CS/ME 469: Machine Learning and Artificial Intelligence for Robotics
Assignment 0: Filtering Algorithms
Author: Jared Berry

Introduction

This article will detail the development and rigorous evaluation of an unscented Kalman filter (UKF). Starting from the design of the motion and measurement models (Sections I), attention will shift to the implementation of the UKF (Section II), as well as experimental analysis of the UKF's performance in comparison to other popular filtering algorithms and under varying levels of uncertainty (Sections III, IV, V).

The dataset used in this report is the provided ds0, which consists of control, measurement, and ground truth state trajectories from runtime. Also included is information about the landmark ground truth positions.

Section I corresponds to Questions 1, 2, 3, 5, and 6 in the assignment, Section II corresponds to Question 4, and Sections III, IV, and V correspond to Questions 7-9.
To Grader: I'm trying to improve at technical writing about robotics, so I'd appreciate any feedback about structure, formatting, figure/table placement and captioning, and experimental design. Thanks!

I. Motion and Measurement Models

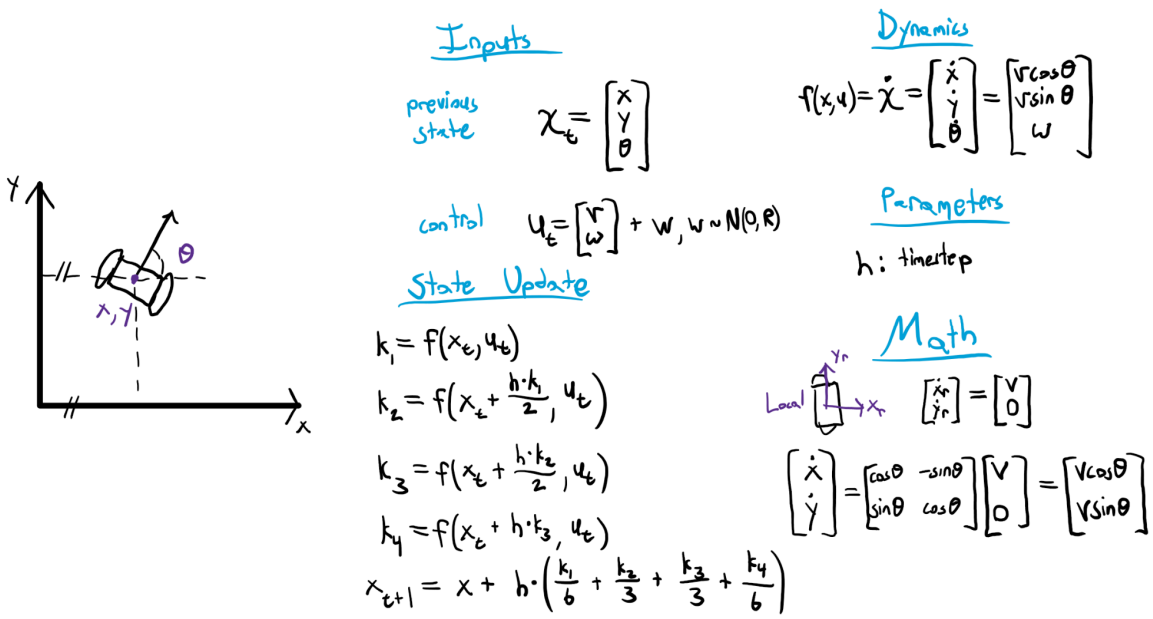

Figure 1: RK4-based motion model for a planar wheeled robot.

The motion model used in this article serves as a simple representation of a planar wheeled robot's nonlinear dynamics. As dictated by the provided control signal trajectory, the inputs to the motion model consist of forward and angular velocity commands, as well as the robot's state at time *t*. The control signals are passed into the robot's time-invariant nonlinear dynamics to produce changes in its state, which consists of x-y coordinates and a heading angle as measured from a line parallel to the global x-axis starting at the robot's 2D centroid. The model is nonlinear because the robot's velocity in the global x and y directions are dependent on the cosine and sin of the robot's heading, respectively. As seen in Figure 1, the state update occurs through a typical 4th-order Runge-Kutta (RK4) integration scheme. The motion model outputs the robot's new state at time $t + h$, with h being the simulation timestep. RK4 was chosen over a method such as Euler integration because of its superior accuracy. Gaussian noise is added to the control signal before being propagated through the nonlinear dynamics for a more robust representation of uncertainty in the system. This noise vector can be set to zero to recover the deterministic state update.

