Stability Enhancement for Leader-Follower by Using Bayesian Filter based on a Probabilistic Framework

2788423 2675875 2803443

***Abstract*—This paper presents a probabilistic framework to achieve reliable directional stability in an IR sensor–based leader–follower robotic system. In an ideal threshold-based system, decisions are binary and assumed to be 100 % reliable; however, in real-world conditions, noise and other factors can cause the system to fail because it lacks tolerance and error management. By adopting a Bayesian probabilistic likelihood approach, the system can evaluate how likely a given instance belongs to the expected model, thereby handling uncertainty and making more robust and reliable decisions.**

***Index Terms*—Bayesian rule, probabilistic framework, Z-score, Log(likelihood),**

I. Introduction

The term "Bayesian filtering" is completely dependent on probabilistic values. Normally, the leader-follower program works on threshold values. If it detects an IR value greater than the threshold value, the robot assumes itself to be facing towards the corresponding direction. For example, if we train the robot that if the IR value of left sensor measured is greater than the right and center sensors, then print ("left"), if the right sensor is greater than the left and center sensors, then print ("right"), and if the center sensor value is measured to be greater than the left and right sensors, then print ("Center"). If the IR value satisfies the condition, then it shows the corresponding direction. This results in instability. The robot will act upon this rule, and this will cause a random disturbance in the motion of the follower robot. These phenomena will also be affected by the difference in noise and illumination. For instance, IR values show a random spike/raise when exposed to a noisy environment, or the IR detection rate will be low when exposed to sunlight. In other words, threshold methods are very sensitive to noise interference and minor fluctuations in IR readings and UV rays.

To obtain smooth motion stability and directional stability and avoid unnecessary disturbance caused by noise and surrounding light. We implement Bayesian filtering techniques.

if the distance between the leader and follower decreases on a particular direction, then this results in high likelihood

If it is noisy or there is fluctuation in illumination or light reflection $\Rightarrow$ It contributes less strongly.

The action by the follower doesn't depend on one reading; rather, it acts upon a series of values that the IR emits. For instance, a series of values shows [1021.92, 1021.66, 1031.55, 1021.50]. There is uncertain value seen in the fourth reading. It filters out the spiked value and follows the other reading. That is where the mean and standard deviation part comes into the picture. This will explain theoretical methodology. This causes less chaos.

The reason behind selecting Bayesian interference rather than traditional conditional probabilities is that in the traditional method we will have to list all the possible outcomes $\rightarrow$ assign probabilities $\rightarrow$ define random variables $\rightarrow$ compute distribution. However, we can measure or estimate the other probabilities that appear on the right in the inequality [1]. This concept will be explained elaborately in a later part.

Similarly, a Bayesian filter helps stabilize sensor readings by reducing the effects of environmental disturbances such as noise and changes in lighting. Sensor noise can make measurements vary even when nothing in the environment has changed. For instance, consider a robot navigating indoors using color sensors to distinguish walls, doors, and furniture. Its readings will shift depending on whether the area is illuminated by sunlight or by indoor lights. Lighting variation is just one form of noise in vision-based sensing. Other issues—such as image jitter, changes in camera gain, blooming, and blurring—can also degrade the amount of reliable information available in a color video stream [2].

The more a program calculates the average, the more it depends on its accuracy. Many algorithms solve calculating the average and optimize speed, accuracy, memory consumption, and other characteristics. One of them is the Welford's algorithm, introduced in 1962—a one-pass online algorithm with high accuracy. It is easy to understand and implement and modify to calculate variances, covariances, and other statistics. Due to its properties, it can be parallelized and used in distributed computing [3]. Welford's method is preferred in case of large numbers and small variance, as it processes data points one by one without storing the entire dataset. As it results in efficient memory by reducing accuracy issues compared to simpler formulas.

