MECH 6326 - Optimal Control and Dynamic Programming
Homework 2

Jonas Wagner
jonas.wagner@utdallas.edu

2023, March 3$^{\text{rd}}$

Contents

Problem 1 Counting the number of policies

Consider a stochastic optimal control problem:

$$\min E\left[\sum_{k=0}^{N-1} g_k(x_k, u_k, w_k) + g_T(x_N, w_N)\right]$$

With state $x_k \in \mathcal{X} = \{1, \dots, n\}$, input $u_k \in \mathcal{U} = \{1, \dots, m\}$, and disturbance $w_k \in \mathcal{W} = \{1, \dots, p\}$. For finite state problems, the number of possible open- and closed-loop problems is finite. Determine the number for each policy type below in terms of the cardinalities of these sets. [1]

1a)

PROBLEM:

An open loop policy: $u = (u_0, \dots, u_{N-1})$ $u : \{0, \dots, N-1\} \to \mathcal{U}$ How many open-loop policies are there?

SOLUTION:

$$|\mathcal{U}|^{|\{0,\dots,N-1\}|} = m^N$$

1b)

PROBLEM:

A closed-loop policy is a sequence of functions: $\pi = (\pi_0, \dots, \pi_{N-1})$, where $\pi_k : \mathcal{X} \to \mathcal{U}$. How many closed-loop policies are there?

SOLUTION:

$$|\mathcal{U}|^{\left(|\mathcal{X}|^{|\{0,\dots,N-1\}|}\right)} = m^{\left(n^N\right)}$$

1c)

PROBLEM:

A closed-loop policy in which the disturbance is known ahead of time: $\phi = (\phi_0, \dots, \phi_{N-1})$, where $\phi_k : \mathcal{X} \times \mathcal{W} \to \mathcal{U}$. (i.e.) $u_k = \phi_k(x_k, w_k)$ How many of these closed-loop policies exist?

$$|\mathcal{U}|^{\left(|\mathcal{X}\times\mathcal{W}|^{|\{0,\dots,N-1\}|}\right)} = |\mathcal{U}|^{\left((|\mathcal{X}||\mathcal{W}|)^{|\{0,\dots,N-1\}|}\right)} = m^{\left((n\cdot p)^N\right)}$$

[1] Hint: Argue that for finite sets A and B, the number of functions from A to B is $|B|^{|A|}$.

Problem 2 A simple stochastic optimal control problem

Consider the system

$$x_{k+1} = w_k(x_k + u_k), \quad k = 0, 1, 2$$

with $u_k \in \{-1, 0, 1\}$, $w_k \in \{0, 1\}$, and initial state $x_0 = -1$. The disturbance w_k is described by

$$\begin{cases} P[w_k = 0] = 1 & |x_k + u_k| > 1 \\ P[w_k = 0] = 0.5, P[w_k = 1] = 0.5 & |x_k + u_k| \leq 1 \end{cases}$$

The cost function to be minimized is

$$u_0^2 + u_1^2 + (x_1 + 1)^2 + (x_2 - 1)^2$$

2a)

PROBLEM:

Determine the state space $\mathcal{X}_k$ for $k = 1, 2$. (i.e) based on u_k and w_k, determine the possible values of x_k.

SOLUTION:

Notation: (x_k, u_k, w_k)
For $k = 0$, we have: $x_0 = -1$, $u_0 \in \{-1, 0, 1\}$, and $w_0 \in \{0, 1\}$.

$$(x_0, u_0, w_0) \in \{(-1, -1, 0), (-1, 0, 0), (-1, 0, 1), (-1, 1, 0), (-1, 1, 1)\} \quad (1)$$

Thus, at $k = 1$, with $x_1 = w_0(x_0 + u_0)$, we have

$$x_1 \in \{(-1, -1, 0) \to x_1 = 0(-1 - 1) = 0, (-1, 0, 0) \to x_1 = 0(-1 + 0) = 0,$$
$$(-1, 0, 1) \to x_1 = 1(-1+0) = -1, (-1, 1, 0) \to x_1 = 0(-1+1) = 0, (-1, 1, 1) \to x_1 = 1(-1+1) = 0\} = \{-1, 0\} \quad (2)$$

