Theory

Q1

a) Direct Sampling

Strengths

- It works well for estimating simple probabilities. $P(\text{air}) = 0.80$
- It can directly sample to common events like leisure travel.

Weakness

- Less efficient for conditional probabilities like $P(\text{leisure} / \text{train}) = 0.400$
- Inefficient for rare events like bus travel with low stress, $P(\text{bus}/\text{lowstress}) = 0.015$

Rejection Sampling

Strengths

- Very useful for estimating conditional probabilities like $P(\text{lisure} / \text{train}) = 0.400$

$P(\text{business}/\text{air}) = 0.350$

- Can sample from difficult distributions using a simple proposal distribution.

Weakness

- High rejection rates when estimating rare events.
- Rejecting samples if proposal doesn't match target well.

Gibbs Sampling

Strengths

- Well suited for estimating joint relationships between Travel, Purpose or stress.
- Efficiently handle conditional probabilities
  $P(\text{stress}/\text{air}) = 0.60$
- Handle complex and rare events easily.

Weakness

- Converge slowly if variables highly correlated.
- It requires specifications of conditional distributions.

b) no. of people $= 100$

train travel $= 30$

$P(\text{leisure}/\text{train}) = 0.400$

expected people $= 30 \times 0.400 = 12$

c) $P(\text{air}) = 0.80$

$P(\text{business}/\text{air}) = 0.20$

$$P(\text{air} \wedge \text{business}) = P(\text{air}) \times P(\text{business}/\text{air})$$
$$= 0.80 \times 0.20$$
$$= 0.160$$

d) On increasing sample size

Accuracy

- Larger sample size reduces sampling error.
- Law of Large Numbers ensures sample mean approaches true.

Precision

- Standard error decreases proportionally as $1/\sqrt{n}$.
- More suitable estimates of rare events.

for this dataset,

rare events like bus estimation gets better.

accurate estimation of conditional probabilities.

reduced variance.

better representation of all travel modes.

Q2

a) R : reads books

A : Access Journals

C : Participates in book club

Now,

S1 : $P(R \cup A) = 0.78$

S2 : $P(A/R) = 0.40$, $P(R \cap \neg A) = 0.60$

S3 : $P(\neg A \cap \neg R) = 0.090$

S4 : $P(A/\neg R) = 0.850$

S5 : $P(C/R) = 0.320$

S6 : $P(C/\neg R) = 0.0044$

S7 : $P(C \cap A) = 0.060$

S8 : $P(C \cup A) = 0.60$

S9 : $P(A/C) = 0.40$

S10 : $P(A) = 0.50$

b) Axioms of probability they satisfy -

- non-negativity as all the $P(x) \geq 0$.
- normalization we can check using a case

$$\Sigma P(\text{outcomes}) = 1$$

- Additivity for disjoint events A and B.

$$P(A \cup B) = P(A) + P(B)$$

c) Now we calclate probability

$$P(\neg A \cap \neg R) = 0.090$$

$$P(A/R) = 0.4, \text{ so, } P(R \cap \neg A) = 0.60 \cdot P(R)$$

$$P(A \cap \neg R) = 0.85 \times P(\neg R)$$

Since we know,

$$P(R) + P(\neg R) = 1$$

$$P(A) + P(\neg A) = 1$$

$P(\neg R \cap \neg A) = 0.090$, $P(R \cup A) = 1 - (P(\neg R \cap \neg A))$

$= 0.91$

$P(R) = P(R \cup A) - P(A) + P(R \cap A)$

$= \dfrac{0.41}{0.60} = 0.683$, $P(\neg R) = 0.317$

$P(R \cap A) = P(A/R) \cdot P(R) = 0.273$

$P(R \cap \neg A) = P(\neg A/R) \cdot P(R) = 0.410$

$P(\neg R \cap A) = P(A/\neg R) \cdot P(\neg R) = 0.269$

$P(\neg R \cap \neg A) = 0.090$

$P(R, A, C) = 0.08736$, $P(R, A, \neg C) = 0.18564$

$P(R, \neg A, C) = 0.1312$, $P(R, \neg A, \neg C) = 0.2788$

$P(\neg R, A, C) = 0.0011836$, $P(\neg R, A, \neg C) = 0.267$

$P(\neg R, \neg A, C) = 0.0039$, $P(\neg R, \neg A, \neg C) = 0.089$

Joint distribution table.

