A robot's pose in 2D, denoted $\{\mathbf{p}, \phi\}$, evolves according to the following simple kinematic model:

$$\mathbf{p}_{k+1} = \mathbf{p}_k + \begin{bmatrix} \cos(\phi_k) & -\sin(\phi_k) \\ \sin(\phi_k) & \cos(\phi_k) \end{bmatrix} \begin{bmatrix} v_k \\ 0 \end{bmatrix} \delta t \tag{1}$$

$$\phi_{k+1} = \phi_k + \omega_k \delta t \tag{2}$$

In this equation v_k and ω_k are the robot's linear and rotational velocity, respectively, and δt is the sampling interval. The robot's odometry sensor provides velocity measurements at each time step:

$$v_{m_k} = v_k + w_{v_k} \tag{3}$$

$$\omega_{m_k} = \omega_k + w_{\omega_k} \tag{4}$$

where w_{v_k} and w_{ω_k} represent the measurement noise.

The robot has access to one of the following three types of measurements:

- GPS-like absolute position measurements:

$$\mathbf{z}_{GPS} = {}^{G}\mathbf{p} + \mathbf{n}_{GPS} \tag{5}$$

- Range measurements to a landmark with known position:

$$z_r = ||\mathbf{p}_L - \mathbf{p}|| + n_r \tag{6}$$

- Bearing measurements to a landmark with known position:

$$z_\theta = \sphericalangle({}^{R}\mathbf{p}_L) + n_\theta \tag{7}$$

  where ${}^{R}\mathbf{p}_L$ is the position of the landmark with respect to the robot.

Implement a particle filter (Algorithm 3) using the importance density (62). Start by implementing the GPS measurement update, which is the easiest one, and then proceed to the other two measurement functions. Note that you can have more than one sensor being used at each time step (e.g., range and bearing), by simply including two updates at each iteration of the filter.

For simplicity, the main file and the real-world simulation is provided (the code structure is similar to the one used in EE 245). You should experiment with different values for the sample number and the minimum effective particle number.