Therefore, $x_1 \in \{-1, 0\}$, $u_1 \in \{-1, 0, 1\}$, and $w_1 \in \{0, 1\}$.

$$(x_1, u_1, w_1) \in \{(-1, -1, 0), (-1, 0, 0), (-1, 0, 1), (-1, 1, 0), (-1, 1, 1),$$
$$(0, -1, 0), (0, -1, 1), (0, 0, 0), (0, 0, 1), (0, 1, 0), (0, 1, 1)\} \quad (3)$$

Thus, at $k = 2$, with $x_2 = w_1(x_1 + u_0)$ we have

$$x_2 \in \{(-1, -1, 0) \to x_2 = 0(-1 - 1) = 0, (-1, 0, 0) \to x_2 = 0(-1 + 0) = 0,$$
$$(-1, 0, 1) \to x_2 = 1(-1 + 0) = -1, (-1, 1, 0) \to x_2 = 0(-1 + 1) = 0, (-1, 1, 1) \to x_2 = 1(-1 + 1) = 0,$$
$$(0, -1, 0) \to x_2 = 0(-1 - 1) = 0, (0, -1, 1) \to x_2 = 1(0 - 1) = -1, (0, 0, 0) \to x_2 = 0(0 + 0) = 0,$$
$$(0, 0, 1) \to x_2 = 1(0 + 0) = 0, (0, 1, 0) \to x_2 = 0(0 + 1) = 0, (0, 1, 1) \to x_2 = 1(0 + 1) = 1\} = \{-1, 0, 1\} \quad (4)$$

For the trajectory of x_k, we have $(x_0, x_1, x_2) \in \mathcal{X} \subseteq \mathcal{X}_0 \times \mathcal{X}_1 \times \mathcal{X}_2 = \{-1, 0, 1\}$. $\mathcal{X}$ specifically though is actually more restrictive and is given as follows:

$$\mathcal{X} = \{(-1, -1, -1), (-1, -1, 0), (-1, 0, -1), (-1, 0, 0), (-1, 0, 1)\}$$

2b)

PROBLEM:

Compute the optimal cost-to-go functions $J_0(x)$ and $J_1(x)$ and optimal control policy $\pi = (\pi_0, \pi_1)$ using the dynamic Programming algorithm.

PRELIMINARIES:

Definition 1. **Optimal Control Policy:** *The optimal control policy,* $\pi^* = \{\mu_0^*, \ldots, \mu_{N-1}^*\}, u_k^* = \mu_k^*(x_k)$*, is defined where* $\mu_k^*(x_k)$ *minimizes the cost function*

$$J_k(x_k) = \min_{u_k \in \mathcal{U}_k} E_{w_k}(g_k(x_k, u_k, w_k) + E[J_{k+1}(f_k(x_k, u_k, w_k))])$$

that is propagated backwards in time from $N-1$ *to* 0 *where* $J_N(x_N) = g_N(x_N)$*.*
Note: *When we are dealing with a finite state system,* $x_k = i \in \mathcal{X}_k$ *and* $x_{k+1} = j = \mathcal{X}_{k+1}$*, the problem can be simplified to minimize* $J_k(x_k)$ *as follows*

$$J_k(x_k = i) = \min_{u \in \mathcal{U}_{\|}(\S_{\|} = \rangle)} \left(g(x_k = i, u) + \sum_{x_{k+1} = j} p_{ij}(u) J_{k+1}(x_{k+1}) \right)$$

where $p_{ij}(u) = P(x_{k+1} = j \ : \ x_k = i, u_k = u)$*.*

SOLUTION:

We take $N = 2$ meaning that $J_N(x_N) = J_2(x_2) = (x_2 - 1)^2$. Thus, for each $x_2 \in \mathcal{X}_2$ we have:

$$\begin{aligned} J_2(x_2 = -1) &= (-1-1)^2 = 4 \\ J_2(x_2 = 0) &= (0-1)^2 = 1 \\ J_2(x_2 = 1) &= (1-1)^2 = 0 \end{aligned}$$