R	A	C	Probability
1	1	1	0.087
1	1	0	0.185
1	0	1	0.132
1	0	0	0.278
0	1	1	0.0011
0	1	0	0.267
0	0	1	0.00039
0	0	0	0.089

d) Independence

$P(C/R,A) = P(C/R)$

$$P(C/R,A) = \frac{P(R \cap A \cap C)}{P(R \cap A)} = \frac{0.096}{0.24} = 0.40$$

$$P(C/R) = \frac{P(R \cap C)}{P(R)} = \frac{0.096 + 0.192}{0.72} = 0.40$$

Hence, C is conditionally independent of A given R.

$P(A/R, C) = P(A/R)$

$$P(A/R, C) = \frac{P(R \cap A \cap C)}{P(R \cap C)} = \frac{0.096}{0.288} = 0.333$$

$$P(A/R) = \frac{0.24}{0.72} = 0.333$$

Hence A is conditionally independent of C given R

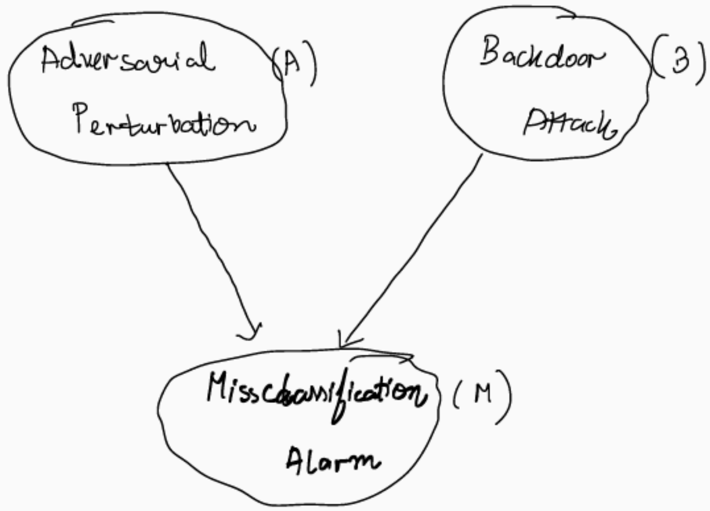
But,

$P(R/A,C) \neq P(R/A)$

R is not conditionally independent.

Q3

a)



Initial observation, A and B are independent

$P(A \cap B) = P(A) \cdot P(B)$

M depends on A and B : $P(M / A, B)$

Bayesian Update

$$P(A/M) = \frac{P(M/A) \cdot P(A)}{P(M)}$$

M is influenced A and B

$$P(M) = P(M/A,B) \cdot P(A \cap B) + P(M/A,\neg B) \cdot P(A \cap \neg B)$$
$$+ P(M/\neg A, B) \cdot P(\neg A \cap B)$$

b) Prior Probabilities

$P(A)$, $P(B)$ $\longrightarrow$ independent

Likelihood

$P(M/A)$, $P(M/B)$

Posterior

$P(A/M)$, $P(B/M)$

Joint

$$P(A \cap B) = P(A) \cdot P(B)$$

c) i) Common effect of M

both A and B can independently cause M. Observing M increases the likelihood of either A or B.

but if P(B) increases the contribution of B to M increases, reducing relative probability that A caused M.

ii) explanation discounting

if backdoor attack are more likely, the need to explain the alarm M via adversarial pertubations diminishes. This reduces P(A/M).

iii) Updated beliefs

increased P(B) shifts probability mass toward P(B/M), decreasing P(A/M).