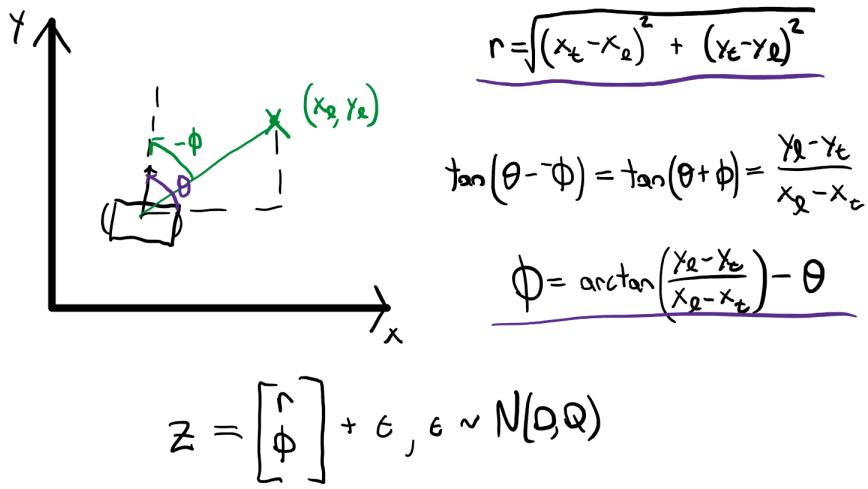

Figure 2: Range-bearing measurement model.

The measurement model is a simple range-bearing model, meant to imitate the use of a camera for identifying and taking measurements of known landmarks. Because the robot is given ground truth landmark positions, the range can be calculated with Euclidean distance and the bearing can be calculated using simple trigonometry. As seen in Figure 2, Gaussian noise is added to the measurement to imitate the uncertainty present in a realistic sensor. The deterministic measurement model can be recovered by removing the Gaussian noise.

The accuracy of the motion and measurement models are demonstrated in Figure 3. Figure 3a shows the path followed after commanding the controls from Question 2 of the assignment, which aligns with the expected path. Figure 3b shows the inaccurate dead-reckoned path after commanding the provided control signal, motivating the use of filtering. Figure 3c depicts the accuracy of the measurement model. All three tests were performed with the deterministic versions of the motion and measurement models, in order to determine their correctness.