II. Theoretical Methodology

As discussed in the Abstract, various inferences like noise, systematic error, and random error may occur during the observation, which might affect the reliability of the sensor

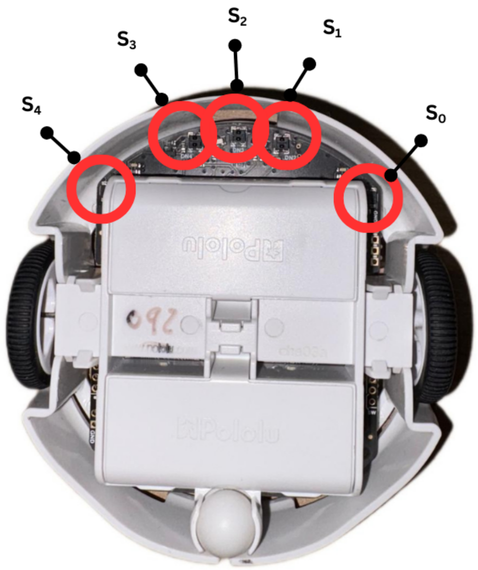

Fig. 1: 5 IR sensors in Pololu 3pi+

values and need manual thresholding. Although systematic errors can not be resolved with a probabilistic or mathematical approach, the noise and random error due to atmospheric changes can be traversed with algorithmic approach. In this experiment we have used Pololu 3pi+ mobile robots as leader-follower robots by making one follow the other with the help for infrared sensors attached to the frontal part of the system. The setup is such that the leader is made to traverse a path in reverse, and the follower is expected to tag along at a certain distance by detecting the IR emitted by the leader.

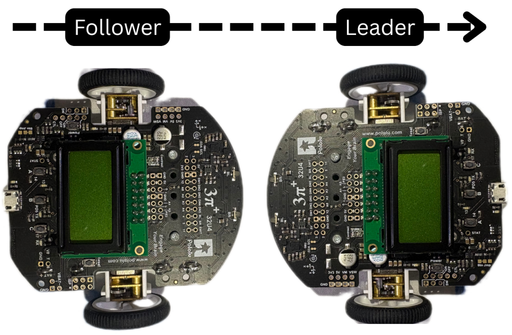

Fig. 2: Leader(inverted) and Follower moving in the forward direction.

The leader was made to travel in a straight line to assess the follower's behaviour. We have used a probabilistic approach which consists of the Gaussian likelihood Bayesian update method, where the system uses the prior IR data of each individual sensor to calculate the posterior probability of the follower in accordance with the leader.

Sensor calibration is done for the follower for 10000 microseconds without the presence of the leader, mainly to distinguish between the normal background readings from the leader's emitter signal variations. Once done, it was made to follow the leader. While both the leader and follower are traversing the path, IR sensor values are collected. To add more evidence to our hypothesis, we conducted our experiment in two different illumination conditions. One during the day, during sunlight interference and the other in a closed, artificially illuminated room, which is night.

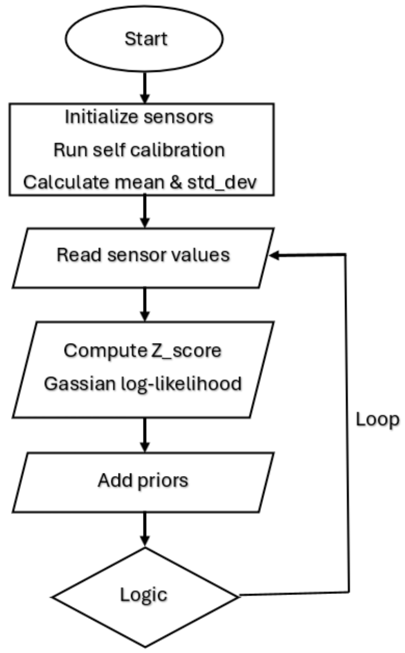

Fig. 3: The flowchart depicts computation sequence of the Bayesian classifier

This flowchart shows the step-by-step procedure we have followed to implement the Bayesian filter. The process starts with initialisation of the sensors and then the calibration is done. With IR readings we calculate the mean and standard deviation of the IR sensors during stable and unstable atmospheric conditions like interference of sunlight and unfiltered noise. IR sensor values varied in both conditions.
The below graph holds the mean and standard deviation of the 50 samples collected for the 5 sensors from leftmost to rightmost in the follower robot.