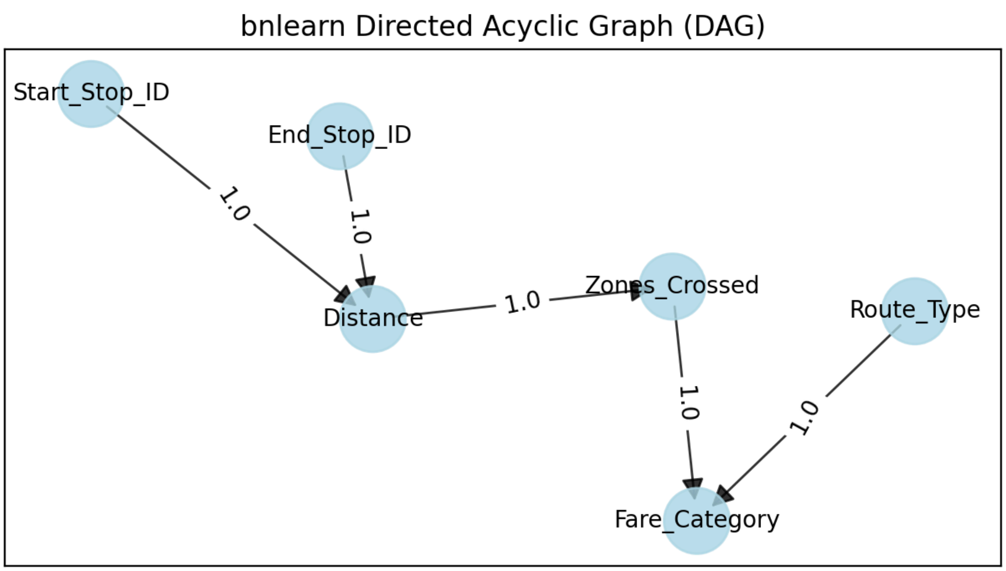
This is due to alarm being better explained by B given it's prevalence.

Bayesian Nets

Task 1)

The initial Bayesian Network for fare classification is constructed using key features like `Start_Stop_ID`, `End_Stop_ID`, `Distance`, `Zones_Crossed`, `Route_Type`, and `Fare_Category`. The network structure includes directed edges that represent logical dependencies among features, ensuring comprehensive coverage of all meaningful relationships. A visualization of the network is provided to illustrate the feature dependencies, forming the foundation for accurate parameter learning and fare classification.

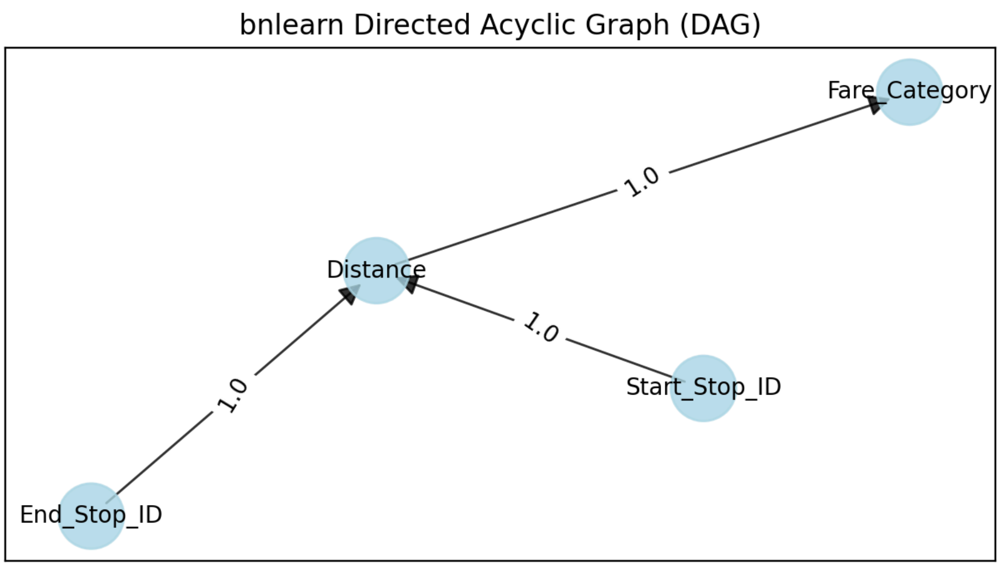
Model: Base Model
Time Taken: 26.15 seconds
Memory Usage: 3.30 MB
Accuracy on Filtered Test Cases: 100.00%



Task 2)

To enhance the performance of the Bayesian Network, pruning techniques such as **Edge Pruning** and **Conditional Probability Table (CPT) Simplification** are applied. Weak dependencies, identified through statistical measures like low mutual information, are removed, and parameters are optimized using the Hill Climbing algorithm with the Bayesian Information Criterion (BIC) to balance model fit and simplicity. These adjustments reduce the network's complexity, improving efficiency by lowering the time required to fit the model and enhancing prediction accuracy by focusing on significant relationships. The pruned Bayesian Network, visualized with fewer edges, retains only the most impactful dependencies for better interpretability and performance.

