (B_{T-1} u_{T-1})^T S_T B_{T-1} u_{T-1}) + Tr(S_T W_{T-1}) \\ M_t = B_t^T S_t B_t + R_t \implies J_{T-1}(\pi_{T-1}) &= \mathbb{E}(x_{T-1}^T Q_{T-1} x_{T-1} + u_{T-1}^T M_{T-1} u_{T-1} + (A_{T-1} x_{T-1})^T S_T (A_{T-1} x_{T-1}) \\ &\quad + 2(B_{T-1} u_{T-1})^T S_T (A_{T-1} x_{T-1})) + Tr(S_T W_{T-1}) \\ &\quad \text{Completing the square} \\ u_{T-1}^T M_{T-1} u_{T-1} + 2(B_{T-1} u_{T-1})^T S_T (A_{T-1} x_{T-1}) &= (u_{T-1} + M_{T-1}^{-1} B_{T-1}^T S_T A_{T-1} x_{T-1})^T M_{T-1} (u_{T-1} + M_{T-1}^{-1} B_{T-1}^T S_T A_{T-1} x_{T-1}) \\ -(B_{T-1}^T S_T A_{T-1} x_{T-1})^T M_{T-1}^{-1} (B_{T-1}^T S_T A_{T-1} x_{T-1}) &= know S_t = A_t^T S_{t+1} A_t - A_t^T S_{t+1} B_t M_t^{-1} B_t^T S_{t+1} A_t + Q_t \\ \implies J_{T-1}(\pi_{T-1}) &= \mathbb{E}(x_{T-1}^T S_{T-1} x_{T-1} + (u_{T-1} + M_{T-1}^{-1} B_{T-1}^T S_{T-1} A_{T-1} x_{T-1})^T M_{T-1} (u_{T-1} + M_{T-1}^{-1} B_{T-1}^T S_{T-1} A_{T-1} x_{T-1})) \\ &\quad + Tr(S_T W_{T-1}) \\ &\quad know u^* = -M_t^{-1} B_t^T S_t A_t \hat{x}_t \\ \implies J_{T-1}(\pi_{T-1}) &= \mathbb{E}(x_{T-1}^T S_{T-1} x_{T-1}) \end{aligned}$$

$$\begin{aligned}
& + (M_{T-1}^{-1} B_{T-1}^T S_{T-1} A_{T-1} (\hat{x}_{T-1} - x_{T-1}))^T M_{T-1} (M_{T-1}^{-1} B_{T-1}^T S_{T-1} A_{T-1} (\hat{x}_{T-1} - x_{T-1})) \\
& \quad \text{know } \Theta_t = K_t^T M_t K_t \text{ and } K_t = -M_t^{-1} B_t^T S_t A_t \\
\implies J_{T-1}(\pi_{T-1}) &= \mathbb{E}(x_{T-1}^T S_{T-1} x_{T-1} + (\hat{x}_{T-1} - x_{T-1})^T \Theta_{T-1} (\hat{x}_{T-1} - x_{T-1})) + Tr(S_T W_{T-1}) \\
& \quad \text{From trace rules } \mathbb{E}(x_{T-1}^T S_{T-1} x_{T-1}) = \hat{x}_{T-1}^T S_{T-1} \hat{x}_{T-1} + Tr(S_{T-1} P_{T-1|T-1}) \\
\implies J_{T-1}(\pi_{T-1}) &= \hat{x}_{T-1}^T S_{T-1} \hat{x}_{T-1} + Tr(S_{T-1} P_{T-1|T-1}) + Tr(\Theta_{T-1} P_{T-1|T-1}) + Tr(S_T W_{T-1})
\end{aligned}$$

Now we need to get the value function for $t = T-2, T-3, \dots, 1, 0$. This can be done in a similer way to in exercise 2.4) Finally,

$$J_t(\pi_t) = \|\hat{x}_t\|_{S_t}^2 + Tr(S_t P_{t|t}) + \sum_{k=1}^{T-1} (Tr(W_k S_{k+1}) + Tr(\Theta_k P_{k|k}))$$

Problem 4

(3.1) Value Iteration Stopping Criteria

3.1.i)

Show: that $\|J_k - J^*\|_\infty \leq \gamma^k \|J_0 - J^*\|_\infty$

Solution:

From Lemma 3.1, we know $\|TJ - T\bar{J}\|_\infty \leq \gamma \|J - \bar{J}\|_\infty$. Also, $TJ^* = J^*$.