For $k = 1$,

$$J_1(x_1) = \min_{u_1 \in \mathcal{U}_1} E_{w_1}[g_1(x_1, u_1, w_1) + J_2(f_1(x_1, u_1, w_1))]$$

and since $\mathcal{X}_k$ is finite, we can solve this simplistically for $x_1 \in \mathcal{X}_1$.
The probabilities for $k = 1 \to k = 2$ are given as follows: Let $P(u_k) = \{p_{ij}(u_k)\}$,

$$P(u_1 = -1) = \begin{bmatrix} 0 & 0.5 \\ 1 & 0.5 \\ 0 & 0 \end{bmatrix}$$

$$P(u_1 = 0) = \begin{bmatrix} 0.5 & 0 \\ 0.5 & 1 \\ 0 & 0 \end{bmatrix}$$

$$P(u_1 = 0) = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0.5 \end{bmatrix}$$

For $x_1 = -1$, we have $g_1(x_1, u_1) = u_1^2 + (x_1 + 1)^2$

$$\begin{aligned} J_1(x_1 = -1) = \min_{u_1 \in \{-1,0,1\}} &\left((-1+1)^2 + u_1^2 + \sum_j p_{-1,j}(u_1) J_2(f(x_1) = j) \right) \\ &= \min_{u_1 \in \{-1,0,1\}} u_1^2 + \sum_j p_{-1,j}(u_1) J_2(x_2 = j) \ldots \quad (5) \end{aligned}$$

or equivalently,

$$\begin{pmatrix} J_1(-1) \\ J_1(0) \end{pmatrix} = \begin{pmatrix} (-1+1)^2 \\ (0+1)^2 \end{pmatrix} + \min_u \begin{pmatrix} u^2 \\ u^2 \end{pmatrix} + P(u)^T \begin{bmatrix} J_2(-1) \\ J_2(0) \\ J_2(1) \end{bmatrix} = \left(\min_u \begin{pmatrix} u^2 \\ u^2 \end{pmatrix} + P(u)^T \begin{bmatrix} 4 \\ 1 \\ 0 \end{bmatrix} \right)$$

$$= \begin{pmatrix} 0 \\ 1 \end{pmatrix} + \min \left\{ u_1 = -1 \implies \begin{pmatrix} 0 \\ 1 \end{pmatrix} + \begin{pmatrix} 1 \\ 1 \end{pmatrix} + \begin{bmatrix} 0 & 0.5 \\ 1 & 0.5 \\ 0 & 0 \end{bmatrix}^T \begin{bmatrix} 4 \\ 1 \\ 0 \end{bmatrix}, u_1 = 0 \implies \begin{pmatrix} 0 \\ 1 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \end{pmatrix} + \begin{bmatrix} 0.5 & 0 \\ 0.5 & 1 \\ 0 & 0 \end{bmatrix}^T \begin{bmatrix} 4 \\ 1 \\ 0 \end{bmatrix}, u_1 = 1 \implies \begin{pmatrix} 0 \\ 1 \end{pmatrix} + \right.$$

$$= \min \left\{ \begin{bmatrix} 2 \\ 4.5 \end{bmatrix}, \begin{bmatrix} 2.5 \\ 2 \end{bmatrix}, \begin{bmatrix} 2 \\ 2.5 \end{bmatrix} \right\} \quad (6)$$