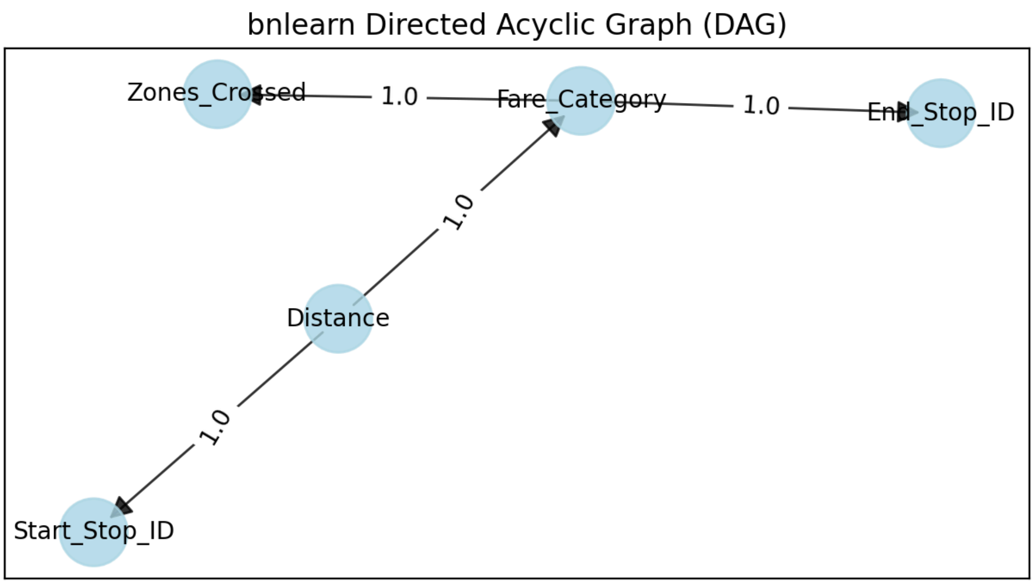
Model: Pruned Model
Time Taken: 19.93 seconds
Memory Usage: 1.51 MB
Accuracy on Filtered Test Cases: 100.00%



Task 3)

To optimize the Bayesian Network, structure refinement techniques like **Hill Climbing** are applied to learn an improved structure based on the training data. This method, combined with the Bayesian Information Criterion (BIC) scoring, ensures a balance between model complexity and fit. Additionally, parameter learning optimizes the Conditional Probability Tables (CPTs) to enhance the network's predictive accuracy. The optimized network is compared with the initial one, showing improved efficiency through reduced complexity and enhanced accuracy by better capturing significant dependencies.

Model: Optimized Model
Time Taken: 16.73 seconds
Memory Usage: 0.97 MB
Accuracy on Filtered Test Cases: 100.00%