IR sensor values varied in both conditions. Figure 4 holds the mean and standard deviation of the 50 cycles collected. for the 5 sensors from left most to right most in the follower robot.

We calculated the likelihood of the data point under the given model. The following steps need to be calculated in order to make a decision for the follower.

Step 1: Data Acquisition
Step 2: Z score Calculation
Step 3: likelihood calculation
Step 4: Combined likelihood

Step 5: Add assumed shared probability log(0.33) to each likelihood combination $(L_0 + L_1,\ L_2,\ L_3 + L_4)$

Step 6: Normalize the likelihood values, using softmax.

Step 7: Manoeuvre

Data Acquisition:

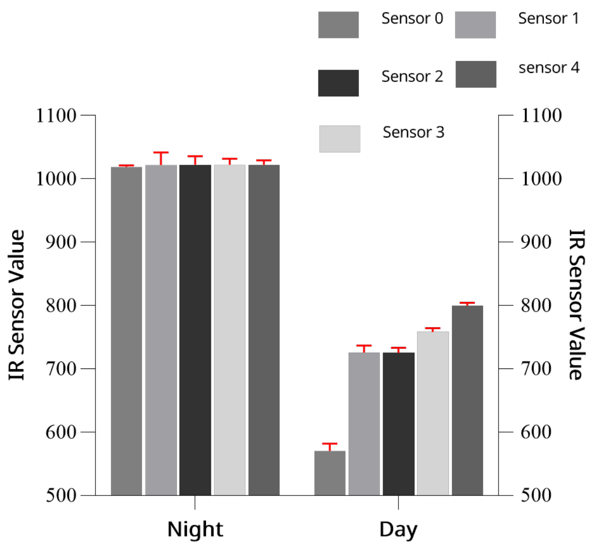

Fig. 4: Mean and Standard Deviation of 5 IR sensors during day and night

Z-score Calculation :
Z-score decides how far the data point is away from the mean in the units of standard deviation [4]

$$z = \frac{x - \mu}{\sigma}$$

where,

$$Z = \frac{\text{data point} - \text{mean}}{\text{Standard Deviation}}$$

Mean and standard deviation are calculated during the calibration.

if $|Z|$ is far away from the mean, the likelihood of the data point is low.

if $|Z|$ is closer to the mean, the likelihood of the data point is high.

The below table has the Z score of data points for each sensor respectively.

TABLE I: Z-score

	Z0	Z1	Z2	Z3	Z4
Day	-0.30	-4.75	-7.77	-8.29	-5.70
Night	-7.18	-7.76	-6.19	-4.84	-1.05

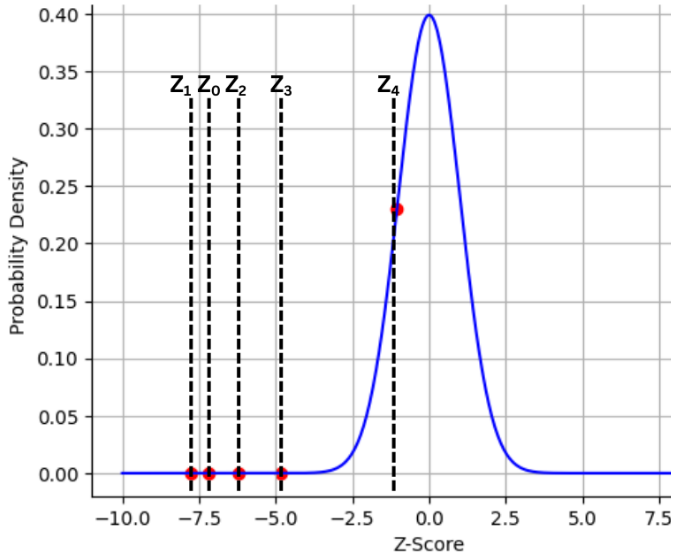

Fig. 6: Z Score during Day for a given instance when the leader turned right.

