Two-wheeled Robot Beacon Based Localization for Gardening: an exercise in estimation theory

Kobe Prior
EENG 519: Estimation Theory and Kalman Filtering
Colorado School of Mines

I. Introduction

In an educational effort to deepen my understanding of system theory, estimation theory and Kalman Filtering, I will create and evaluate a system with a reasonable degree of complexity. This paper gives a thorough description of a garden with a watering robot that uses 4 beacons to localize itself and respond accurately to desired user input watering location requests.

II. System Description

Figure 1 gives a overview of the system environment. A two-wheeled robot is located on a perforated but rigid acrylic plate so that water can flow from a spigot located underneath the robot onto plants in a terrarium below. The robot is assumed to be tethered to a water source to achieve this. On the plane where the robot exists there is a home location (x_i, y_i) where the robot homes to after completing a watering session. The boundary of the system environment is defined by the range of x and y satisfying equation 1 below.

$$\mathcal{B} = \left\{(x, y) \in R^2 \;\middle|\; x_i \leq x \leq x_f,\ y_i \leq y \leq y_f\right\} \tag{1}$$

There are 4 beacons placed at the midpoint of each vertex on the boundary. These are beacons operating in the 2.1 to 2.4 GHz range each transmitting continuous waves at distinct frequencies separated by 100 MHz. $f_{B_1} = 2.1$ GHz, $f_{B_2} = 2.2$ GHz, $f_{B_3} = 2.3$ GHz, $f_{B_4} = 2.4$ GHz. The robot hosts a software defined radio that can detect the power of signals at each distinct carrier frequency and uses that information to compute the range to each beacon. The user will be able to select a "wave point" (x_0, y_0) for their watering best friend to drive to. But in order to control the robot estimation of the current position of the robot is needed to generated an error function for the controller supplying voltages, for example, to the motors of the two-wheeled robot.

III. System State Space Model

At any time the robots velocity in the x and y directions can be related to the amount of velocity of each wheel v_r for the right and v_l for the left and the current heading ϕ. Then ϕ, measured counterclockwise is 0° when the robot is parallel to the x axis and is $= 90^\circ$ when parallel to the y axis, is determined by the the difference in velocity from the right wheel and the left wheel divided by the wheel base (center to center distance of wheels) denoted b. Figure 2 illustrates an arbitrary state ϕ and b and offers shows the axis for velocity. Equations 2, 3, and 4 summarize these relationships.

$$\dot{x} = \cos\phi(v_l + v_r)/2 \tag{2}$$

$$\dot{y} = \sin\phi(v_l + v_r)/2 \tag{3}$$

$$\dot{\phi} = (v_r - v_l)/b \tag{4}$$

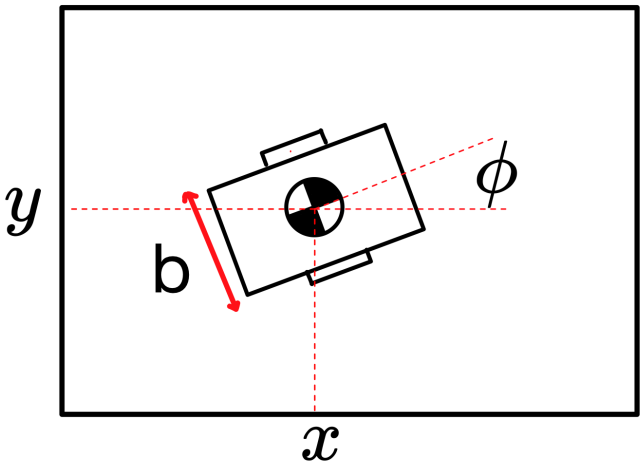

Fig. 2. Two-wheeled robot position diagram.

This can be written in as a state space representation

$$\dot{\mathbf{x}} = \begin{bmatrix}\dot{x}\\ \dot{y}\\ \dot{\phi}\end{bmatrix} = \begin{bmatrix}\frac{1}{2}\cos\phi & \frac{1}{2}\cos\phi\\ \frac{1}{2}\sin\phi & \frac{1}{2}\sin\phi\\ -\frac{1}{b} & \frac{1}{b}\end{bmatrix}\begin{bmatrix}v_l\\ v_r\end{bmatrix} \tag{5}$$

$$\mathbf{x} = \begin{bmatrix}x\\ y\\ \phi\end{bmatrix} \tag{6}$$

IV. Estimation Scenario

Without any sensors the robot is blind and impossible to control. The measurements that will be taken from the robot consists of power measurements from the four beacons placed at the boundary. The range to each beacon can be described by the following equations where R_i represents the ith beacon (beacons are labeled in Figure 1).