Proof by induction.

Base Case: $k = 1$. Let $J = J_0, \bar{J} = J^* \implies \|J_1 - J^*\|_\infty \leq \gamma \|J - J^*\|_\infty$

Inductive Hypothesis: $\|J_k - J^*\|_\infty \leq \gamma^k \|J_0 - J^*\|_\infty \implies \|J_{k+1} - J^*\|_\infty \leq \gamma^{k+1} \|J_0 - J^*\|_\infty$

Inductive Step: Assume $\|J_k - J^*\|_\infty \leq \gamma^k \|J_0 - J^*\|_\infty$

Let $J = J_k, \bar{J} = J^*$

$$\begin{aligned} TJ &= TJ_k = J_{k+1}, \quad T\bar{J} = TJ^* = J^* \\ \|J_{k+1} - J^*\|_\infty &\leq \gamma \|J_k - J^*\|_\infty \\ \|J_{k+1} - J^*\|_\infty &\leq \gamma (\gamma^k \|J_0 - J^*\|_\infty) \\ \|J_{k+1} - J^*\|_\infty &\leq \gamma^{k+1} \|J_0 - J^*\|_\infty \end{aligned}$$

Which proves our inductive hypothesis. Therefore, by induction $\|J_k - J^*\|_\infty \leq \gamma^k \|J_0 - J^*\|_\infty$.

3.1.ii)

Show: that the stopping condition $\|J_{k+1} - J_k\|_\infty \leq \epsilon \implies \|J_{k+1} - J^*\|_\infty \leq \frac{\epsilon\gamma}{1-\gamma}$

Solution:

Note the identity: $\frac{1}{1-x} = 1 + x + x^2 + \dots + x^k + \dots$. This also means $\frac{x}{1-x} = x + x^2 + \dots + x^k + \dots$

The triangle inequality says $\|a + b\|_\infty \leq \|a\|_\infty + \|b\|_\infty$

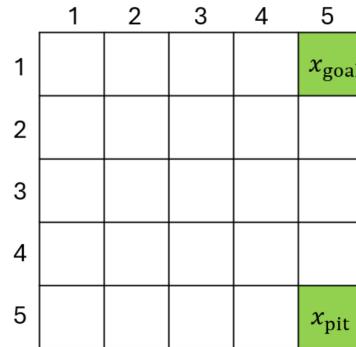
From Lemma 3.1 and the argument in 3.1.i) we know $\|J_{k+a+1} - J_{k+a}\|_\infty \leq \gamma^a \|J_{k+1} - J_k\|_\infty$

$$\begin{aligned} \|J_{k+1} - J^*\|_\infty &= \|J_{k+1} - J_{k+2} + J_{k+2} - J^*\|_\infty \leq \|J_{k+2} - J_{k+1}\|_\infty + \|J_{k+2} - J^*\|_\infty \\ &\implies \|J_{k+1} - J^*\|_\infty \leq \|J_{k+2} - J_{k+1}\|_\infty + \|J_{k+3} - J_{k+2}\|_\infty + \|J_{k+3} - J^*\|_\infty \\ \|J_{k+1} - J^*\|_\infty &\leq \|J_{k+2} - J_{k+1}\|_\infty + \|J_{k+3} - J_{k+2}\|_\infty + \|J_{k+4} - J_{k+3}\|_\infty + \dots \\ \|J_{k+1} - J^*\|_\infty &\leq \gamma \|J_{k+1} - J_k\|_\infty + \gamma^2 \|J_{k+1} - J_k\|_\infty + \gamma^3 \|J_{k+1} - J_k\|_\infty + \dots \\ \|J_{k+1} - J^*\|_\infty &\leq (\gamma + \gamma^2 + \gamma^3 + \dots) \|J_{k+1} - J_k\|_\infty = \frac{\gamma}{1-\gamma} \|J_{k+1} - J_k\|_\infty \\ \therefore \|J_{k+1} - J_k\|_\infty &\leq \epsilon \implies \|J_{k+1} - J^*\|_\infty \leq \frac{\epsilon\gamma}{1-\gamma} \end{aligned}$$

Problem 5

(3.2) Value Iteration on Gridworld

Consider the gridworld setup in Example 3.1 and assume $c_{\text{pit}} = +1$, $c_{\text{goal}} = -1$, $c_{\text{step}} = 0.05$ and $p = 0.2$.