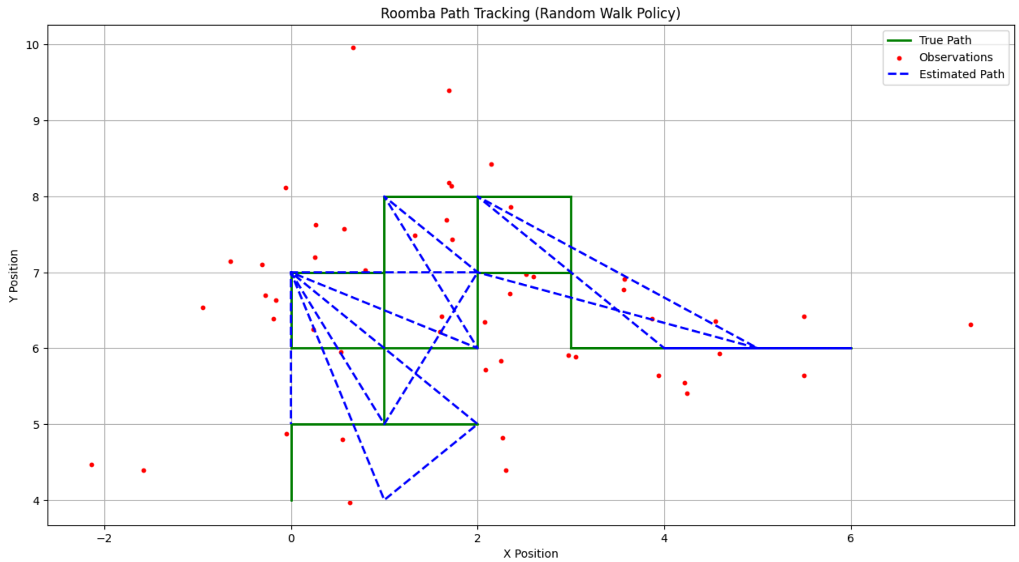
HMM

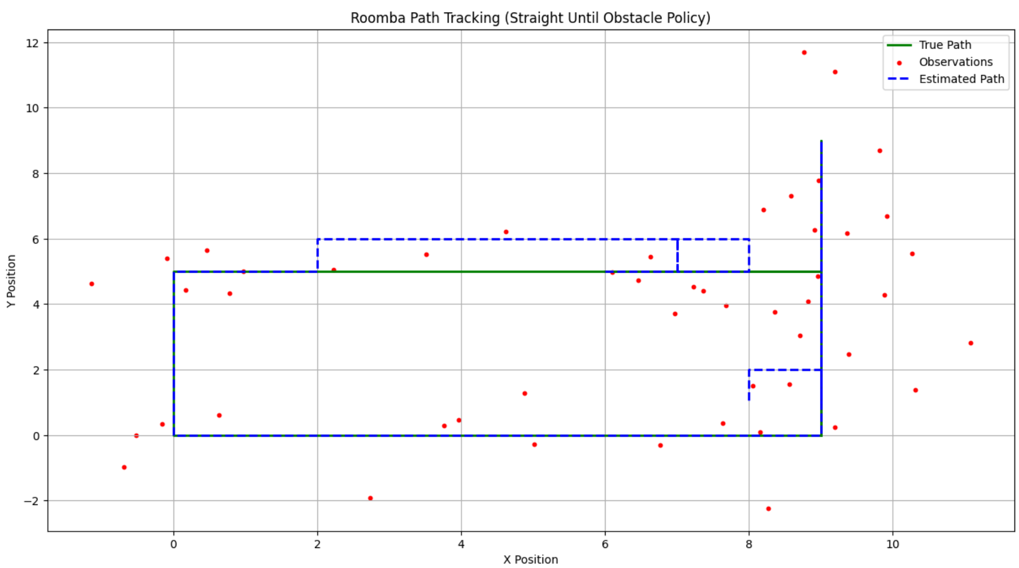
The approach employs a **Hidden Markov Model (HMM)** framework to estimate the most likely path of a Roomba robot navigating within a grid. The **emission probability** models the likelihood of observing a noisy position given a true position. This is computed using a Gaussian distribution with a specified standard deviation (σ\sigmaσ), where the mean is the true position, and deviations in observations reflect noise. The **transition probability** defines the likelihood of moving from one state (position and heading) to another, based on the robot's movement policy (e.g., `random_walk` or `straight_until_obstacle`). The transition logic ensures adherence to the policy, penalizing improbable transitions. The **Viterbi algorithm** is then applied to determine the most probable sequence of states (path) given a series of noisy observations. It iteratively computes the maximum probability path to each state at every time step, leveraging both emission and transition probabilities. Backtracking is used to reconstruct the optimal path, which is evaluated for accuracy and visualized for comparison against the true trajectory.

SEED=111

Processing policy: random_walk
Tracking accuracy for random walk policy: 34.00%

Processing policy: straight_until_obstacle
Tracking accuracy for straight until obstacle policy: 68.00%