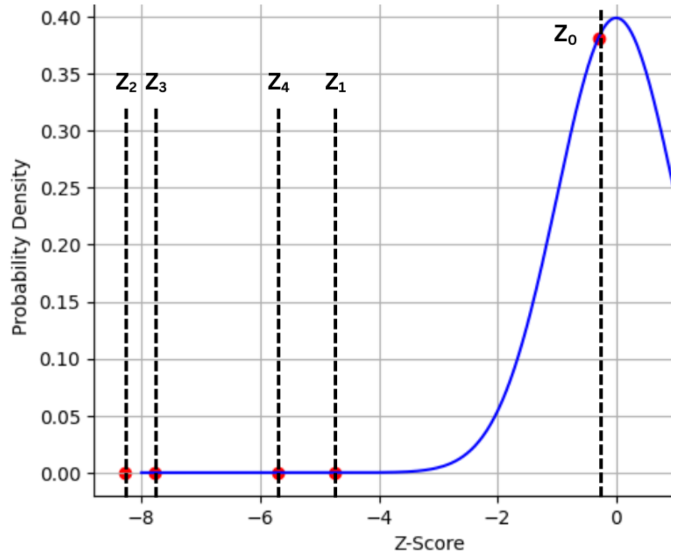

Fig. 7: Z Score during night for a given instance when the leader turned left.

In the day-time graph, Z_0 and Z_1 are near zero. Since the Z-score is exponentially related to the likelihood, this serves as a visual cue for how the follower decides the direction. Similarly in Night-time graph Z_3 and Z_4 are closer to the zero.

Gaussian Likelihood Function:

The parameters used to determine how likely a data point is under the model are the **mean** (μ) and **standard deviation** (σ).

The **likelihood** measures how well the observed data fits the model:

$$L = f(X \mid \mu, \sigma),$$

where X is the observed data point, and μ, σ are the parameters of the model. [5]

$$L(X \mid \mu, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(- \frac{(x-\mu)^2}{2\sigma^2} \right)$$

Unlike probability, likelihood is not between 0 and 1. It can be a number greater than 1 or less based on the given circumstances. Probability is the chance of observing a particular outcome, given a model. where likelihood measures how plausible it is to observe x if the true mean and sigma are μ, σ.

Let the data point be $x = 1019$ and the model have parameters $\mu = 1018$ and $\sigma = 0.2$.

$$L(1018, 0.2 \mid 1019) = \frac{1}{\sqrt{2\pi}\,0.2} \exp\left[-\frac{1}{2 \cdot 0.2^2} (1019 - 1018)^2 \right].$$

Although 1019 seems reasonable, due to very low standard deviation we can conclude that the data point is unlikely given the mean and standard deviation. Also the exponential part of the function can be represented with the help of Z score and calculate the likelihood.

Since we have only three possible decisions for a follower to follow the leader (Turn Left, Go Straight, Turn Right). The experiment was conducted for two different methods of likelihood data.