We have

$$\begin{pmatrix} J_1(-1) \\ J_1(0) \end{pmatrix} = \begin{pmatrix} 2 \\ 2 \end{pmatrix}$$

with $\mu(-1) = 1$ and $\mu(0) = 0$
For $k = 0$ the process is simpler:

$$P_0(-1) = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$P_0(0) = \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix}$$

$$P_0(1) = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

Then we have

$$J_0 = \min_u u_0^2 + P(u_0)^T \begin{bmatrix} J_1(-1) \\ J_1(0) \end{bmatrix} = \min \left\{ (-1)^2 + \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{pmatrix} 2 \\ 2.5 \end{pmatrix}, (0)^2 + \begin{bmatrix} 0.5 & 0..5 \end{bmatrix} \begin{pmatrix} 2 \\ 2.5 \end{pmatrix}, (1)^2 + \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{pmatrix} 2 \\ 2.5 \end{pmatrix}, \right\}$$

$$= \min \left\{ (-1)^2 + \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{pmatrix} 2 \\ 2.5 \end{pmatrix}, (0)^2 + \begin{bmatrix} 0.5 & 0..5 \end{bmatrix} \begin{pmatrix} 2 \\ 2.5 \end{pmatrix}, (1)^2 + \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{pmatrix} 2 \\ 2.5 \end{pmatrix} \right\}$$

$$= min\{1 + 2.5, 0 + 2.25, 1 + 2.5\} = min\{3.5, 2.25, 3.5\} = 2.25 \quad (7)$$

Thus $u_0 = 0$.
The optimal control policy becomes: $\pi = (\pi_0, \pi_0)$ with $\pi_0 = 0$ and

$$p_1 = \begin{cases} 0 & x_1 = -1 \\ 1 & x_1 = 0 \end{cases}$$

Problem 3 Another simple stochastic optimal control problem

Consider a system with state $x \in \mathcal{X} = \{1, 2\}$, input $u \in \mathcal{U} = \{1, 2\}$, disturbance $w \in \mathcal{W} = \{1, 2\}$, update equation

$$x_{k+1} = f_k(x, u, w) = u$$

, and cost function

$$g_k(x, u, w) = \begin{cases} -1 & x = u = 1, w = 1, \\ (k-1)^2 & x = 1, u = 2 \\ 0 & \text{otherwise} \end{cases} \qquad g_N = \begin{cases} 0 & x = 2 \\ \infty & \text{otherwise} \end{cases}$$

Suppose $x_0 = 1$, $N = 4$, and w_k are i.i.d. random variables with

$$P(w_k = 1) = 0.3, \quad P(w_k = 2) = 0.7$$

3a)

PROBLEM:

Based on your results from Problem 1, how many open-loop, closed-loop state feedback, and closed-loop disturbance state feedback policies are there? This gives are rough idea of how many computations would need to be performed to compute the optimal policy by brute force.

SOLUTION:

Open-loop

$$|\mathcal{U}|^{|\{0,\ldots,N-1\}|} = m^N = (2)^(4) = 16$$

Closed-loop state feedback

$$|\mathcal{U}|^{\left(|\mathcal{X}|^{|\{0,\ldots,N-1\}|}\right)} = m^{\left(n^N\right)} = (2)^{\left(2^4\right)} = 2^16 = 65536$$

Closed-loop state and disturbance feedback

$$|\mathcal{U}|^{\left(|\mathcal{X} \times \mathcal{W}|^{|\{0,\ldots,N-1\}|}\right)} = |\mathcal{U}|^{\left((|\mathcal{X}||\mathcal{W}|)^{|\{0,\ldots,N-1\}|}\right)} = m^{\left((n \cdot p)^N\right)} = 2^{\left((2 \cdot 2)^4\right)} = 2^(4^4) = 2^256 \approx 1.15 \cdot 10^77$$

3b)

PROBLEM:

Compute the optimal policy and value functions using the dynamic programming algorithm for (i) state feedback and (ii) state and disturbance feedback.

SOLUTION:

State Feedback

Since the cost function is only ever dependent on the current state, input, and disturbance it forward/backward propagation over multiple timesteps is unnecessary.
For determining u_3, everything is unimportant outside of knowing g_4 is infinity unless $u_3 = 2$; thus, $\pi_3(\cdot) = 2$.
For $k = 3$, we determined that $u_3 = \mu_3^* = 2$

$$J_3 = \begin{cases} (3+1)^2 + 0 = 16 & x_3 = 1, u_3 = 2 \\ 0 & x_3 \neq 1, u_3 = 2 \\ \infty & u_3 \neq 2 \end{cases}$$

