Bayesian Tracking and Robot Simulation on Prime Number Map Localization

1st Jaehah Shin
Engineering Science: Robotics Engineering
University of Toronto
Toronto, Canada
jaehah.shin@mail.utoronto.ca

I. Introduction

This assignment is to takle several concepts that are covered in the course: Robot Modeling, Simulating linear robotics system, and Bayesian Tracking while tackling the beatifulness of prime numbers. The blank template of assignment version for student is located here. The solution for the assignment is located here.

II. Prologue of Problem

Prime numbers, with their inherent unpredictability, are fundamental to modern cryptography—a cornerstone of secure and reliable robotic systems with sensitive applications. This project involves a human-transport robot tasked with delivering passengers to a point **50 meters beyond the last station** on its route, a location famed for hosting the *best bakery in the world*. There are 21 stations, each identified by a unique numerical address. Randomly generated numbers within the range of 2 to 11 are assigned to each address, ensuring a balanced mix of 5 prime and 5 composite numbers for fairness. These numbers are then interpreted as follows:

- **Prime Numbers:** Represented by **1** and displayed in *green*.
- **Composite Numbers:** Represented by **0** and displayed in *blue*.

The robot must fulfill two key objectives:

- **Localization:** Identify its position along the route. (Addressed in Question 1)
- **Precise Stopping:** Stop exactly **50 meters beyond the last station**. (Addressed in Question 2)

The robot begins its journey at `addresses[0] = 1` and progresses sequentially along the path.

III. Question 1: Bayesian Tracking (Localization)

In this problem, we will use the Bayesian tracking to localize the robot.

A. What is Bayesian Tracking?

Bayesian tracking estimates a robot's state using its previous state and current observations, leveraging Bayes' rule: the posterior probability of a state is proportional to the likelihood of the observation given the state, multiplied by the prior probability of the state. This method is essential for robot localization, enabling accurate state estimation. The process is mathematically expressed as follows:

$$p(x_{k+1} \mid z_{0:k}) = \sum_{x_k \in \Lambda} p(x_{k+1} \mid x_k, u_k) p(x_k \mid z_{0:k}) \tag{1}$$

The equation 1 represents the state prediction step in the bayesian tracking. This is simplified version with the *Markov's Assumption*.

$$p(x_{k+1} \mid z_{0:k+1}) = \frac{p(z_{k+1} \mid x_{k+1}) p(x_{k+1} \mid z_{0:k})}{\sum_{\xi_{k+1} \in \Lambda} p(z_{k+1} \mid \xi_{k+1}) p(\xi_{k+1} \mid z_{0:k})} \tag{2}$$

The equation 2 represents the state update step in the bayesian tracking with *Markov's Assumption*.

B. Question 1: Approach and Methodology

In Bayesian tracking, the measurement model uses station colors to estimate the robot's location, leveraging a pre-implemented and modified *Miller-Rabin with confidence* function that returns confidence which will be used in listing 1. Students must implement this model to localize the robot dynamically and apply Bayesian tracking using equations 1 and 2.

```
if confidence > 0.9:  # High confidence
    blue_meas = [0.90, 0.05, 0.05]
    green_meas = [0.05, 0.90, 0.05]
elif confidence < 0.5:  # Low confidence
    blue_meas = [0.50, 0.40, 0.10]
    green_meas = [0.40, 0.50, 0.10]
```

Listing 1. Probability for measurements

This listing 1 shows how students should implement measurement model based on confidence value from *Miller-Rabin with confidence*. This table III-B is given to implement the dynamic

$x_{k+1} \mid u_k$	-1	0	+1
$X-\chi$	0.85	0.05	0.05
X	0.10	0.90	0.10
$X+\chi$	0.05	0.05	0.85

TABLE I
State model $p(x_{k+1} \mid x_k = X, u_k)$

prediction step function of the algorithm. The randomized noise model is pre-implemented to provide students with a more realistic experience, dynamically adjusting based on randomly generated colour map at the start.