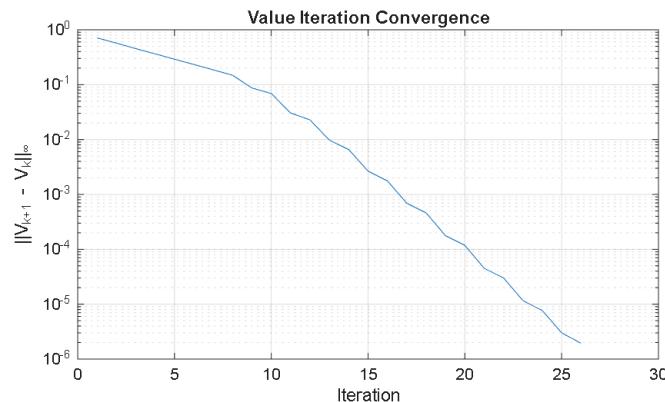


3.2.i)

Using: Algorithm 6 compute the value function. Hint: You should obtain a value function as shown in Figure 3.3.

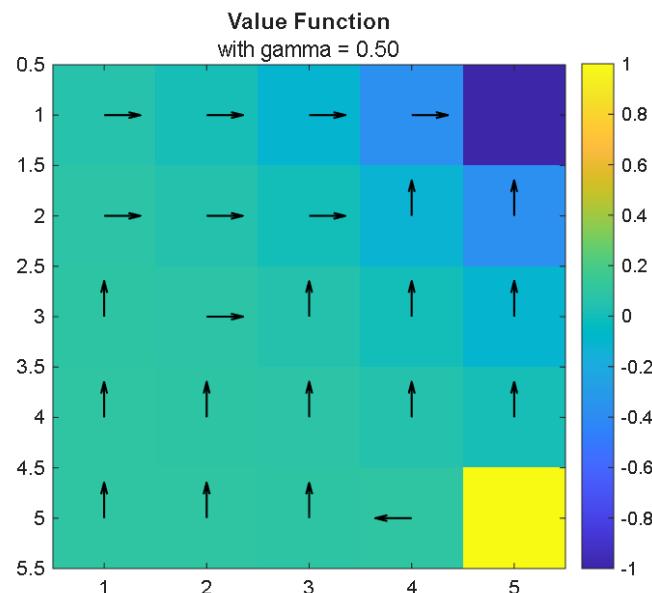
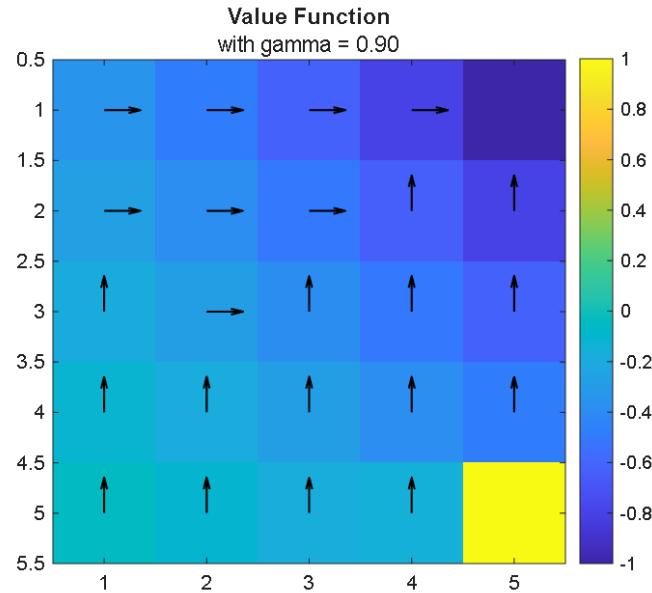
Solution:

Figure 1: Convergence of value iteration algorithm for $\gamma = 0.9$



3.2.ii)

Compute: the optimal policy. Display the optimal action at each state by showing an arrow in each cell.
Solution:



We can see as γ is decreased the policy becomes more risk averse because in the square to the left of the pit the policy moves in the opposite direction so there is no chance of slipping into it. For higher γ it takes this risk at the reward of potentially getting to the goal faster.