For $k = 2$,

$$g_2 = \begin{cases} (0.3)(-10) + (0.7)(0) + 16 = 13 & x_2 = 1, u_2 = 1 \\ (2+1)^2 + 0 = 9 & x_2 = 1, u_2 = 2 \\ 0 + 16 = 16 & x_2 \neq 1, u_2 = 1 \\ 0 + 0 & x_2 \neq 1, u_2 = 2 \end{cases}$$

thus we have $\pi_2(\cdot) = 2$ with

$$J_2(x_2 = 1) = 9, \quad J_2(x_2 = 2) = 0$$

For $k = 1$,

$$g_1 = \begin{cases} (0.3)(-10) + (0.7)(0) + 9 = 6 & x_1 = 1, u_1 = 1 \\ (1+1)^2 + 0 = 4 & x_1 = 1, u_1 = 2 \\ 0 + 9 = 9 & x_1 = 2, u_2 = 1 \\ 0 + 0 & x_2 = 2, u_2 = 2 \end{cases}$$

thus we have $pi_1(\cdot) = 2$ with

$$J_1(x_1 = 1) = 4, \quad J_2(x_1 = 2) = 0$$

For $k = 0$,

$$g_1 = \begin{cases} (0.3)(-10) + (0.7)(0) + 4 = 1 & x_0 = 1, u_0 = 1 \\ (0+1)^2 + 0 = 1 & x_0 = 1, u_0 = 2 \end{cases}$$

thus we have $pi_0(\cdot) = 1$ (or 2) with

$$J_0 = 1$$

State and Disturbance Feedback

or determining u_3, everything is unimportant outside of knowing g_4 is infinity unless $u_3 = 2$; thus, $\pi_3(\cdot) = 2$. For $k = 3$, we determined that $u_3 = \mu_3^* = 2$

$$J_3 = \begin{cases} (3+1)^2 + 0 = 16 & x_3 = 1, u_3 = 2 \\ 0 & x_3 \neq 1, u_3 = 2 \\ \infty & u_3 \neq 2 \end{cases}$$

For $k = 2$,

$$g_2 + g_3 = \begin{cases} -10 + 16 = 6 & x_2 = 1, u_2 = 1, w_2 = 1 \\ 0 + 16 = 16 & x_2 = 1, u_2 = 1, w_2 = 2 \\ (2+1)^2 + 0 = 9 & x_2 = 1, u_2 = 2 \\ 0 + 16 = 16 & x_2 = 2, u_2 = 1 \\ 0 + 0 & x_2 = 2, u_2 = 2 \end{cases}$$

thus we have $\pi_2(\cdot) = 2$ with

$$J_2(x_2 = 1) = 6, \quad J_2(x_2 = 2) = 0$$

For $k = 1$,

$$g_1 + g_2 = \begin{cases} -10 + 9 = -1 & x_1 = 1, u_1 = 1, w_1 = 1 \\ 0 + 9 = 6 & x_1 = 1, u_1 = 1, w_1 = 2 \\ (1+1)^2 + 0 = 4 & x_1 = 1, u_1 = 2 \\ 0 + 9 = 9 & x_1 = 2, u_2 = 1 \\ 0 + 0 & x_2 = 2, u_2 = 2 \end{cases}$$

thus we have $pi_1(x_1 = 1, w_1 = 1) = 1$ otherwise $\pi_1 = 2$ with

$$J_1(x_1 = 1, w_1 = 1) = -1, \quad J_1(x_1, w_1 = 2) = 6, \quad J_2(x_1 = 2) = 0$$

For $k = 0$,

$$g_0 + g_1 = \begin{cases} -10 + -1 = -11 & x_0 = 1, u_0 = 1, w_0 = 1, w_1 = 1 \\ 0 + -1 = -1 & x_0 = 1, u_0 = 1, w_0 = 2, w_1 = 1 \\ -10 + 6 = -4 & x_0 = 1, u_0 = 1, w_0 = 1, w_1 = 2 \\ 0 + 6 = 6 & x_0 = 1, u_0 = 1, w_0 = 2, w_1 = 2 \\ (0+1)^2 - 1 = 0 & x_0 = 1, u_0 = 2, w_1 = 1 \\ (0+1)^2 + 6 = 7 & x_0 = 1, u_0 = 2, w_1 = 2 \end{cases}$$

thus we have $pi_0(\cdot) = 1$.