C. Results and Discussion

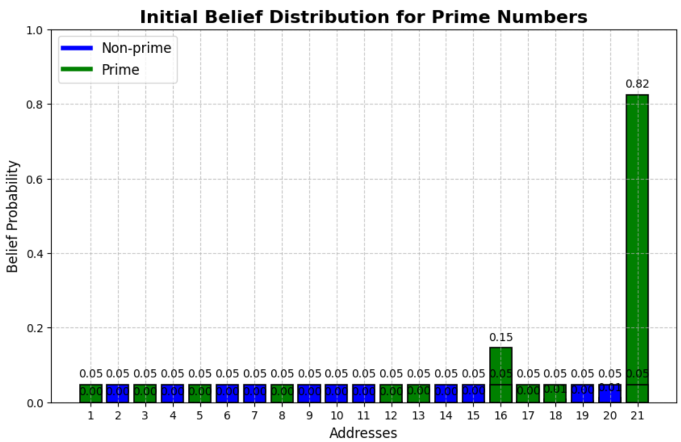

Fig. 1. Bayesian Tracking Results

Figure III-C is the result of the bayesian tracking where the given randomized input prime number was Random Numbers: [2, 6, 2, 6, 11, 9, 8, 5, 6, 9, 10, 7, 11, 10, 6, 11, 3, 5, 9, 9, 2] where corresponds to Colors: [1, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 1, 1, 0, 0, 1, 1, 1, 0, 0, 1]. (1: Prime Number, 0: Composite) Measurements are [1, -1, 1, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 1, 1, 0, 0, 1]. In this *run* of the code, there were few measurements errors, and it achieves 82% belief probability that robot is at the last station (station 21) correctly.

D. Possible Improvements

Using only two colors simplifies the problem, but consecutive repetitions of same colors (more than five) or repeated colour patterns could make localization harder, with having possibilities of localizing wrong robot itself wrong. However, the low probability of such repetitions makes additional adjustment not necessary, but also could be improved.

IV. Question 2: Robot Modeling and Simulation of Linear Robotics System

In question 2, the main goal is to simulate robot slows down and stops at 50m away from the final station to drop off the passengers. Students will be asked to solve for A, B, C, and D matrices in the state-space representation of the robot's dynamics. Then, they will be asked to solve for A_d and B_d matrices in the discrete-time state-space representation of the robot's dynamics. Finally, students will be asked to simulate the robot's dynamics and plot the robot's position and velocity over time.

A. Approach and Methodology

Students will be completing a state-space representation of the robot's dynamics, with the state vector containing the robot's position and velocity. The control input is the acceleration, and the output is the position. The robot's dynamics are modeled as a linear system, with the state-space representation given by the following equations:

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx + Du \end{aligned} \tag{3}$$

$$x = \begin{bmatrix} d \\ \dot{d} \\ D \end{bmatrix}, \quad y = \begin{bmatrix} D - d \\ \dot{d} \end{bmatrix} \tag{4}$$

where

- d Distance to the last prime-number station, changing over time.
- $\dot{d}$ Rate of change of d, representing robot's velocity.
- D Constant reference distance to the last prime-number station.

Students can find the A, B, C, and D matrices by computing the partial derivatives of the state and output equations or by inspection either by hand or code. solving for the A_d and B_d matrices can be done by using the Python library *scipy.signal* with the *cont2discrete* function.

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \quad C = \begin{bmatrix} -1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}, \quad D = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

$$A_d = \begin{bmatrix} 1 & 0.1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad B_d = \begin{bmatrix} 0.005 \\ 0.1 \\ 0 \end{bmatrix}$$

B. Results and Discussion

Figure IV-B shows the final snapshot of the simulation where the robot slows down and stops at 50m away from the final station. The robot's position and velocity are plotted over time, showing the robot's stopped at the desired location. The simulation results are consistent with the expected behavior of the robot.

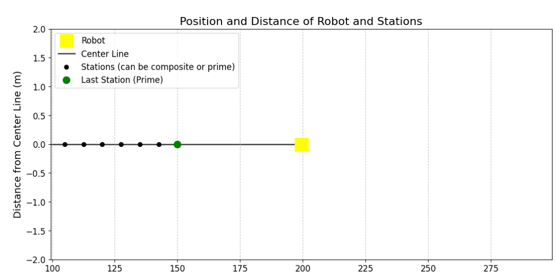

Fig. 2. Robot stopping 50 away from last station

C. Possible Improvements

For more advanced and complex project, the result from the bayesian tracking can be used as the input for the robot's dynamics simulation. This would create a more realistic and challenging scenario where the robot must navigate based on the estimated location of the prime numbers. Additionally, students can explore different control strategies and optimization techniques to improve the robot's performance and efficiency in reaching the desired location. (such as PID control, LQR, etc.)