$$R_i = \sqrt{\left(\frac{x_f + x_i}{2} - x\right)^2 + (y_i - y)^2} \tag{7}$$

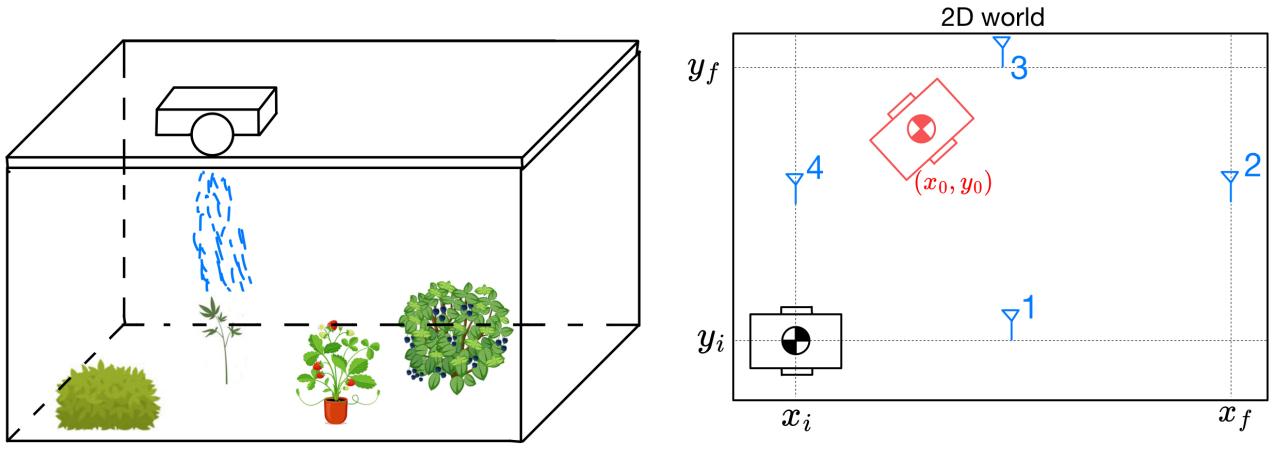

Fig. 1. Localization Based Gardening Robot System Environment.

These ranges are not directly available from the measurements. However we can relate range using the Friis Transmission equation with some simplifications. If we assume that the antennas are all isotropic and lossless the ratio of power received to power transmitted is equal to free space path loss.

$$\frac{P_{r_k}}{P_t} = \frac{1}{R^2}\left(\frac{\lambda_k}{4\pi}\right)^2 \tag{8}$$

Where P_{r_k} represents the power received on the kth carrier from the kth beacon and $\lambda_k = \frac{c}{f_k}$ is proportional the frequency associated with the kth beacon. Then if P_t is held constant at 0dBm, for example, we can create a constant $\Psi_k = P_t\left(\frac{\lambda_k}{4\pi}\right)^2$ and express our measurements in terms of ranges.

$$P_{r_k} = \frac{K_k}{R_k^2} \tag{9}$$

Replacing R_k in this expression from equation 7 gives a non linear measurement model with additive white Guassian sensor noise.

$$y_j = \frac{K_j}{(x_j - x)^2 + (y_j - y)^2} + v_k \tag{10}$$

where, (x_j, y_j) is the location of the jth beacon.

V. SIMULATION OF ESTIMATION SCENARIO

The purpose of this assignment is to explore estimation theory not necessarily control design so the simulation scenario will consist of the robot traversing a straight path from (x_i, y_i) to (x_f, y_f) illustrated in Figure 4 below. First the robot must rotate to the desired angle defined by equation 11.

$$\phi_{desired} = \tan^{-1}\left(\frac{y_f}{x_f}\right) \tag{11}$$

Rotating in place involves applying an input such that $v_l = -v_r$ so that $\dot{x} = \dot{y} = 0$ and $\dot{\phi} = \frac{2v_r}{b}$. Then we specify a time to rotate t_{rotate} seconds and solve for the required velocity to apply to achieve that rotation shown in equation 12 below.