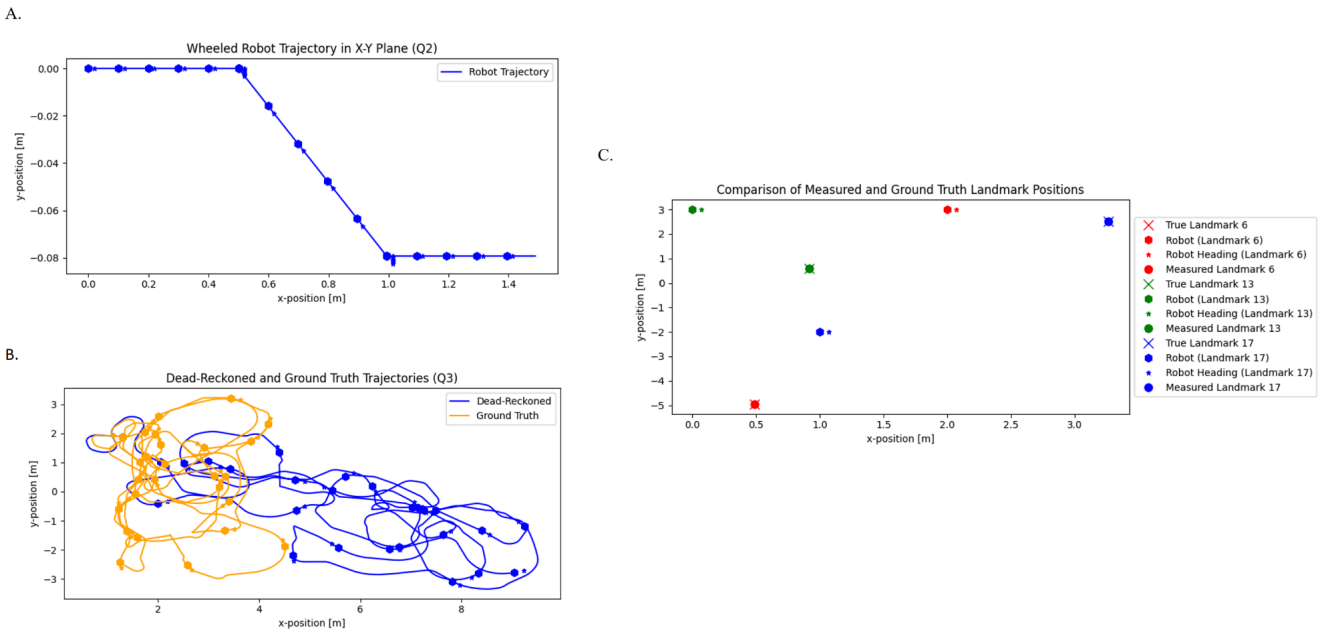

Figure 3: All figure units are in meters, and the timestep is varied according to timestamps in the control trajectory. a) Question 2 demo control signal. The control signal is made up of 3 segments of only forward velocity, interspersed with 2 turns. b) The dead-reckoned path is noticeably unaligned with the ground truth. c) Landmark locations calculated from the measurement model output align with ground truth locations.

II. UKF Implementation

The UKF is a version of the Kalman filter that allows for nonlinear dynamics and measurement models [1]. Instead of linearly approximating these models like the extended Kalman filter (EKF), the UKF takes advantage of sampling as well as the assumption that both the prior and posterior distributions are represented by Gaussians to develop a more robust estimate of state. The Gaussian assumption allows a fewer number of samples to be used, as opposed to a non-parametric filter like the particle filter.

The UKF is made up of four major steps: sampling, prediction, resampling, and measurement update. In the sampling step, samples (called sigma points) are selected in order to best preserve the Gaussian prior with a minimal number of points. The first sigma point is always the mean of the prior, and the subsequent sigma points are selected based on the square root of a matrix proportional to the prior's covariance. The matrix square root is implemented using a Cholesky decomposition. This results in *2n + 1* sigma points, where *n* is the dimension of the state vector. The sigma point sampling process can be seen in Figure 4.

$$\chi^{[0]} : \mu$$

$$\chi^{[i]} : \mu + \left(\sqrt{(n+\lambda)\Sigma}\right)_j \quad i,j = 1,\dots,n$$

$$\chi^{[i]} : \mu + \left(\sqrt{(n+\lambda)\Sigma}\right)_j \quad i = n+1,\dots,2n \quad j = 1,\dots,n$$

$$\lambda = \alpha^2(n + K) - n$$

Figure 4: Sigma point sampling method. Index i refers to the row in the sigma point matrix *X*, and index j refers to the respective column of the output of the Cholesky decomposition.

Next, the sigma points are passed through the motion model to generate a set of transformed sigma points according to the nonlinear dynamics. The unscented transform allows for the recovery of the predicted mean and covariance from this transformed set through a weighted averaging process, which can be seen in Figures 5a and 5c. If there is no new measurement, the predicted mean and covariance are used as the best estimate of state and passed through the UKF at the next timestep.

If there is a new measurement available, it can be used to update the predicted belief. First, new sigma points are sampled from the prediction at the previous step. These sigma points are passed through the measurement model to produce a transformed set of measurements. Mean belief and covariances can be extracted from this set as shown in Figure 5. Finally, the belief update occurs through the typical Kalman filter update, which was demonstrated in lecture.