1. Combined Likelihood

- Left turn: $L_0 + L_1$ (Likelihood of 2 left side sensors)
- Straight: L_2 (Likelihood of center sensor)
- Right turn: $L_3 + L_4$ (Likelihood of 2 right side sensors)

2. Individual likelihood

- Left turn: L_0 (Likelihood of leftmost sensors)
- Straight: L_2 (Likelihood of center sensor)
- Right turn: L_4 (Likelihood of rightmost sensors)

Next, we introduce a prior to the likelihood combinations to calculate the posterior probability. The prior is assumed as:

$$\text{Prior} = \log(0.33)$$

The prior probability for all three directions is 0.33 to maintain equal confidence before calculating the posterior probability. That is:

$$p(\text{Left}) + p(\text{Center}) + p(\text{Right}) = 1$$

After adding the prior to the likelihoods, we normalize the values between 0 and 1 for probabilistic decision-making using a softmax function [7]:

$$P_i = \frac{e^{L_i - \max(L)}}{\sum_{j=1}^{3} e^{L_j - \max(L)}}$$

The softmax probabilities are then:

$$P_{\text{left}} = \frac{e^{L_{\text{left}}}}{e^{L_{\text{left}}} + e^{L_{\text{center}}} + e^{L_{\text{right}}}}$$
$$P_{\text{center}} = \frac{e^{L_{\text{center}}}}{e^{L_{\text{left}}} + e^{L_{\text{center}}} + e^{L_{\text{right}}}}$$
$$P_{\text{right}} = \frac{e^{L_{\text{right}}}}{e^{L_{\text{left}}} + e^{L_{\text{center}}} + e^{L_{\text{right}}}}$$

$$\text{Maneuver: } \begin{cases} \text{Straight,} & \text{if } P_{\text{center}} > P_{\text{left}} \text{ and } P_{\text{center}} > P_{\text{right}} \\ \text{Turn Left,} & \text{if } P_{\text{left}} > P_{\text{center}} \text{ and } P_{\text{left}} > P_{\text{right}} \\ \text{Turn Right,} & \text{if } P_{\text{right}} > P_{\text{center}} \text{ and } P_{\text{right}} > P_{\text{left}} \end{cases}$$

III. Experiment

IV. Experimental Methodology

The hypothesis was tested with the help of experiments in two different scenarios:

1) Leader–Follower by Thresholding
2) Leader–Follower with Bayesian Filter

In both conditions, the white bumper part of the Pololu 3pi+ robot was removed to increase the intensity of the IR sensor values.

A. Thresholding

In this experiment, the follower needs to move with the leader and take left, right, or go straight with respect to the strength of the combination of the five IR values.

Thresholding Algorithm:

For sensors in $[S0, S1, S2, S3, S4]$:

- If $(S0 + S1) < S2$ and $(S3 + S4)$, turn **Left**
- If $S2 < (S0 + S1)$ and $(S3 + S4)$, go **Straight**
- If $(S3 + S4) < S2$ and $(S0 + S1)$, turn **Right**

The following tables show the IR readings in the follower robot when the leader is at different positions during daytime and night-time. Measurements were taken with a distance of 2 cm between the robots.

TABLE II: IR Readings (Day-Time)

Position	S0	S1	S2	S3	S4
Centre	912	879	858	885	967
Left	812	880	921	952	980
Right	921	938	910	883	887

TABLE III: IR Readings (Night-Time)

Position	S0	S1	S2	S3	S4
Centre	1017	976	966	975	1015
Left	972	972	988	1005	1020
Right	1020	1010	997	985	989

B. Bayesian Filter

The setup is similar to the thresholding method, but instead of using IR value ranges directly to decide direction, a likelihood model is used to evaluate the IR data points and determine the behaviour of the follower using the logic shown in Figure 8.