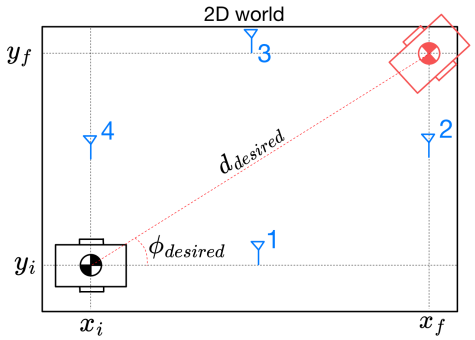

Fig. 3. Specific Measurement Straight Path Scenario.

$$v_{rotate} = v_r = -v_l = \frac{(b)(\phi_{desired})(t_{rotate})}{2} \tag{12}$$

Next we can solve for the input that needs to be applied to travel straight a distance $d_{desired} = \sqrt{x_f^2 + y_f^2}$ by setting $v_l = v_r$ so that the final destination is reached in some time $t_{forward}$ seconds. The expression to solve for the velocity required is shown in equation 13 below.

$$v_{forward} = v_r = v_l = \frac{d_{desired}}{t_{forward}} \tag{13}$$

This creates a piecewise function for the inputs through time

$$\mathbf{u}(t) = \begin{cases} \begin{bmatrix} -v_{rotate} \\ v_{rotate} \end{bmatrix}, & 0 \le t < t_{\text{rotate}} \\ \begin{bmatrix} v_{forward} \\ v_{forward} \end{bmatrix}, & t_{\text{rotate}} \le t \le t_{\text{forward}} \end{cases} \tag{14}$$

Now since the platform that the robot drives on is perforated as illustrated in Figure 5 where each hole can cause a velocity perturbation depending on the cross-section that is bisected

as in Figure 6. We can create a correlated sinusoidal process driven by iid noise with variable amplitude.

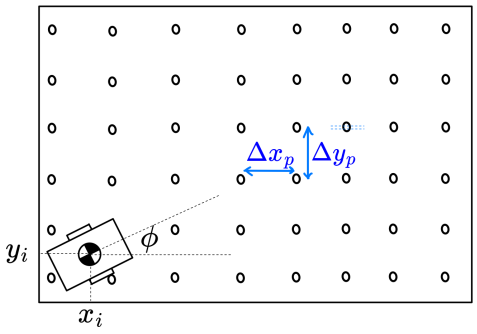

Fig. 4. Description of perforation on driving surface.

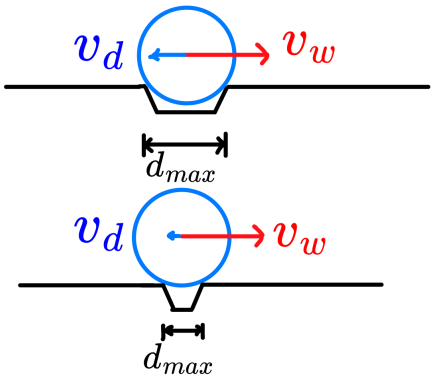

Fig. 5. Side view of wheel over different cross sections of perforations.

For sufficiently small hole spacing Δx_p and $\Delta y_p = L \ll b$ the disturbance an be modeled as a sinusoid with random amplitude A and fixed frequency approximated to be $\tilde{\omega}$

$$D = A\cos(\tilde{\omega}) \tag{15}$$

The frequency of velocity perturbation is determined by the wheel velocity ($v_{forward}$), heading direction ($\phi_{desired}$ the spacing between holes ($L = \frac{b}{20}$).

$$\tilde{\omega} = \frac{2\pi v}{L}[|\cos\phi_{desired}| + |\sin\phi_{desired}|] \approx 742.1\frac{rad}{s}$$

We model the disturbance as acting equally on each wheel velocity:

$$\tilde{\mathbf{u}}_k = \begin{bmatrix} v_{l,k} + d_k^l \\ v_{r,k} + d_k^r \end{bmatrix} \tag{16}$$

The disturbance is modeled as a correlated sinusoidal process:

$$\mathbf{d}_{k+1} = \alpha\Phi_d(\phi_k)\mathbf{d}_k + \Gamma_d n_k \tag{17}$$

where

$$\Phi_d = \begin{bmatrix} \cos(\tilde{\omega}) & -\sin(\tilde{\omega}) \\ \sin(\tilde{\omega}) & \cos(\tilde{\omega}) \end{bmatrix}, \quad \Gamma_d = \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix} \tag{18}$$

$$n_k \sim \mathcal{N}(0, \sigma_n^2)$$

$$\alpha = 0.9$$

The augmented state is defined as:

$$\tilde{\mathbf{x}}_k = \begin{bmatrix} x_k & y_k & \phi_k & d_k^l & d_k^r \end{bmatrix}^\top \tag{19}$$