Code Snippet 1: Q3.2 Code

```

1 %%%%%%%%%%%%%%%%
2 % AAE590 Stochastic Control
3 % Machine Replacement Dynamic Programming
4 % Author: Travis Hastreiter
5 % Created On: 16 September, 2025
6 % Description: Solves machine replacement problem using dynamic
7 % programming.
8 % Most Recent Change: 16 September, 2025
9 %%%%%%%%%%%%%%%%
10
11 C_pit = 1;
12 C_goal = -1;
13 C_step = 0.05;
14 p = 0.2;
15
16 C = C_step * ones([5, 5]);
17
18 % Create absorbing states
19 X_abs = zeros([5, 5]);
20 % Goal
21 goal_i = 5;
22 X_abs(goal_i) = 1;
23 C(goal_i) = -1;
24 % Pit
25 pit_i = 25;
26 X_abs(pit_i) = 1;
27 C(pit_i) = 1;
28
29 % Create regular states
30 X_reg = ~X_abs;
31
32 % Get regular and absorbing state indices
33 X_reg_i = find(X_reg);
34 X_abs_i = find(X_abs);
35
36 % Create U
37 U = 1 : 4; % { Left , Right , Up , Down }
38
39 % Create tau
40 tau = zeros(5 ^ 2, 5 ^ 2, 4);
41
42 % Absorbing state
43 tau(X_abs_i(1), X_abs_i(1), :) = 1;
44 tau(X_abs_i(2), X_abs_i(2), :) = 1;
45
46 %% Left
47 % Left is wall, not at corner
48 for i = 2 : 5
49     tau(i, i, 1) = 1 - p;
50 end
51
52 % Left is wall, at corner
53 for i = [1, 5]