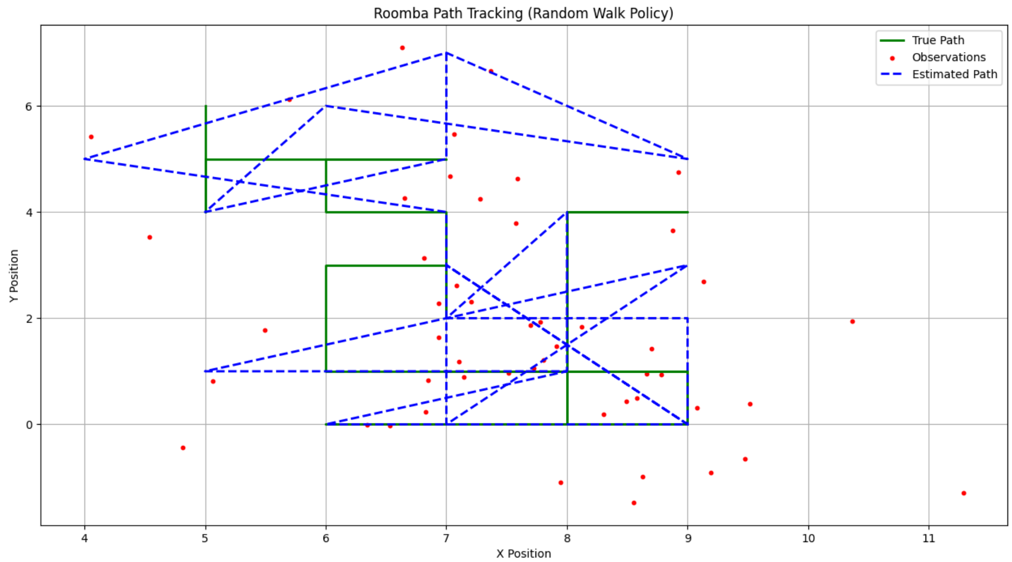


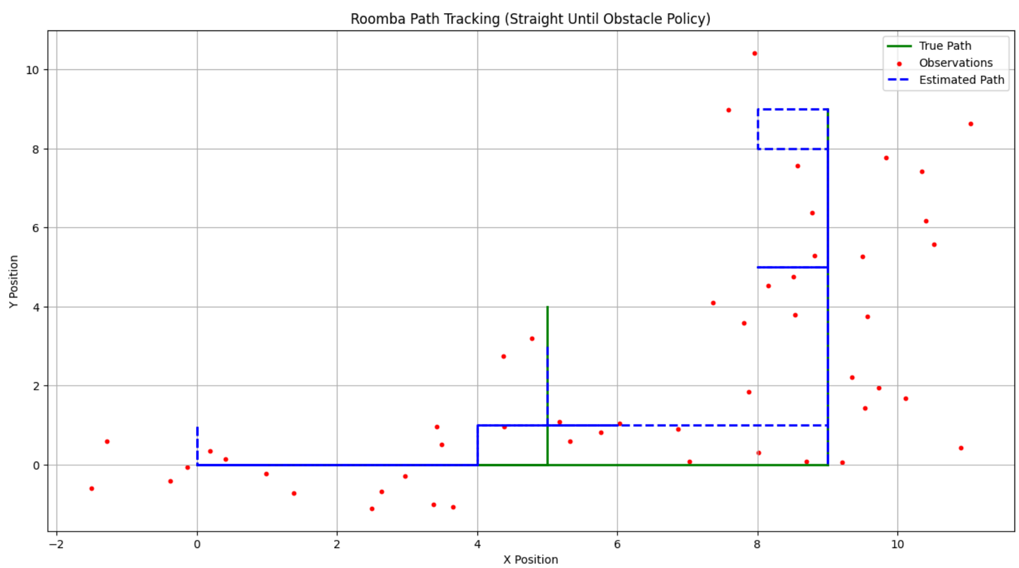


SEED=93

Processing policy: random_walk
Tracking accuracy for random walk policy: 32.00%

Processing policy: straight_until_obstacle
Tracking accuracy for straight until obstacle policy: 34.00%