The augmented nonlinear system becomes:

$$\tilde{\mathbf{x}}_{k+1} = \begin{bmatrix} f(\mathbf{x}_k, \mathbf{u}_k + \mathbf{d}_k) \\ \alpha\Phi_d(\phi_k)\mathbf{d}_k \end{bmatrix} + \begin{bmatrix} 0 \\ \Gamma_d \end{bmatrix} n_k \tag{20}$$

The power measurements will be generated at an arbitrary sample rate T_s and will be corrupted by noise distributed $w_i \sim \mathcal{N}(0, R)$. R is chosen to be a matrix representing independent measurements with equal variance. $R = \sigma^2 I$ Which is a fair assumption under the precedent that the beacons are emitting 0 bandwidth tones at the exact frequencies they are assigned. This means that as you get further away the noise has a larger effect because the measured power drops $\propto 1/R^2$. The variance σ^2 is the same as the noise power which is assume to be -100 dBm. The deterministic input for the left and right wheel velocities is illustrated in Figure 7.

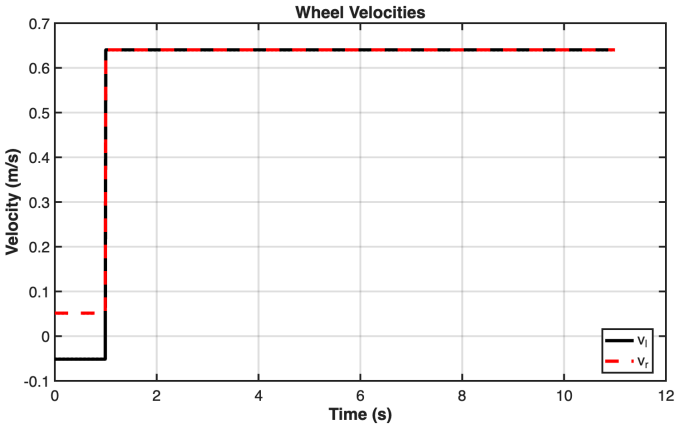

Fig. 6. Wheel Velocity (noiseless) Vs Time.

The sinusoidal disturbance is created separately and added to the input. Figures 8 illustrates the velocity disturbance for the left and right wheels versus time for input disturbances driven by n_k with $\sigma_n^2 = 0.01$.

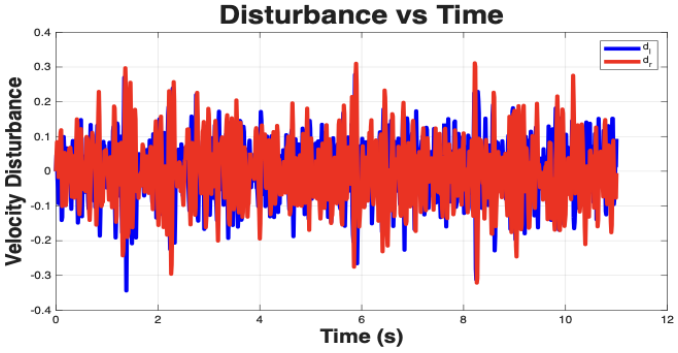

Fig. 7. Velocity Disturbance for Left and Right Wheels($\sigma_n^2 = 0.01$).

The x and y position of the Robot plotted versus time wtih the disturbance applied is shown in Figure 9.

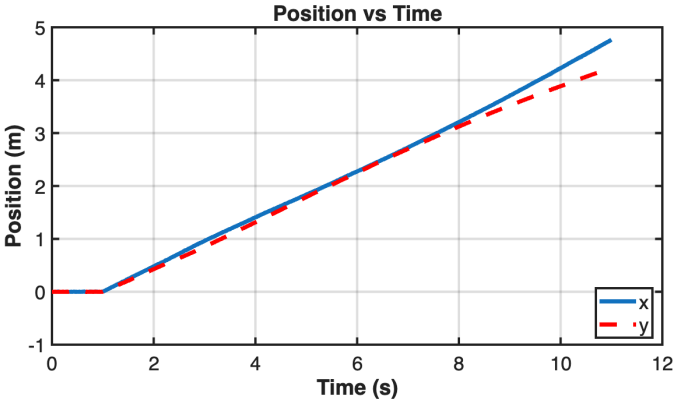

Fig. 8. Position Vs Time With Input Disturbance

Figure 10 below shows measurements at each beacon labeled in Figure 1 corrupted by additive white gaussian noise with noise power equal to -100 dBm (or equivalently $\sigma^2 = 10^{-13}$ W).

