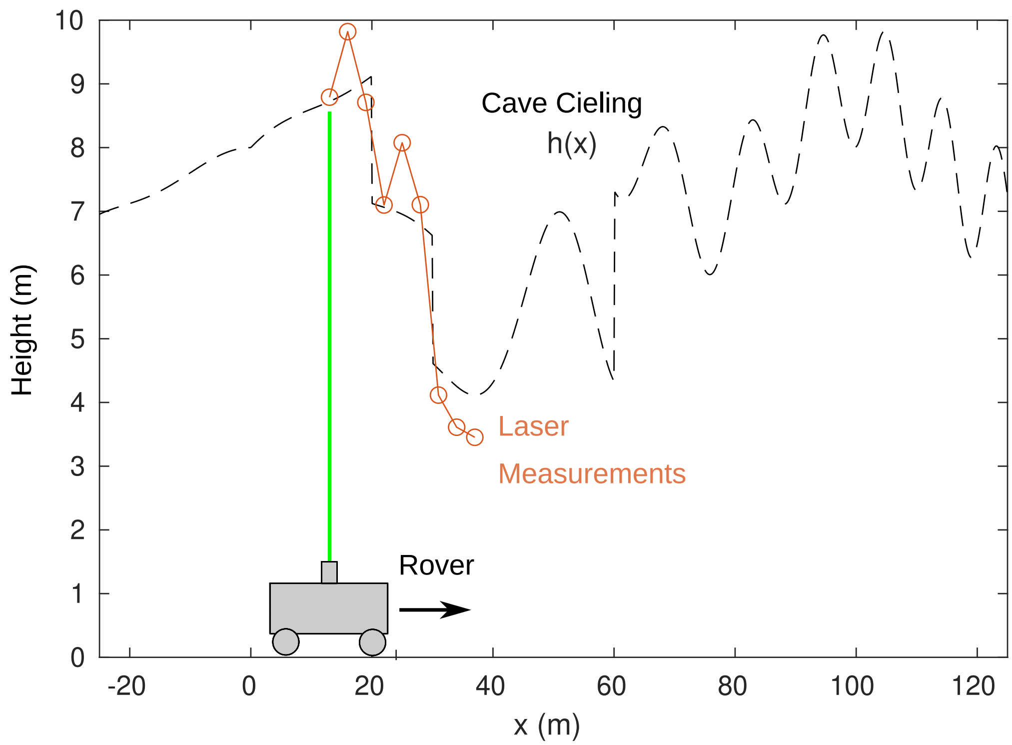
Homework 10 (due at 11pm December 1st, 2022)

1 Problem: Particle Filter

The aim of this excercise is to illustrate the use of particle filters for robot localization in a simple one dimensional model. Consider a cave-exploring robot[1] that moves in a long narrow cave as shown below.



The robot's motion is modeled as

$$x_k = x_{k-1} + u_{k-1}T \tag{1}$$

where $u_{k-1} = 3$ m/s is the robot's constant velocity input and $T = 1$ seconds is the sampling-time of the sensor. The robot has a prior knowledge of a height map $h(x)$ of the cave's ceiling and obtains noisy laser-range measurements of this height according to

$$y_k = h(x_k) + v_k \tag{2}$$

where $v_k \sim \mathcal{N}(0, \sigma_v^2)$ is an additive Gaussian measurement noise with $\sigma_v = 1$. Your goal is to design a particle filter to help the robot localize itself based on the sequence of measurements. In reality, the robot begins at $x_0 = 10$ and moves to the right; however, the initial condition is unknown to the filter. Design a particle filter according to the algorithm presented in Lecture 20 with the following parameters and modifications:

1. Use $N = 25$ particles and assume a uniform initial distribution with particles evenly spaced from 0 to 100 meters.

[1]NASA has been working on these [Link 1]and [Link 2]

2. For the roughening step (g) use the following method instead of the one mentioned in the notes: add noise $w_i \sim \mathcal{N}(0, \sigma_w^2)$ with $\sigma_w = 0.5$ to each particle after the measurement update to form the posterior.