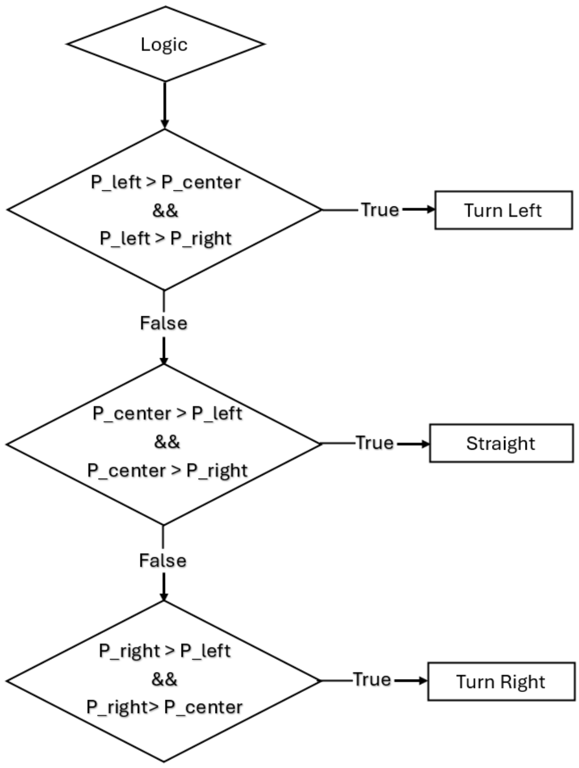

Fig. 8: Logic behind manoeuvring decision

Figure 8 shows the flowchart that depicts the conditional statements behind the follower robot's decision-making process.

V. Result

To compare follower trajectories fairly, the leader was made to move in a straight line. With Figure 9, it is inferred that the Follower shows unusual behaviour and extreme deviation from the Leader's path. To overcome this, we tested the leader-follower with Bayesian filter.

In Bayesian filter method, we tested Leader-follower with two types of likelihood calculation. In the first method, the unified Likelihood estimates are derived from Left, centre and right sensor readings. The latter approach uses likelihood from the leftmost S0, Rightmost S4 and Center S2 sensors, right and center posterior probability.

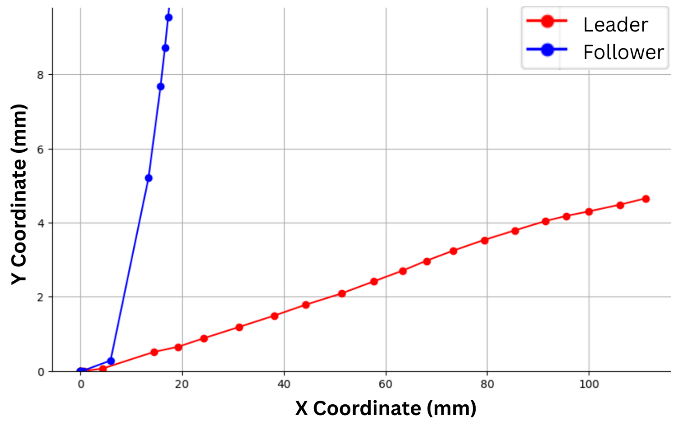

Fig. 9: Trajectory profile of Leader-Follower using threshold method

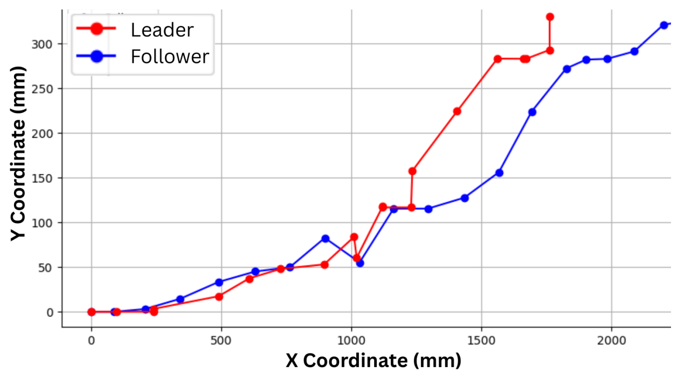

Fig. 10: Trajectory profile of Leader-Follower using Bayesian filter

The follower cannot track when combined likelihood is used. These deviations are caused by the center sensor, which is less effective in likelihood, while the left and right sensors have higher combined likelihoods.

To mitigate this problem, We used individual likelihood. Figure 10 shows the leader-follower trajectory only with the likelihoods of 3 sensors, namely S0, S2, S4. The follower is able to track the path of the leader with some deviations.