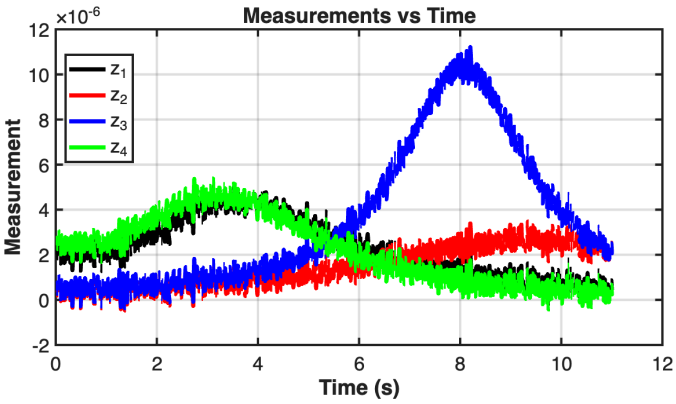

Fig. 9. Power Measurements Corrupted by Noise Captured at Each Beacon.

Later we will discuss how increasing transmitting power from the beacons will lower the overall effect of this noise effectively lowering the error covariance of the estimates that will be made.The transmitting power is set to 20 dBm which could be either interpreted as 100 mW of power radiated by an isotropic radiator or using a higher gain antenna with lower output power.

VI. EXTENDED KALMAN FILTER IMPLEMENTATION

The Extended Kalman Filter (EKF) addresses the nonlinear estimation problem by linearizing both the dynamics and measurement model at each timestep about the current state estimate. The augmented state vector incorporating pose and wheel velocity disturbances is:

$$\tilde{x}_k = \begin{bmatrix} x_k & y_k & \phi_k & d_k^l & d_k^r \end{bmatrix}^\top$$

initialized at zero with small initial covariance $P_0 = \epsilon I_5$, reflecting high confidence in the known starting position. The process and measurement noise covariances are:

$$Q = \begin{bmatrix} 0 \\ \Gamma_d \end{bmatrix} \sigma_n^2 \begin{bmatrix} 0 & \Gamma_d^\top \end{bmatrix}, \quad R = \sigma_v^2 I_4$$

Measurement Update

The nonlinear measurement function maps the robot position to received beacon power:

$$g(\hat{x}_k^{(-)}) = \begin{bmatrix} \frac{\Psi_1}{(x_1-x)^2+(y_1-y)^2} \\ \vdots \\ \frac{\Psi_4}{(x_4-x)^2+(y_4-y)^2} \end{bmatrix}$$

The observation Jacobian C_k linearizes g about the current prior estimate. Since beacon power depends only on x and y, the last three columns corresponding to ϕ, d_l, and d_r are zero:

$$C_k = \begin{bmatrix} \frac{2\Psi_1(x_1-x)}{[(x_1-x)^2+(y_1-y)^2]^2} & \frac{2\Psi_1(y_1-y)}{[(x_1-x)^2+(y_1-y)^2]^2} & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{2\Psi_4(x_4-x)}{[(x_4-x)^2+(y_4-y)^2]^2} & \frac{2\Psi_4(y_4-y)}{[(x_4-x)^2+(y_4-y)^2]^2} & 0 & 0 & 0 \end{bmatrix}$$

The Kalman gain, state estimate, and covariance are then updated:

$$\bar{K}_k = P_k^{(-)} C_k^\top \left(C_k P_k^{(-)} C_k^\top + R \right)^{-1}$$

$$\hat{x}_k^{(+)} = \hat{x}_k^{(-)} + \bar{K}_k \left(y_k - g(\hat{x}_k^{(-)}) \right)$$

$$P_k^{(+)} = (I - \bar{K}_k C_k) P_k^{(-)}$$

Time Update

The augmented state is propagated forward using the Euler-discretized dynamics. The full nonlinear propagation is written explicitly as:

$$\hat{x}_{k+1}^{(-)} = \begin{bmatrix} x_k + \frac{T_s}{2} \cos\phi_k (v_{l,k} + d_{l,k} + v_{r,k} + d_{r,k}) \\ y_k + \frac{T_s}{2} \sin\phi_k (v_{l,k} + d_{l,k} + v_{r,k} + d_{r,k}) \\ \phi_k + \frac{T_s}{b} (-v_{l,k} - d_{l,k} + v_{r,k} + d_{r,k}) \\ \alpha(\cos\tilde{\omega}\, d_{l,k} - \sin\tilde{\omega}\, d_{r,k}) \\ \alpha(\sin\tilde{\omega}\, d_{l,k} + \cos\tilde{\omega}\, d_{r,k}) \end{bmatrix}$$

