MTE544 Lab 3

Group 29

Friday 3:00 PM

Kevin Xue (20814292)

Hunter McCullagh (20899434)

Justin Zhu (20879375)

Station Number: 162

Robot Number: 12

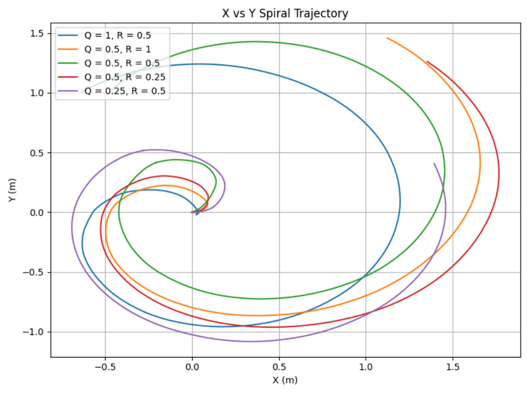


Figure 1: X vs Y Spiral Trajectory

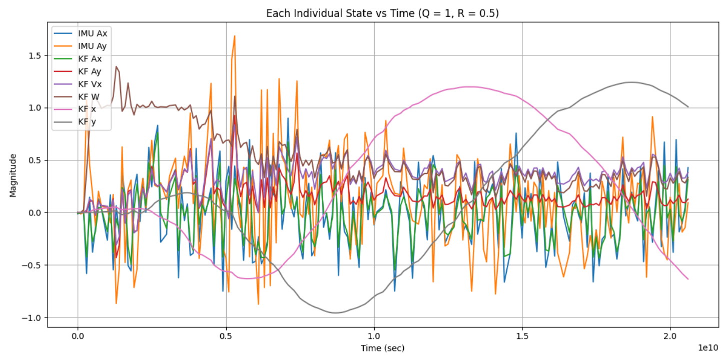


Figure 2: States with Q=1,R=0.5

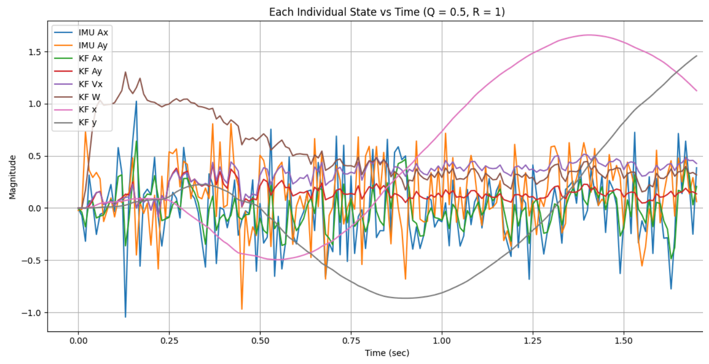


Figure 3: States with Q=0.5,R=1

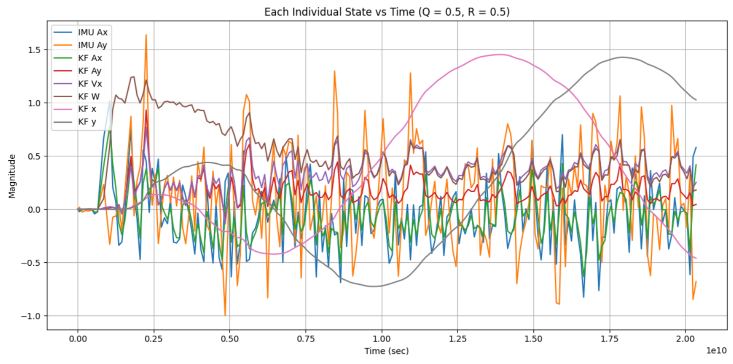


Figure 4: States with Q=0.5,R=0.5

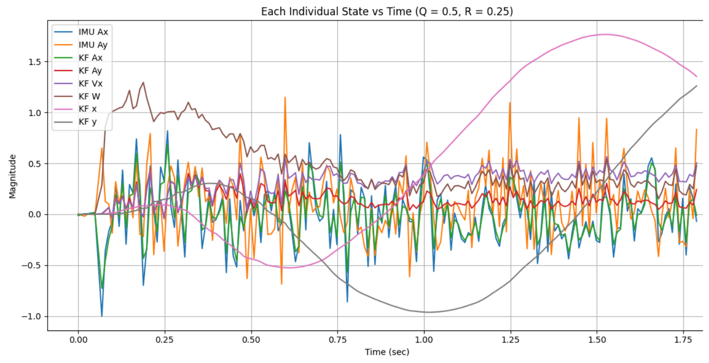


Figure 5: States with Q=0.5,R=0.25

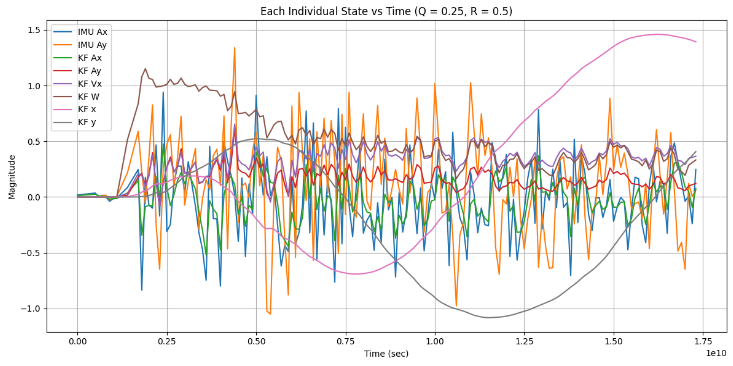


Figure 6: States with Q=0.25,R=0.5

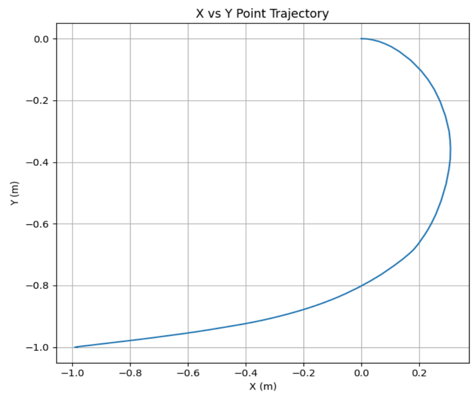


Figure 7: X vs Y Point Trajectory

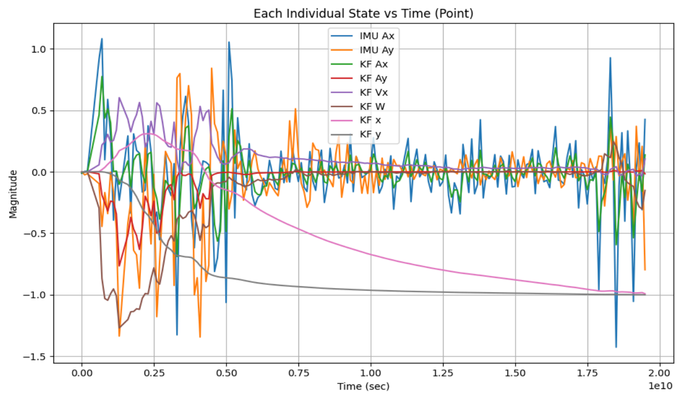


Figure 8: States vs Time

Extended Kalman Filter Algorithm

The extended Kalman Filter consists of two steps, the prediction step & the update step. The purpose of the prediction step is to estimate the next state based on the motion model. The purpose of the update step is to alter the next predicted state using measurement data and the measurement model. The equation for the prediction step can be seen in (1) where P is the matrix that projects the error covariance ahead. The equations used to derive the rows of the Jacobian motion model A can be seen in (2)-(7) where each equation represents the next state of the respective variable. By taking the partial derivative of each variable, matrix A is created. Q represents the covariance of the process noise and is altered throughout the lab. Also, the motion model is used to determine the next state. The values of the next state can be determined by using equations (2)-(7).

$P_k = AP_{k-1}A^T + Q$	(1)
$x_k = x_{k-1} + v_{k-1} * cos(th_{k-1}) * \Delta t$	(2)
$y_k = y_{k-1} + v_{k-1} * sin(th_{k-1}) * \Delta t$	(3)
$th_k = th_{k-1} + w_{k-1} * \Delta t$	(4)
$w_k = w_{k-1}$	(5)
$v_k = v_{k-1} + v_{dot\ k-1} * \Delta t$	(6)
$v_{dot\ k} = v_{dot\ k-1}$	(7)

For the update step, the Kalman gain can be seen in (8), and S is derived in (9). S relies on the Jacobian measurement model represented as C which is a matrix with its rows derived using equations (11)-(14). The equation for the next state can be seen in (15) where z represents the difference between the actual sensor data and the measurement model. The error covariance matrix P is also updated in this step as seen in (16) where A is the Jacobian motion model.