```

```
54     tau(i, i, 1) = 1 - p / 2;
55 end
56
57 % Left isn't wall
58 for i = 6 : 5 * 5
59     tau(i - 5, i, 1) = 1 - p;
60 end
61
62 % Up isn't wall
63 for i = reshape((2 : 5) + 5 * (0:4)', 1, [])
64     tau(i - 1, i, 1) = p / 2;
65 end
66
67 % Up or down is wall, not at left corner
68 for i = reshape([1, 5] + 5 * (1:4)', 1, [])
69     tau(i, i, 1) = p / 2;
70 end
71
72 % Down isn't wall
73 for i = reshape((1 : 4) + 5 * (0:4)', 1, [])
74     tau(i + 1, i, 1) = p / 2;
75 end
76
77 %% Right
78 % Right is wall, not at corner
79 for i = 22 : 24
80     tau(i, i, 2) = 1 - p;
81 end
82
83 % Right is wall, at corner
84 for i = [21, 25]
85     tau(i, i, 2) = 1 - p / 2;
86 end
87
88 % Right isn't wall
89 for i = 1 : 5 * 4
90     tau(i + 5, i, 2) = 1 - p;
91 end
92
93 % Up isn't wall
94 for i = reshape((2 : 5) + 5 * (0 : 4)', 1, [])
95     tau(i - 1, i, 2) = p / 2;
96 end
97
98 % Up or down is wall, not at right corner
99 for i = reshape([1, 5] + 5 * (0 : 3)', 1, [])
100    tau(i, i, 2) = p / 2;
101 end
102
103 % Down isn't wall
104 for i = reshape((1 : 4) + 5 * (0 : 4)', 1, [])
105     tau(i + 1, i, 2) = p / 2;
106 end
107
```

```
108 %% Up
109 % Up is wall, not at corner
110 for i = 1 + 5 * (1 : 3)
111     tau(i, i, 3) = 1 - p;
112 end
113
114 % Up is wall, at corner
115 for i = [1, 21]
116     tau(i, i, 3) = 1 - p / 2;
117 end
118
119 % Up isn't wall
120 for i = reshape((2 : 5) + 5 * (0 : 4)', 1, [])
121     tau(i - 1, i, 3) = 1 - p;
122 end
123
124 % Right isn't wall
125 for i = reshape((1 : 5) + 5 * (0 : 3)', 1, [])
126     tau(i + 5, i, 3) = p / 2;
127 end
128
129 % Right or left is wall, not at upper corner
130 for i = reshape((2 : 5) + 5 * [0, 4]', 1, [])
131     tau(i, i, 3) = p / 2;
132 end
133
134 % Left isn't wall
135 for i = reshape((1 : 5) + 5 * (1 : 4)', 1, [])
136     tau(i - 5, i, 3) = p / 2;
137 end
138
139 %% Down
140 % Down is wall, not at corner
141 for i = 5 + 5 * (1 : 3)
142     tau(i, i, 4) = 1 - p;
143 end
144
145 % Down is wall, at corner
146 for i = [5, 25]
147     tau(i, i, 4) = 1 - p / 2;
148 end
149
150 % Down isn't wall
151 for i = reshape((1 : 4) + 5 * (0 : 4)', 1, [])
152     tau(i + 1, i, 4) = 1 - p;
153 end
154
155 % Right isn't wall
156 for i = reshape((1 : 5) + 5 * (0 : 3)', 1, [])
157     tau(i + 5, i, 4) = p / 2;
158 end
159
160 % Right or left is wall, not at lower corner
161 for i = reshape((1 : 4) + 5 * [0, 4]', 1, [])
```

```

162     tau(i, i, 4) = p / 2;
163 end
164
165 % Left isn't wall
166 for i = reshape((1 : 5) + 5 * (1 : 4)', 1, [])
167     tau(i - 5, i, 4) = p / 2;
168 end
169
170 %% Solve
171 % Algorithm parameters
172 gamma = 0.5;
173 max_iter = 100;
174 eps = 2e-6;
175
176 [J, mu, J_cost, J_diff, converged_i] =
177     value_iteration_discounted_cost_finiteMDP(X_reg_i, X_abs_i, tau, C(:, ,
178 gamma, eps, max_iter);
converged_i
179
180 figure
181 plot(J_diff)
182 xlabel( Iteration )
183 ylabel(||V_{k+1} - V_k||_\infty )
184 title( Value Iteration Convergence )
185 grid on
186yscale( log )
187
188 J_cost(:, :, end)
189 J_grid = reshape(J(:, end), [5, 5])
190 grid_i = zeros(5, 5);
191 grid_i(X_reg_i) = mu(:, end);
192 reshape(grid_i', [5,5])
193
194 % { Left , Right , Up , Down }
195 action_x = [0, 0, -1, 1];
196 action_y = [-1, 1, 0, 0];
197
198 figure
199 image(J_grid', 'CDataMapping', 'scaled'); hold on
200 action_base_x = zeros([1, numel(X_reg_i)]);
201 action_base_y = zeros([1, numel(X_reg_i)]);
202
203 action_head_x = zeros([1, numel(X_reg_i)]);
204 action_head_y = zeros([1, numel(X_reg_i)]);
205
206 for index = 1 : numel(X_reg_i)
207     i = X_reg_i(index);
208     x_i = mod(i - 1, 5) + 1;
209     y_i = floor((i - 1) / 5) + 1;
210
211     action_base_x(index) = x_i;
212     action_base_y(index) = y_i;
213
214     if mu(index, end) ~= 0

```

```
214     action_head_x(index) = action_x(mu(index, end));
215     action_head_y(index) = action_y(mu(index, end));
216   end
217 end
218 quiver(action_base_x, action_base_y, action_head_x, action_head_y, 0.3,
219         filled, LineWidth = 1, Color = k); hold off
220 colorbar()
221 title( Value Function )
222 subtitle(sprintf( with gamma = %.2f , gamma))
223
224 %% Algorithm 6
225 function [J, mu, J_cost, J_diff, converged_i] =
226   value_iteration_discounted_cost_finiteMDP(X_reg_i, X_abs_i, tau, C,
227   gamma, eps, max_iter)
228   % Assign J(X_abs) to C_exit, arbitrary values to X_reg
229   J = C;
230
231   for iter = 1 : max_iter
232     % Step value function
233     for x_i = X_reg_i
234       J_cost(:, :, iter) = C(x_i, :)' + gamma * sum(J(:, iter) .* tau(:, x_i, :, 1);
235
236       [J(x_i, iter + 1), mu(:, iter + 1)] = min(J_cost(:, :, iter),
237           [], 2);
238       J(X_abs_i, iter + 1) = C(X_abs_i); % maintain correct
239       % absorbing state cost
240     end
241
242     % Check stopping condition
243     J_diff(iter) = norm(J(:, iter + 1) - J(:, iter), inf);
244     if J_diff(iter) <= eps
245       converged_i = iter;
246       break;
247     end
248   end
249 end
250 end
```