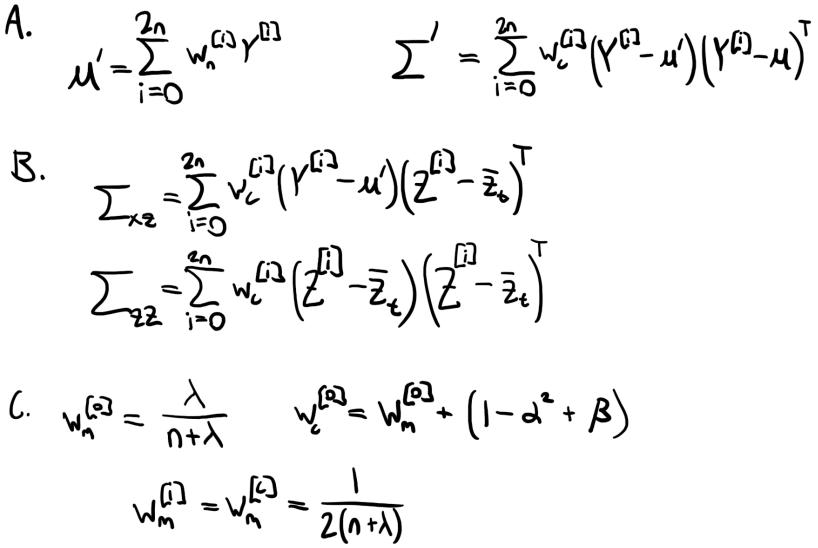

Figure 5: a) Recovery of the mean and covariance from a transformed set of sigma points. b) Covariance recovery in the measurement update step. c) Weight calculations.

The version of the UKF included in the original paper is sometimes referred to as the augmented UKF [2]. The augmented UKF algorithm involves an augmented state and covariance matrix, which incorporate the process and measurement noise into the sigma point sampling process. The augmented UKF also propagates this noise through the nonlinear models. The implementation of the augmented UKF is almost identical to the regular UKF except for two key differences: 1. The number of sigma points is now $2n_{aug} + 1$ where n_{aug} is the dimension of the augmented state vector and 2. The transformed state/measurement and respective noise are separated after sigma point generation and passed through the nonlinear models together. The version of the UKF in this report can optionally be turned into the augmented UKF, and a comparison of the two methods will be made later on.

III. Comparison to Dead-Reckoned Performance

The analysis of results will begin with a comparison between the UKF and the dead-reckoned trajectory, starting with a qualitative evaluation of the UKF's performance with the demo control signal from Question 2. The implementation of the dead-reckoned path in this report does not simulate any noise in the motion model, which is likely one cause for its inaccuracy. Figure 6 demonstrates that the UKF performs comparably to the dead-reckoned case, and that the similarity between the two methods is proportional to the amount of noise introduced into the UKF through the matrices R and Q. R and Q represent the noise present in the motion and measurement models, respectively. It is also worth noting that, because the demo trajectory data doesn't contain measurements, the matrix Q does not have any effect on performance in this case.

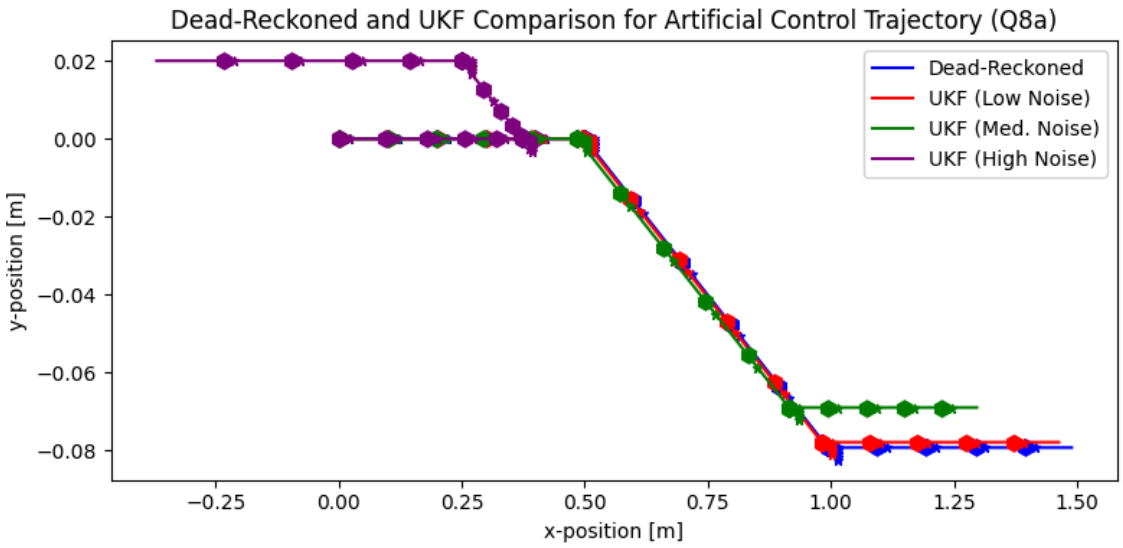

Figure 6: Performance of dead-reckoned strategy and UKF on demo trajectory, with varying noise.

