1. Sensor fusion

1.1 Sensor models

The sensor model is used to estimate how far a robot moves from the measured data of the sensor. The chosen sensors are IR1, IR3, and Sonar 1 because they have less noise, no ambiguity, and their overall results are better than the rest. All models are created by considering the measurements as random variables. The model that is modeled by

$$Z = h(x) + V(x)$$

where $h(x)$ is the sensor data corresponding to the distance that the robot has moved and $V(x)$ is the sensor's noise, the $h(x)$ for linear sensor and non-linear sensor are described as Equation 1 and Equation 2, respectively.

$$h(x) = ax^2 + bx + c \qquad (1) \qquad\qquad h(x) = a + \frac{b}{x+c} + dx \qquad (2)$$

where a, b, c, and d are constant numbers and determined by generating the best parametric fit for the given dataset of sensors in the "calibration.csv" file to minimize the variance of the model. The point is considered an outlier if the difference of its value from $h(x)$, compared to the corresponding value in the dataset, is out of the acceptable range. This range is determined based on the distribution of the error plot's values. Coefficients of $h(x)$will be re-determined based on the new dataset. These steps are repeated until all outliers are entirely removed, and the corresponding $h(x)$ is achieved. Sensors' calibration plots can be found in Appendix A.

1.2 Motion model

The motion model is designed to associate with the sensor model to provide a better estimation of the robot's movement. The estimated range is calculated from the previous position of the robot and its current command speed. The model is represented as

$$X_n = X_{n-1} + g(X_{n-1}, u_{n-1}) + W_n \qquad \text{where: } g(X_{n-1}, u_{n-1}) = u_{n-1}\Delta t$$

This estimated speed obtained from the motion model against the true speed of the training dataset can be found in Figure 1a and Figure 1b.

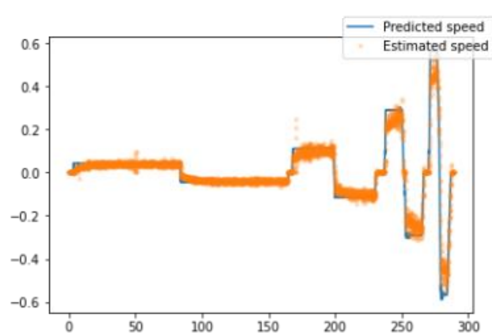

Figure 1a: Estimated speed versus predicted speed of the motion model of "training1.csv"

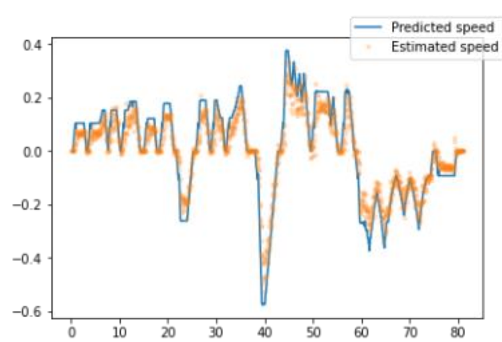

Figure 1b: Estimated speed versus predicted speed of the motion model of "training2.csv"

1.3 Bayes filter

The Bayes filter that is implemented is the Parallel Kalman filter. The posterior belief of the filter is achieved by using Equation 3 below. The posterior belief is used as the initial belief for the motion model in the next state. At initial, the initial belief is zero, and the Kalman gain is 0.8. For IR1 and IR3, because of their non-linear characteristics, their sensor models need to be linearized and $\sigma_{\hat{X}}{}^2(x)$ is obtained by the Equation 4 below.

$$\hat{x}_i{}^+ = (1 - K_i)\hat{x}_i + K_i\hat{x}_i{}^- \qquad (3)$$

$$\sigma_{\hat{X}}{}^2 = \frac{\sigma_V{}^2(x_0)}{\left(h'(x_0)\right)^2} \qquad (4)$$

where $\hat{x}_i$ is the estimate of the sensor models and $\hat{x}_i{}^-$ is the estimation of the motion model. The forecast from each sensor at specific state x is combined by using a BLUE. The $\sigma_{\hat{X}}{}^2(x)$ of the Sonar 1 is assumed to equal to the $\sigma_V{}^2(x)$.

1.4 Results

Figure 2a and Figure 2b below show the estimated range generated by the Kalman filter against the actual scope of the training datasets. The noise variance of the motion model is set at 0.0007 to avoid bias in the filter when testing the filter with the training dataset, t. For the test dataset, the initial Kalman gain is 1. The noise variance varies based on the robot's position. The estimated range of the test dataset achieved from the Kalman filter is represented in Figure 3. Figure 4 is the graph of the Kalman gain versus time.

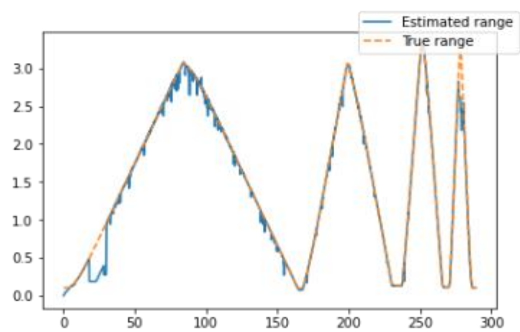

Figure 2a: Estimated range of the Kalman filter against the true range (training1.csv)

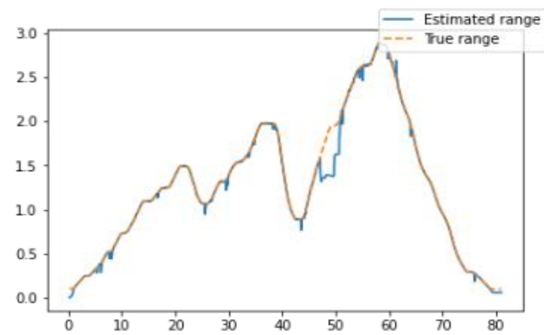

Figure 2b: Estimated range of the Kalman filter against the true range (training2.csv)