where the heading is wrapped to $[-\pi, \pi]$ after propagation to prevent unbounded drift. The linearized transition matrix is assembled from the Jacobians of f with respect to the pose states and disturbance states:

$$\nabla_x f = \begin{bmatrix} 1 & 0 & -\frac{T_s}{2} \sin\phi (v_l + d_l + v_r + d_r) \\ 0 & 1 & \frac{T_s}{2} \cos\phi (v_l + d_l + v_r + d_r) \\ 0 & 0 & 1 \end{bmatrix}$$

$$\nabla_d f = \begin{bmatrix} \frac{T_s}{2}\cos\phi & \frac{T_s}{2}\cos\phi \\ \frac{T_s}{2}\sin\phi & \frac{T_s}{2}\sin\phi \\ -\frac{T_s}{b} & \frac{T_s}{b} \end{bmatrix}$$

These are assembled into the full 5×5 transition matrix:

$$\Phi_k = \begin{bmatrix} \nabla_x f & \nabla_d f \\ 0_{2\times 3} & \alpha\Phi_d \end{bmatrix}$$

and the covariance is propagated as:

$$P_{k+1}^{(-)} = \Phi_k P_k^{(+)} \Phi_k^\top + Q$$

Results

For correct initial conditions the EKF tracks x and y accurately throughout the trajectory. Heading ϕ is well estimated during straight-line travel but shows elevated uncertainty during rotation, where the robot is not translating and beacon measurements provide limited position information. The disturbance states d_l and d_r are estimated near zero, which is physically consistent since the disturbances are zero-mean and the last three columns of C_k are zero, meaning disturbances have no direct measurement observability.

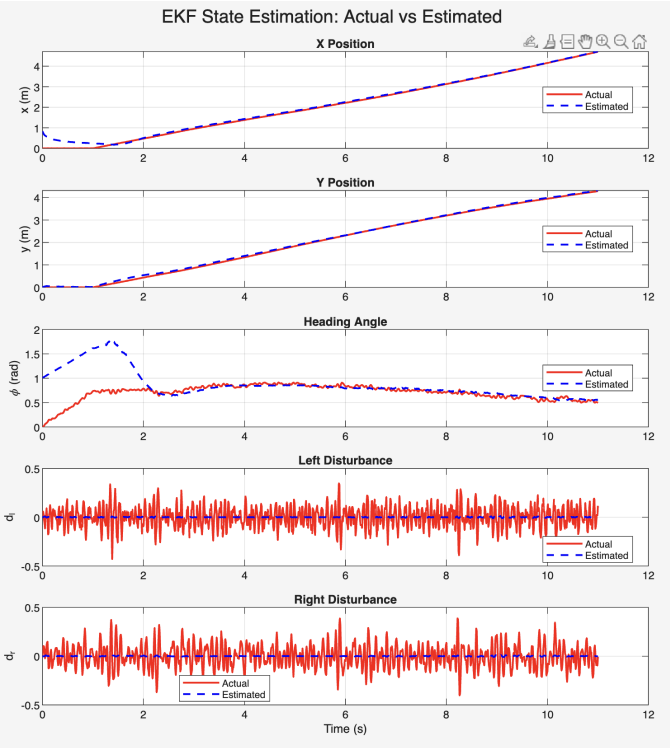

Fig. 10. Estimated State Vs Actual for Wrong Initial X Position and Heading ϕ

Initial covariance tuning proved critical to filter performance. A large P_0 inflates the Kalman gain on the first update, driving large state corrections before the filter has sufficient measurement information to settle. Reducing P_0 to reflect confidence in the known starting position significantly improved convergence when initial state guesses were incorrect, as shown above.