Figure 11 Shows the heading value of the follower relative to the leader in a straight line while using a Bayesian filter. The deviation drifts larger over the distance.

Therefore, We hypothesize that although both methods show deviation, in comparison, the one with bayesian filter performed better with directional stability by adopting the Bayesian probabilistic likelihood approach.

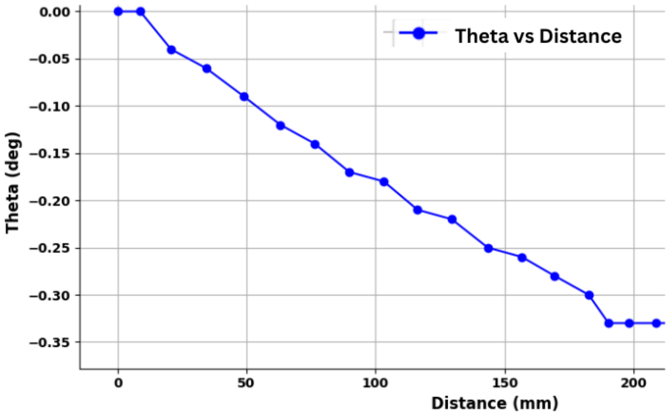

Fig. 11: Heading profile of Leader-Follower using Bayesian filter

VI. Conclusion

Based on the experimental evaluation we noticed that the follower robot was able to follow the leader robot with the Bayesian filter. In the case of the threshold method, little deviation by the leader robot would cause drift in the motion of the follower. This threshold method makes the follower behaves in a unusual manner.

As we stated in the hypothesis using prior probability tracking error was reduced. Bayesian filter helped in attaining better directional stability than the threshold method. We were able to reduce the drifts during straight motion. Also Bayesian filters works well when the distance between leader and follower is less.

Below are some of the obstacles we faced during data collection and maneuvering:

1. At night, in some instances, all the IR values reach almost 1023 ADC values, which in turn affect the Bayesian filter.

2. when distance between leader-follower increased beyond 14 cm. There was notable tracking error, jitter during motion lack of sensibility during directional change due to weak signal.

3. The Probability division between 3 posterior position is unfair because the Posterior-Left uses 2 sensors and the Posterior-Right uses 2 sensors for the combined likelihood, but the Posterior-Centre uses only one sensor,which in turn affects the confidence of the center sensor.

VII. Future scope

In future work, we can still work on improving noise filtering and develop an advanced filtering algorithm to filter out disturbance caused by noise fluctuations from the surroundings.

Improving the sensitivity when the distance between leader-follower is large. More robust probabilistic filtering can be developed to minimize variations and improve the reliability of sensor-based decision making

References

References

[1] A. Neapolitan, R. E., *Learning Bayesian networks*, Upper Saddle River, NJ: Pearson Prentice Hall, 2004.

[2] A. Gilmartin, M.J., *INTRODUCTION TO AUTONOMOUS MOBILE ROBOTS*, MIT Press, 2005.

[3] A. A.Efanov, B. S.A.Ivliev C. A.G.Shagraev *Welford's algorithm for weighted statistics*, International Youth Conference on Radio Electronics, Electrical and Power Engineering (REEPE), Moscow, Russia, 2021.

[4] Andrade C. Z Scores, Standard Scores, and Composite Test Scores Explained. *Indian J Psychol Med.* 2021;XX:1–3.

[5] R. Siegwart and I. R. Nourbakhsh, *Introduction to Autonomous Mobile Robots*, Intelligent Robotics and Autonomous Agents, A Bradford Book, MIT Press, 2004. pp. 145–149 ISBN 0-262-19502-X.

[6] *PROBABILISTIC ROBOTICS*, Sebastian Thrun, Dieter Fox, Wolfram Burgard, 1999-2000. pp. 10–22

[7] https://www.geeksforgeeks.org/deep-learning/the-role-of-softmax-in-neural-networks-detailed-explanation-and-applications/