The next experiment entailed testing the UKF performance for varying levels of uncertainty. For simplicity, noise matrices R and Q were set to the identity matrix multiplied by a noise constant p. For zero to small p, the UKF performs relatively consistently. Figure A1 shows that the filtered path oscillates around the ground truth, and contains a non-insignificant amount of noise. It also significantly deviates from the ground truth path, likely due to longer periods without updated measurements. For larger p, the performance worsens considerably. Even with a moderate amount of noise, the deviations from the ground truth path become unacceptably large (Figure 7). Full results of the varying noise comparisons can be found in Table A1. It is possible that the high amount of noise visible in the UKF path is due to the method for which uncertainty is incorporated into the state estimate. Because the regular UKF algorithm does not sample noise, the noise matrices R and Q are simply added to the covariance matrices like in the EKF algorithm. A more complicated approach to uncertainty incorporation could see better results from the regular UKF algorithm.

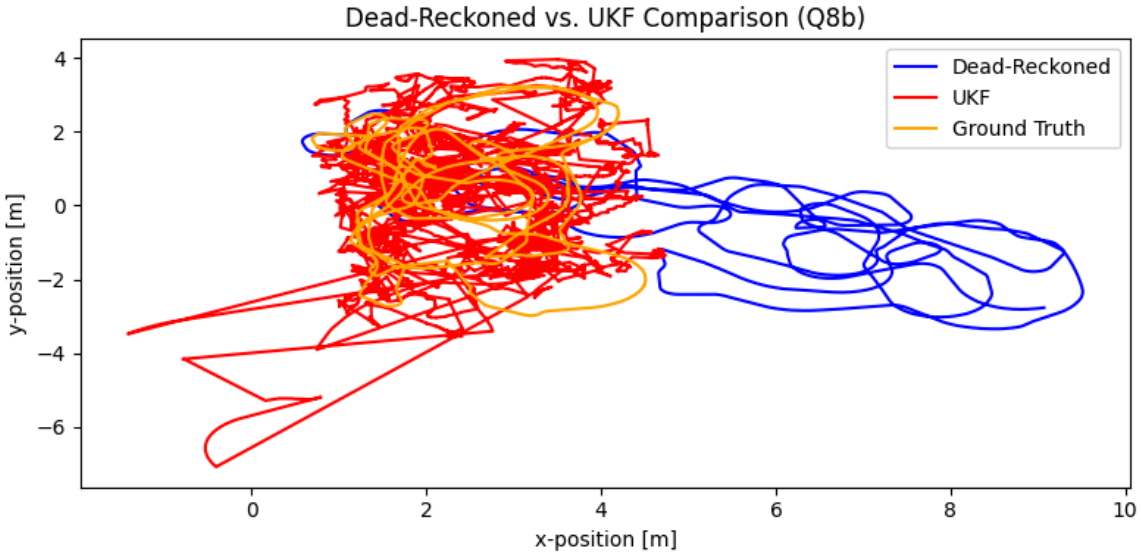

Figure 7: UKF performance for $p = 0.01$.

IV. Analysis of Augmented UKF Benefits

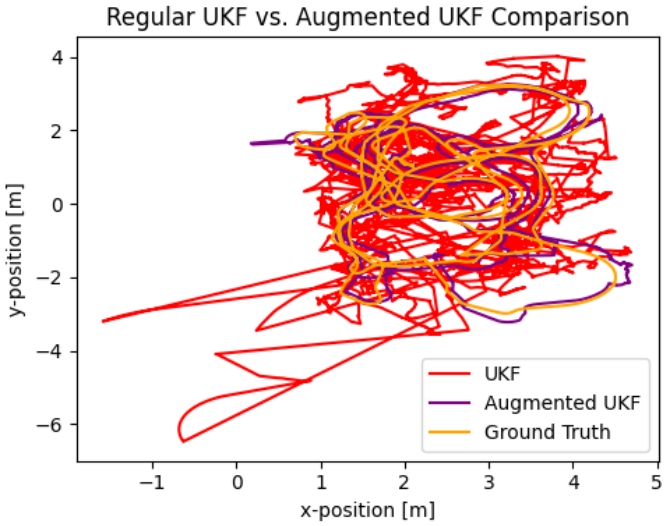

Figure 8: Path comparison of regular and augmented UKF, alongside ground truth path.