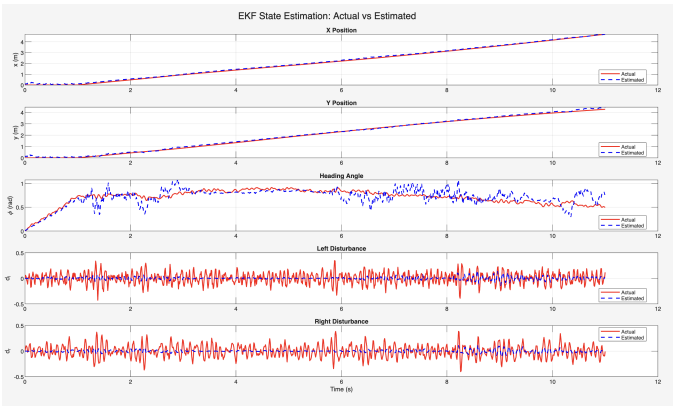

Fig. 11. State Estimation Vs Actual For Poor SNR.

SNR sensitivity analysis showed that degrading measurement quality primarily increased heading uncertainty while x and y estimation remained robust, as the deterministic input continues to aid the time update regardless of measurement noise level.

VII. MONTE CARLO KALMAN FILTER IMPLEMENTATION

The Monte Carlo Kalman Filter (MCKF) completely avoids Jacobian linearization, replacing it with sample-based statistics computed from N_p particles drawn at each timestep. The same augmented state and noise covariances are used as in the EKF.

Measurement Update

1) Draw N_p state particles from the current prior:

$$x_k^{(i)} \sim \mathcal{N}(\hat{x}_k^{(-)}, P_k^{(-)}), \quad i = 1, \ldots, N_p$$

2) Draw N_p measurement noise particles:

$$v_k^{(i)} \sim \mathcal{N}(0, R)$$

3) Evaluate the nonlinear measurement function for each particle:

$$y_k^{(i)} = g(x_k^{(i)}) + v_k^{(i)} = \begin{bmatrix} \frac{\Psi_1}{(x_1 - x^{(i)})^2 + (y_1 - y^{(i)})^2} \\ \vdots \\ \frac{\Psi_4}{(x_4 - x^{(i)})^2 + (y_4 - y^{(i)})^2} \end{bmatrix} + v_k^{(i)}$$

No Jacobian is required — the nonlinearity is captured exactly through sampling.

4) Compute sample statistics:

$$\bar{y} = \frac{1}{N_p} \sum_{i=1}^{N_p} y_k^{(i)}$$

$$P_{xy} = \frac{1}{N_p} \sum_{i=1}^{N_p} (x_k^{(i)} - \hat{x}_k^{(-)})(y_k^{(i)} - \bar{y})^\top$$

$$P_y = \frac{1}{N_p} \sum_{i=1}^{N_p} (y_k^{(i)} - \bar{y})(y_k^{(i)} - \bar{y})^\top$$

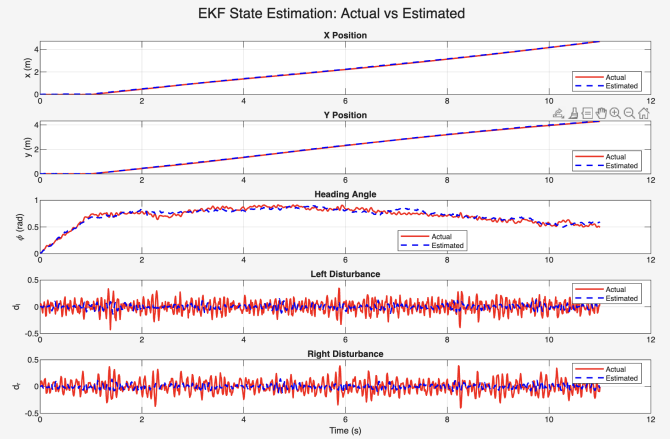

Fig. 12. $N_p = 200$, Correct IC's Estimated State Vs Actual

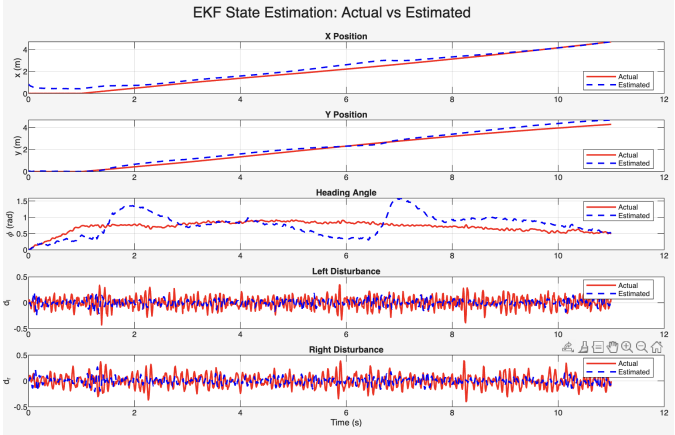

Fig. 13. $N_p = 200$,Incorrect IC's Estimated State Vs Actual

5) Compute Kalman gain and perform update:

$$K = P_{xy}P_y^{-1}$$

$$\hat{x}_k^{(+)} = \hat{x}_k^{(-)} + K(y_k - \bar{y}), \quad P_k^{(+)} = P_k^{(-)} - KP_{xy}^{\top}$$