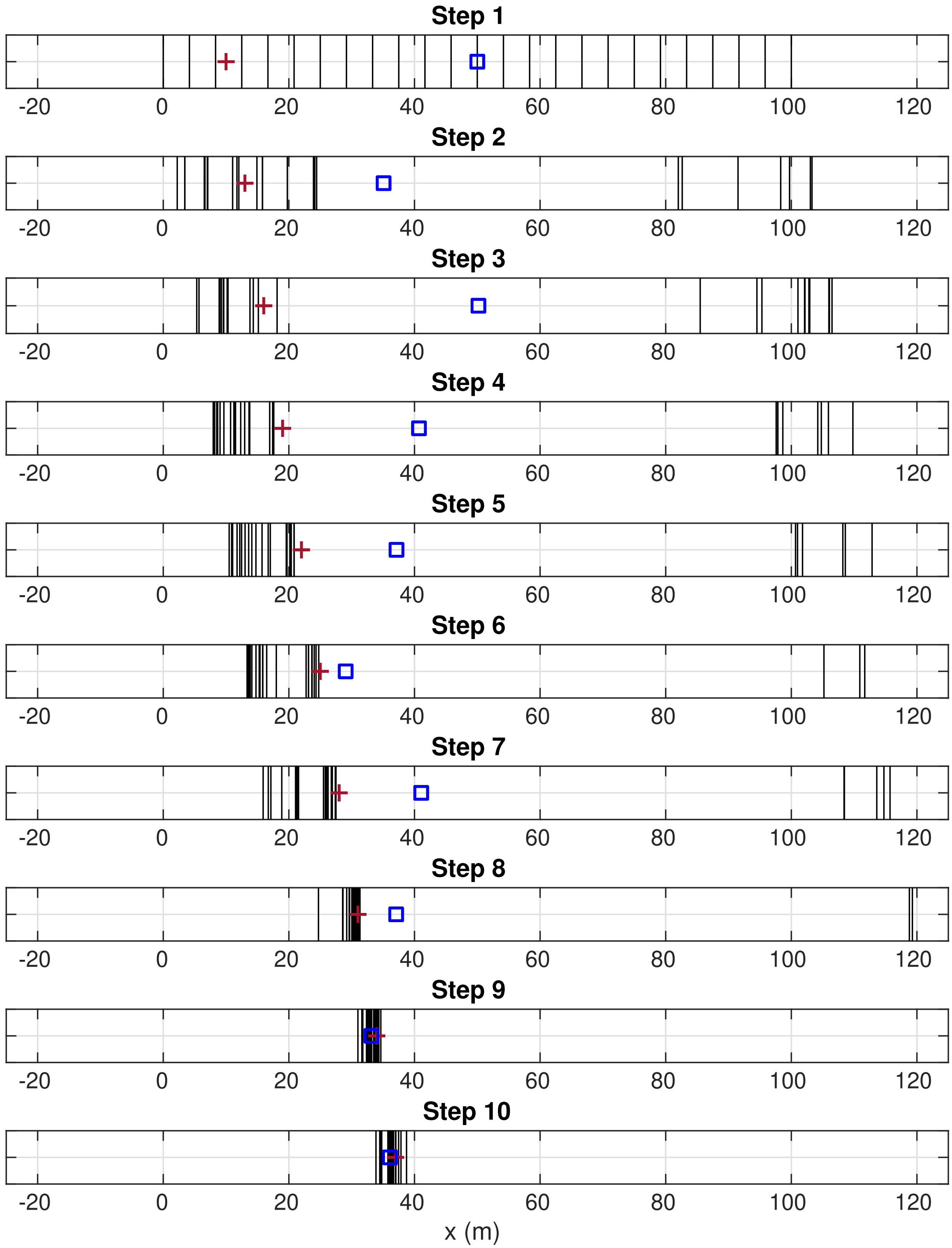
Your filter should be implemented by modifying the file `pf_process_template.m` provided on Canvas. Note that this function loads in the variable `y_noisy` containing the noisy sensor data along with `x_true` (the true position of the robot) and `h` which is an anonymous MATLAB function modeling the known height map that can be evaluated for any x. For example

```
>> h(1)
ans =
    8.0951
```

To demonstrate your filter you should submit your code along with the auto-generated plot `pf_forward.pdf` similar to the one shown on the following page labeled “forward pass”. Write a few sentences commenting on the distribution of particles and your reasoning on where they appear and why they converge to the solution when they do.

Bonus: Modify your filter to run backwards in time to create a particle-based smoother. That is, use the final set of particles at time-step $k = 10$ and reprocess the measurements from $k = 10$ down to $k = 1$ using the backward-in-time dynamics. An example reference solution for this backward pass is shown below.

Reference solution (forward pass). Particles are represented by vertical black lines. The blue square is the mean of the particles. The red plus marker is the true position of the robot.



Reference solution backward pass).