Problem 4 Markov Chain Distributions

Consider a Markov Chain with the transition graph in Figure 1.

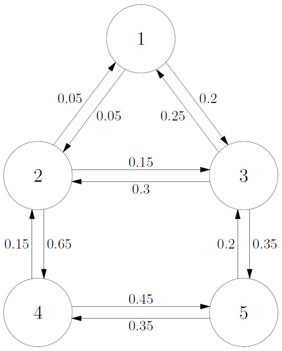

Figure 1: Markov Chain transition graph for Problem 4 (self-loops are omitted)

Suppose the initial state is $x_0 = 1$, and therefore $p_0 = \begin{bmatrix}1 & 0 & 0 & 0 & 0\end{bmatrix}$.

4a)

PROBLEM:

Use distribution propagation to compute the following quantities at $N = 50$:

- $P(x_N = 1)$
- $\frac{1}{N+1}\sum_{k=0}^{N} P(x_k = 1)$

SOLUTION:

The Markov Chain in Figure 1 is associated with the transition matrix

$$P = \begin{bmatrix} 0.75 & 0.05 & 0.25 & 0 & 0 \\ 0.05 & 0.15 & 0.3 & 0.15 & 0 \\ 0.2 & 0.15 & 0.1 & 0 & 0.2 \\ 0 & 0.65 & 0 & 0.4 & 0.35 \\ 0 & 0 & 0.35 & 0.45 & 0.45 \end{bmatrix}$$

Since the propagation of probabilities is given as

$$p_{k+1} = p_k P$$

we have

$$p_k = p_0 P^k$$

thus

$$p_N = p_0 P^N$$

which we can calculate with matlab.
The resulted probability for $P(x_N = 1)$ is the first element in p_N.
Similarly, the sum is the first element from the sum calculated as

$$\frac{1}{N+1}\sum_{k=1}^{N} p_k = \frac{1}{N+1}\sum_{k=1}^{N} p_0 P^k = \frac{1}{N+1} p_0 \sum_{k=1}^{N} P^k$$

which can also be calculated in matalab.

4b)

PROBLEM:

Then use Monte Carlo estimation to approximate these quantities. Generate plots of your estimates versus the number of sample trajectories used from 10 up to 10000 sample trajectories. Compare the simulation results with the exact values computed previously.

SOLUTION:

See attached MATLAB results.
Since there were issues publishing results, the plots are shown in Figure 2 and Figure 3.