Time Update

1) Draw N_p state particles from the posterior:

$$x_k^{(i)} \sim \mathcal{N}(\hat{x}_k^{(+)}, P_k^{(+)})$$

2) Draw N_p process noise particles:

$$w_k^{(i)} \sim \mathcal{N}(0, Q)$$

3) Propagate each particle through the full nonlinear dynamics:

$$x_{k+1}^{(i)} = \begin{bmatrix} x^{(i)} + \frac{T_s}{2}\cos\phi^{(i)}(v_l + d_l^{(i)} + v_r + d_r^{(i)}) \\ y^{(i)} + \frac{T_s}{2}\sin\phi^{(i)}(v_l + d_l^{(i)} + v_r + d_r^{(i)}) \\ \phi^{(i)} + \frac{T_s}{b}(-v_l - d_l^{(i)} + v_r + d_r^{(i)}) \\ \alpha(\cos\tilde{\omega}\, d_l^{(i)} - \sin\tilde{\omega}\, d_r^{(i)}) \\ \alpha(\sin\tilde{\omega}\, d_l^{(i)} + \cos\tilde{\omega}\, d_r^{(i)}) \end{bmatrix} + w_k^{(i)}$$

4) Compute the prior mean and covariance for the next timestep:

$$\hat{x}_{k+1}^{(-)} = \frac{1}{N_p}\sum_{i=1}^{N_p} x_{k+1}^{(i)}$$

$$P_{k+1}^{(-)} = \frac{1}{N_p}\sum_{i=1}^{N_p}\left(x_{k+1}^{(i)} - \hat{x}_{k+1}^{(-)}\right)\left(x_{k+1}^{(i)} - \hat{x}_{k+1}^{(-)}\right)^{\top}$$

Results

With correct initial conditions and $N_p = 200$ particles the MCKF produces estimates comparable to the EKF with no Jacobian derivation required. Position states x and y converge quickly while heading and disturbance estimation mirror the EKF behavior for the same reasons of limited observability.

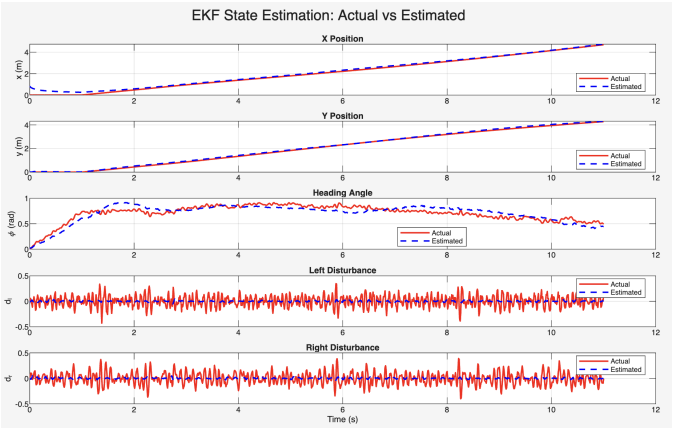

Fig. 14. $N_p = 3000$Incorrect IC's Estimated State Vs Actual

For incorrect initial conditions (x = 1 meter instead of 0) with $N_p = 200$ the filter struggled to converge, as the particle sample was insufficient to capture the true posterior. Increasing to $N_p = 3000$ yielded significantly improved estimates at the cost of increased computation time, illustrating the fundamental MCKF tradeoff: accuracy scales directly with particle count.

VIII. CONCLUSION

This paper presented a beacon-based localization system for a two-wheeled gardening robot using two nonlinear estimation approaches. Complex disturbances and measurement noise were applied. Both the EKF and the MCKF successfully tracked the robot position using only RF power measurements from four boundary beacons and deterministic wheel velocity inputs. The EKF offered fast and consistent performance with Jacobian linearization and tuned initial covariance. Linearization error manifested itself during rotation phase where heading changes quickly and the first-order approximation is least accurate. The MCKF avoided linearization entirely, capturing nonlinearities exactly through sampling, but required significantly more compute power to match EKF performance under incorrect initial conditions.