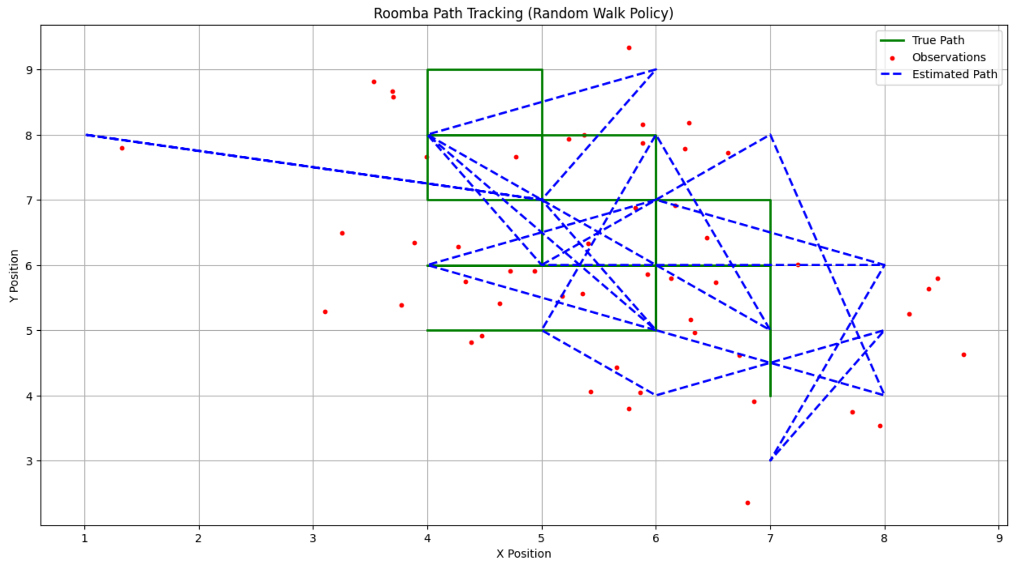


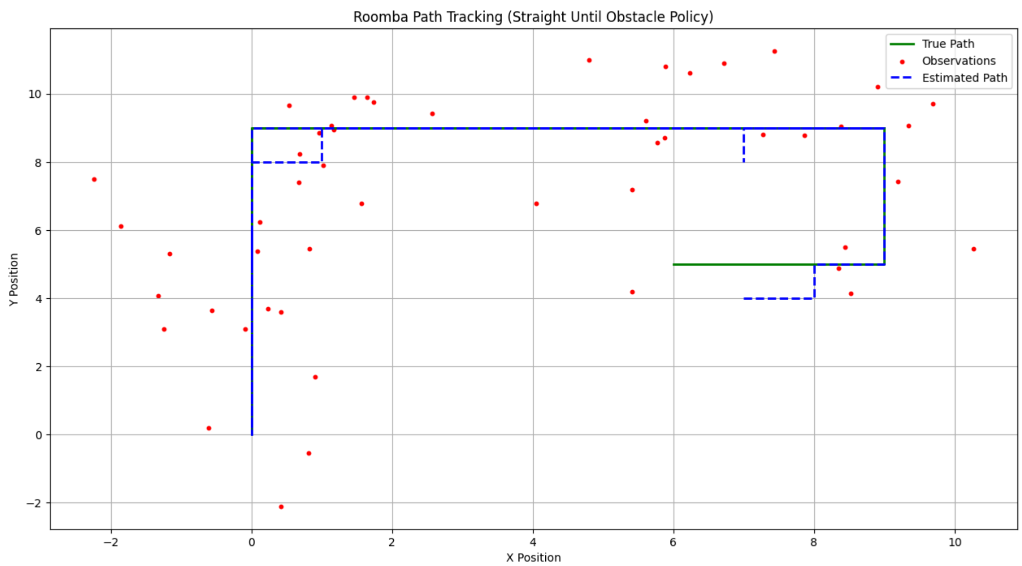


SEED=150

Processing policy: random_walk
Tracking accuracy for random walk policy: 16.00%

Processing policy: straight_until_obstacle
Tracking accuracy for straight until obstacle policy: 70.00%





Based on the results, the **`straight_until_obstacle` policy** is generally more accurate in tracking the true path of the Roomba compared to the **`random_walk` policy**. Here's an analysis:

1. **Accuracy Analysis**:
   - For **`straight_until_obstacle`**, the tracking accuracy ranges from **34% to 70%**, with higher accuracy in two out of the three seeds (68% and 70%).
   - For **`random_walk`**, the accuracy is consistently lower, ranging from **16% to 34%**, with no significant improvement across seeds.

2. **Reason for Higher Accuracy**:
    - The **`straight_until_obstacle` policy** follows a deterministic movement strategy, where the robot moves in a straight line until encountering an obstacle. This predictability reduces the variability in the robot's trajectory, making it easier for the Viterbi algorithm to infer the correct sequence of states.
    - In contrast, the **`random_walk` policy** introduces high randomness in the movement, leading to greater uncertainty in transitions. This makes it challenging for the algorithm to accurately model the transition probabilities, resulting in lower accuracy.
3. **Impact of the Policy on Transition Probabilities**:
    - The **transition probabilities** in the `straight_until_obstacle` policy are more structured, as the robot's next position is largely determined by its heading and obstacle constraints.
    - The **random_walk policy**, however, allows equal likelihood of movement in any direction, increasing the ambiguity in estimating the correct state transitions.