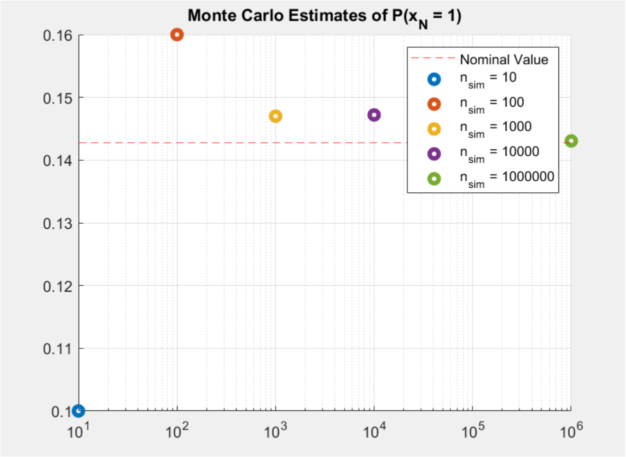

Figure 2: Monte Carlo Final Probability Estimate

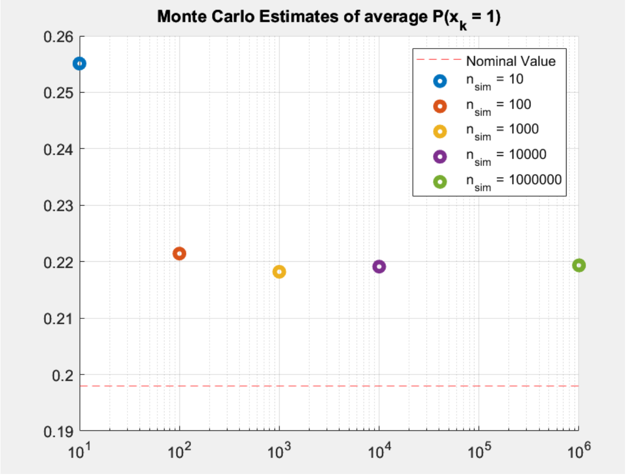

Figure 3: Monte Carlo Average Probability Estimate

MECH 6326 - HW2

Jonas Wagner

```
clear
close all
```

Problem 4

```
% Setup
P = [
    0.75 0.05 0.25 0 0
    0.05 0.15 0.3 0.15 0
    0.2 0.15 0.1 0 0.2
    0 0.65 0 0.4 0.35
    0 0 0.35 0.45 0.45
];
p_0 = [1 0 0 0 0];
```

Part 1

```
N = 50;
% P(x_N = 1)
p_N = p_0 * P^N
```

```
p_N = 1×5
    0.1428    0.1428    0.1428    0.1428    0.1428
```

```
fprintf('P(x_k = 1) = %f\n',p_N(1))
```

```
P(x_k = 1) = 0.142756
```

```
for k=1:N
    p_all(k,:) = p_0 * P^k;
end
p_avg = mean(p_all,1);
fprintf('Average P(x_k = 1) = %f\n', p_avg(1))
```

```
Average P(x_k = 1) = 0.197995
```

Part 2

```
% figure(1); hold on; grid on;
% title('Monte Carlo Estimates of P(x_N = 1)');
% legend();
% yline(p_N(1),'r--', 'DisplayName', 'Nominal Value');
% set(gca, 'xscale', 'log')
%
% figure(2); hold on; grid on;
% title('Monte Carlo Estimates of average P(x_k = 1)');
% legend();
% yline(p_avg(1),'r--', 'DisplayName', 'Nominal Value');
% set(gca, 'xscale', 'log')


N = 50;
x_0 = 1;
for n_sim = [10,100,1000,10000]
    rng(1);% set standard seed
    X = zeros(n_sim,N);
    for j = 1:n_sim
        X(j,1) = x_0;
        for i = 2:N
            X(j,i) = find(rand < cumsum(P(:,X(j,i-1))),1); % complicated... but essentially jus
        end
    end
    % P(x_N = 1)
    p_N_1 = sum(X(:,N)==1)/n_sim;
    % Average P(x_k = 1)
    p_avg_1 = sum(sum(X==1,2)/(N-1))/n_sim;

    % Display Results
    disp(['Monte Carlo with ', num2str(n_sim), ' simulations:'])
    disp(['P(x_N = 1) = ', num2str(p_N_1)])
    disp(['Average P(x_k = 1) = ', num2str(p_avg_1)])

%     % Plot Final Results
%     figure(1);
%     plot( ...
%         n_sim, p_N_1, 'o', 'LineWidth', 3, ...
%         'DisplayName', ['n_{sim} = ', num2str(n_sim)])
%
%     figure(2);
%     plot( ...
%         n_sim, p_avg_1, 'o', 'LineWidth', 3, ...
%         'DisplayName', ['n_{sim} = ', num2str(n_sim)])
end
```

```
Monte Carlo with 10 simulations:
P(x_N = 1) = 0.1
Average P(x_k = 1) = 0.2551
Monte Carlo with 100 simulations:
P(x_N = 1) = 0.16
Average P(x_k = 1) = 0.22143
Monte Carlo with 1000 simulations:
P(x_N = 1) = 0.147
Average P(x_k = 1) = 0.2182
Monte Carlo with 10000 simulations:
P(x_N = 1) = 0.1472
Average P(x_k = 1) = 0.21912
```