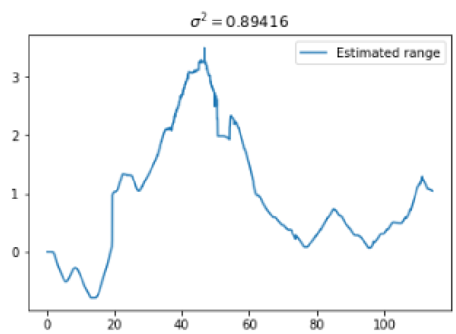

Figure 3: Estimated range of the test.csv

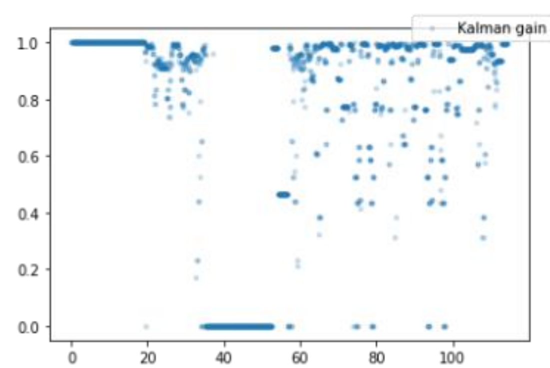

Figure 4: Kalman gain versus time

1.5 Discussion

The motion model overestimates the speed which can be seen in figure 1a and 1b. However, this does not the overall results of the Kalman filter. As Figure 3 and Figure 4 show, the Kalman filter can almost predict the next movement of the robot based on the previous state and the current command speed of the robot. However, because of the noise in the sensors, there are dissimilarities between the true range and the estimated one. From Equation 3 and Figure 2b, the motion model plays a significant role, the Kalman filter mostly trusts the motion model, comparing with the sensor models, therefore, when being tested with the training dataset, there is bias in the result of the Kalman filter. Therefore, improving the performance of the sensor models will minimize the potential of bias's occurrence in the filter.

Appendix A

Figures below represent the calibration data of IR1 IR3 and Sonar 1 with their histogram of error between measured data and models. The graphs are generated from the dataset after remove outliers

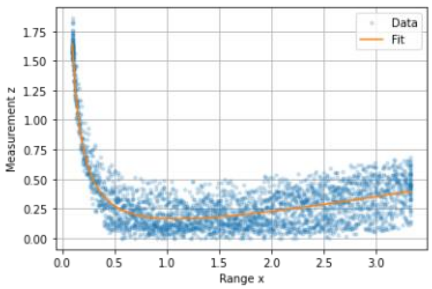

Figure 7a: IR 1 calibration data

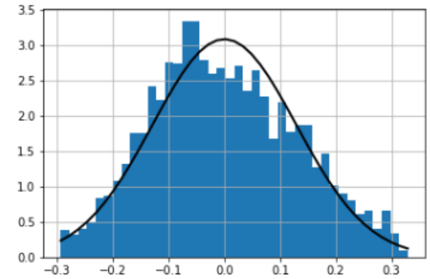

Figure 7b: Histogram of error of IR 1

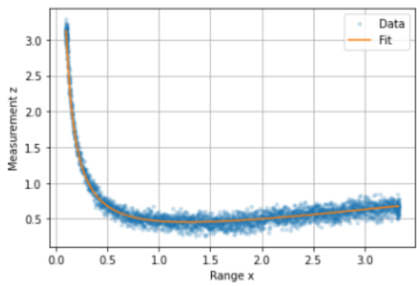

Figure 8a: IR 3 calibration data

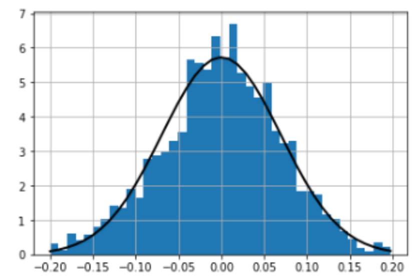

Figure 8b: Histogram of error of IR 3

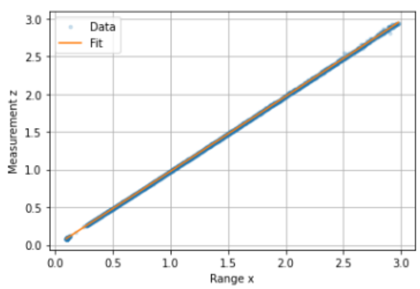

Figure 9: Sonar 1 calibration data

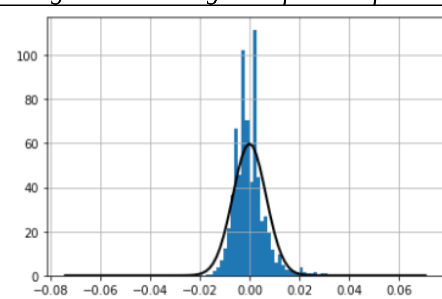

Figure 9a: Histogram of error of IR 1

Appendix B

Before the motion model in section 1.2 is used with the sensor models, another motion model is generated. That motion model has form as

$$g(X_{n-1}, u_{n-1}) = constant * u_{n-1} * \Delta t + 0.28 * accel * \Delta t^2 + \Delta v * \Delta t$$

where Δv is the difference between the current estimated speed and the previous value

This motion model provided a better estimation than the used model; however, when it is implemented in the Kalman filter with sensor models, the result is less accurate. Additionally, the constant value depends on the dataset, and it is challenging to determine this value without knowing the actual range. Therefore, the motion model in section 1.2 is used instead.