Experimental results show that the augmented UKF not only smooths a lot of the visible path oscillations and inaccuracies, but also provides more consistent results for varying levels of noise. Figure 8 shows a visual comparison of the regular UKF and augmented UKF next to the ground truth trajectory for moderate levels of noise (p=0.01). The augmented UKF follows the shape of the ground truth trajectory much more consistently, and even when it deviates the differences are comparatively small. Table 1 contains summary statistics that support the conclusions reached from viewing the path plots. Cumulative path error is both consistent and small for all levels of noise, and a high level of correlation exists between the ground truth and augmented UKF paths. These results are particularly telling when juxtaposed with the varying noise comparison statistics of the regular UKF in Table A1.

Table 1: Results from varying noise comparison between augmented UKF and ground truth paths. Metrics are cumulative root mean squared error (RMSE) and Pearson correlation coefficient (CORR).

p value	RMSE x	RMSE y	RMSE θ	CORR x	CORR y	CORR θ
0.0	6.18	7.19	13.97	0.992	0.994	0.924
0.0001	6.18	7.19	13.97	0.992	0.994	0.924
0.01	6.17	7.19	13.97	0.992	0.994	0.924
0.1	6.11	7.19	13.97	0.992	0.994	0.924

One possible explanation for the augmented UKF's performance gains is the inclusion of the process and measurement noise in the sigma point sampling process, and the subsequent propagation through the nonlinear models. By including the noise representation as part of state, the drawn sigma points may provide a more well-rounded representation of the underlying distribution. Additionally, because of the augmented state's increased size more sigma points are sampled. It is possible that the improved performance is partially a result of the larger sigma point set size allowing a better representation of the prior. Finally, because the noise is sampled alongside the state, it can be used as an input to the nonlinear motion and measurement models. This may allow for a more realistic representation of the noise's effect on real hardware, aligning the predicted path with the ground truth. The augmentation of the covariance matrix likely has a similar effect.

Table 2: Results from experiment in literature, replicated for the regular and augmented UKF.

Variant	*p* value	RMSE x	RMSE y	RMSE θ	CORR x	CORR y	CORR θ
Regular UKF	1.0	115.04	86.01	27.12	0.088	0.248	0.072
	10.0	331.91	263.96	26.61	-0.240	-0.075	0.021
Augmented UKF	1.0	5.81	7.23	13.83	0.993	0.994	0.927
	10.0	10.97	8.87	14.90	0.893	0.986	0.901

Another observation of note is the augmented UKF's performance for high levels of noise, represented in this experiment by high *p* values. The authors of [2] found that, while the augmented UKF performed better than the typical variant for lower levels of noise (p=1.0), it actually performed worse for higher noise levels (p=10.0). As can be seen from Table 2, the results in this report contradict their findings. For both p=1.0 and p=10.0, the augmented UKF significantly outperformed the regular UKF. Potential reasons for this are differences in how the authors of [2] propagated noise through their motion and measurement models, as well as the structure of the models themselves.

V. Comparison to Other Filters

In addition to the UKF and augmented UKF, the given dataset was tested on custom implementations of the EKF and the particle filter. Table 3 shows a comparison of the resultant cumulative errors and correlations for each method. The EKF slightly outperformed the regular UKF, with comparable error but higher path correlations. The augmented UKF and particle filter had comparable results for all metrics, with both filtered paths having high accuracy and low amounts of noise. Figure A2 shows a visual comparison between all filtering methods. It is worth noting that the particle filter had negligible performance gains for significant computational costs, taking far longer to run than the augmented UKF despite using only 50 particles. The EKF's performance sat in between the UKF and augmented UKF, but was far more lightweight because the Jacobians required were computed analytically and only evaluated during runtime. This could serve as a decent filtering option when motion and measurement model Jacobians are simple enough to compute by hand, but ultimately the augmented UKF seems to be the best choice out of the set.

Table 3: Metrics comparing popular filtering algorithms with ground truth. All filters run with p=0.01.