$Kalman\ Gain = PC^TS^{-1}$	(9)
$S_k = CPC^T + R$	(10)
$v_k = v_{k-1}$	(11)
$w_k = w_{k-1}$	(12)
$ax_k = v_{dot\ k-1}$	(13)
$ay_k = v_{k-1} * w_{k-1}$	(14)
$x_k = x_{k-1} + Kalman\ Gain * z$	(15)
$P_k = (A - Kalman\ Gain * C) * P_{k-1}$	(16)

Measured vs Estimated

Table 1: RMSE of measured a_x vs estimated a_x

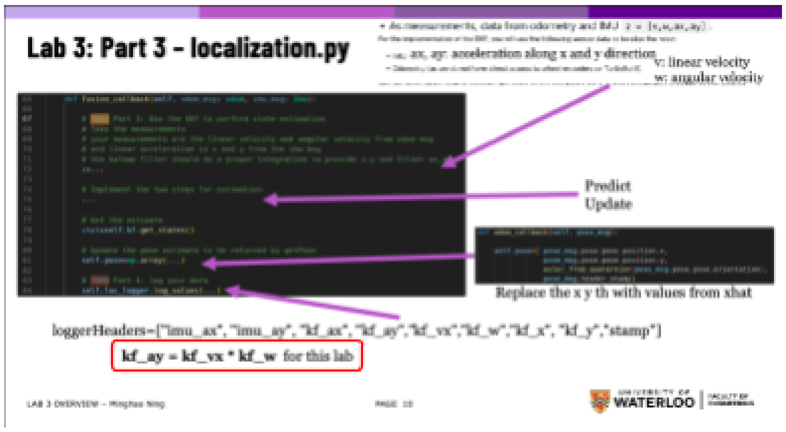
	R = 1	**R = 0.5**	**R = 0.25**
Q = 1	-	0.12077007138009052	-
Q = 0.5	0.2059573586992269	0.14160029849874023	0.10105796119680721
Q = 0.25	-	0.2063273184447196	-

Table 2: RMSE of measured a_y vs estimated a_y

	R = 1	**R = 0.5**	**R = 0.25**
Q = 1	-	0.34701717692250483	-
Q = 0.5	0.2793488861402208	0.35751389645597753	0.2664900413649417
Q = 0.25	-	0.36159201662107954	-

Discussion

Looking at the results from **Table 1**, it is clear that as the measurement noise covariance (R) decreases, so does the RMSE for a_x . This is as expected, because having a lower measurement noise covariance leads the filter to rely more on the measurement and less on the model's prediction (equivalent to a higher Kalman gain). This makes it so that the final estimated state is corrected to be closer to the measured state, thus decreasing the RMSE. This trend was not evident with a_y, which may be due to the way a_y was "measured" and "calculated":



Lab 3: Part 3 – localization.py

- ax, ay: acceleration along x and y direction
- v: linear velocity
- w: angular velocity

Predict
Update

Replace the x y th with values from xhat

loggerHeaders=["imu_ax", "imu_ay", "kf_ax", "kf_ay","kf_vx","kf_w","kf_x", "kf_y","stamp"]

kf_ay = kf_vx * kf_w for this lab

Figure 9: Screenshot from LEARN (Tutorial 6 - Lab 3 Overview, page 10)

It is also clear that as the motion model noise covariance (Q) decreases, the RMSE for a_x increases. This makes intuitive sense since a lower motion model noise covariance leads the filter to rely more on the model prediction and less on the measurement (equivalent to a lower Kalman gain), allowing for inaccuracies in the model to accumulate more over time, resulting in the larger error of the predicted state with the measured state. The same phenomenon can also be said about a_y in **Table 2**.

These observations can all be observed in the state plots of **Figures 2 - 6**, where the filtered a_x and a_y values effectively track and smoothen the IMU a_x and a_y data, while the filtered linear and angular velocity and position estimates all evolve relatively smoothly, suggesting accurate pose tracking over time.