Filter	RMSE x	RMSE y	RMSE θ	CORR x	CORR y	CORR θ
EKF	10.31	8.90	15.46	0.919	0.985	0.885
Regular UKF	13.97	13.84	14.26	0.751	0.939	0.916
Augmented UKF	6.17	7.19	13.97	0.992	0.994	0.924
Particle	6.51	6.84	14.07	0.991	0.996	0.922

Conclusion

This report has detailed the development of motion and measurement models, and the implementation of several filtering algorithms for robot state estimation and localization. Primarily discussed among these algorithms is the UKF, for which two variations were compared: the regular and augmented UKFs. By experimentally varying the amount of noise present in the simulated system, it was found that the augmented UKF is more robust to uncertainty and more accurate at state estimation for the given dataset. A subsequent analysis of alternative filtering algorithms such as the particle filter and EKF demonstrated some of the advantages of the augmented UKF that may not be immediately obvious.

Citations

[1] S. J. Julier and J. K. Uhlmann, "A new extension of the Kalman filter to nonlinear systems," in *Proc. SPIE Int. Soc. Opt. Eng.*, vol. 3068, Orlando, FL, USA, 1997, pp. 182–193, doi: 10.1117/12.280797.

[2] F. Sun, G. Li and J. Wang, "Unscented Kalman Filter Using Augmented State in the Presence of Additive Noise," 2009 IITA International Conference on Control, Automation and Systems Engineering (case 2009), Zhangjiajie, China, 2009, pp. 379-382, doi: 10.1109/CASE.2009.51.

A.

Table A1: Results from varying noise comparisons between UKF and ground truth paths. Metrics are cumulative root mean squared error (RMSE) and Pearson correlation coefficient (CORR).

p value	RMSE x	RMSE y	RMSE θ	CORR x	CORR y	CORR θ
0.0	10.27	8.87	15.14	0.921	0.986	0.894
0.0001	10.18	8.86	15.09	0.923	0.986	0.896
0.01	13.97	13.84	14.26	0.751	0.939	0.916
0.1	39.49	31.62	24.09	-0.077	0.323	0.364

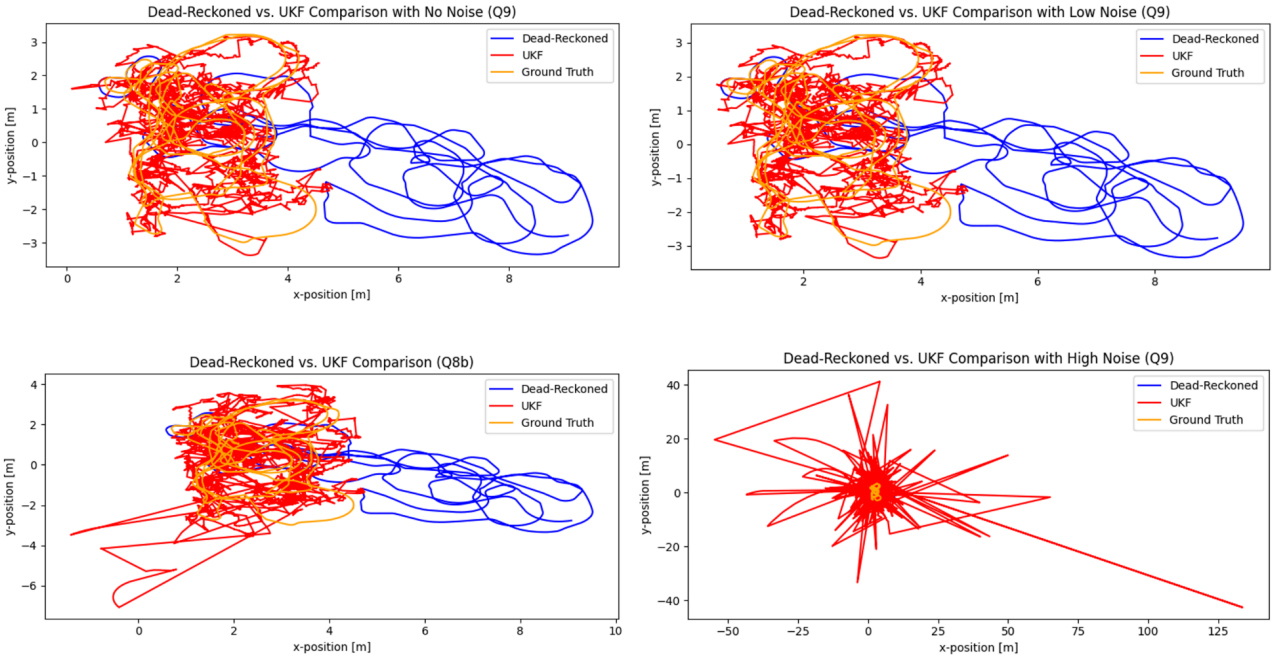

Figure A1: Performance of regular UKF for varying amounts of noise.

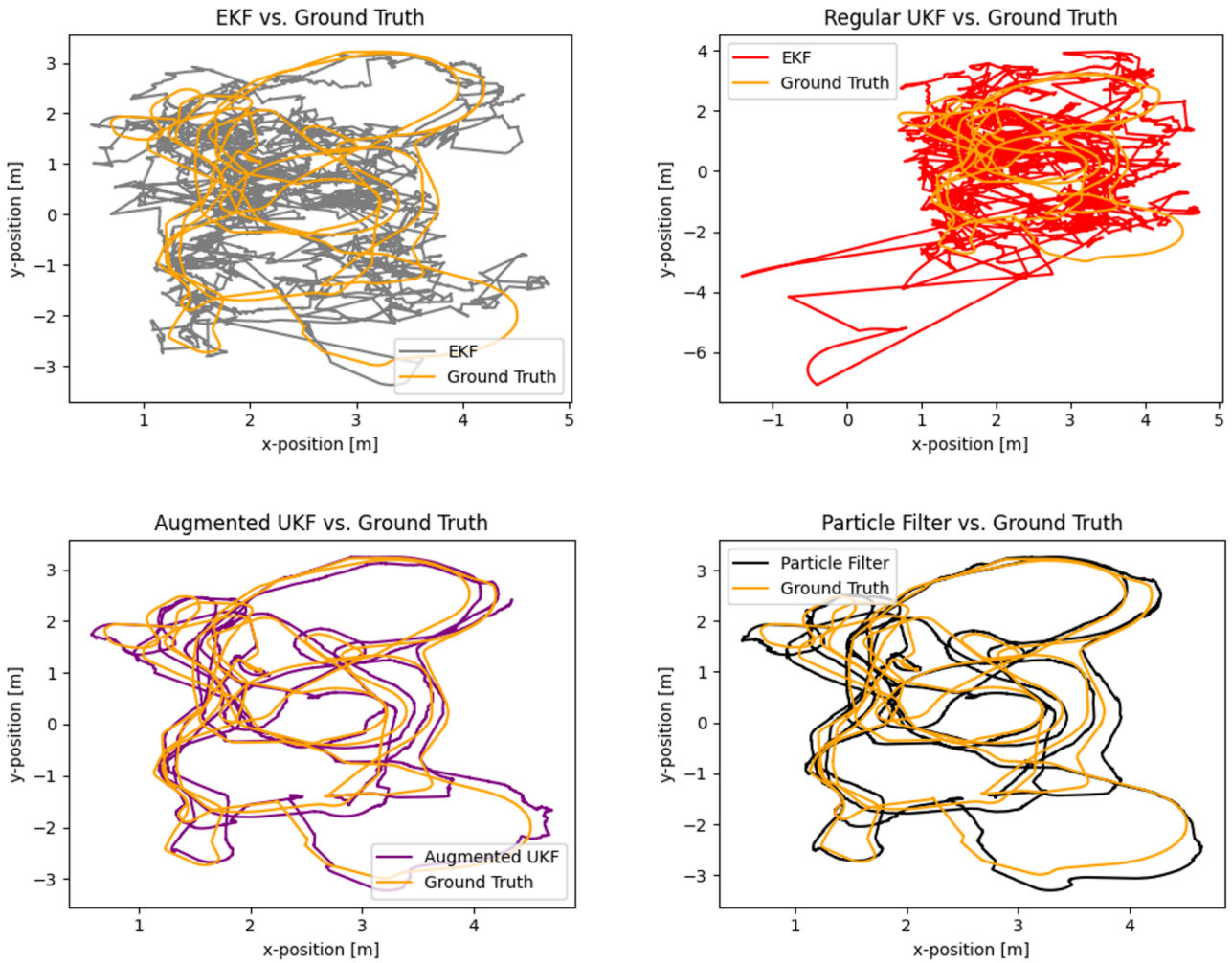

Figure A2: Filtered paths produced by EKF, UKF, augmented UKF, and particle filter. All algorithms were run at p=0.01, and plotted adjacent to the ground truth path..