MATEC Web of Conferences **42**, 03008 (2016)
DOI: 10.1051/matecconf/20164203008
© Owned by the authors, published by EDP Sciences, 2016

Multi-modal mobile sensor data fusion for autonomous robot mapping problem

M. H. Kassem, Omar M. Shehata, E. I. Imam Morgan
Robotics and Autonomous Systems (RAS) Research Group, German University in Cairo (GUC),
5th settlement - New Cairo, 11432, Cairo, Egypt
Website: http://www.ras-lab.com, Email: ras.research.group@gmail.com

Abstract—**Perception is the first step for a mobile robot to perform any task and for it to gain perception mobile robots use sensors to measure the states which represent the surrounding environment. Sensors measurements are always combined with some sort of uncertainty and noise. Which can make the system very unstable and unreliable. In order to get better readings we can always use better types of sensors where we come to a trade off between price and quality. And that's why our proposed approach to solve this problem was to use data fusion techniques to eliminate the noise and reduce the uncertainty in the readings. The topic of data fusion has been under extensive research in the past decade many approaches had been suggested and yet the research on data fusion is increasing and this because of its importance and applications. This study discuss the use of probabilistic data fusion techniques to reduce the uncertainty and eliminate the noise of the measurements from range finder active sensors to improve the task of mapping for mobile robots. The data fusion methods used were Kalman filter and Bayes filter.**

Keywords: Mobile Robots, Data fusion, sensor noise, uncertainty, Kalman filter, Bayes filter, Ultrasonic.

I. Introduction

After the revolution of the industrial robots and there great success there was an extensive research on mobile robots due to there applications, the wide range of tasks which they can perform and their ability to perform the tasks in more precise, accurate and even faster way than humans and that's why one can easily see them replacing us. One task which has been highly required from mobile robots to perform is mapping. Mapping is the ability of the robots to build a map of the surrounding environment and this can be in places which we can't reach as the space and other planets, places which is dangerous for us to reach as the mine fields and even for normal buildings.

Humans are using data fusion every day of their lives because the brain is the best data fusion device that we ever knew and will ever know. The human brain fuse the data from all your senses vision, hearing, touching, smelling and taste to give you a perception of the environment. It fuse the data from homogeneous sensors as the eyes to give one fused vision of the world and it fuse the data from all the other heterogeneous sensors together to tell you what is actually happening around you and to make you able to take proper decisions, to be able to balance, walk, run and all the other tasks. What we are interested in in Sensor Data fusion is the idea of integrating the data from different sensors to get a better set of data which is more accurate(reduce uncertainty) and precise(eliminate noise) than what we get using only one sensor alone. Given those sensors are of the same type(Homogeneous).

In the rest of this paper we will talk about the type of sensor we used, explain the fusion techniques used, present our results and simulations and finally we will sum up our work and state the future recommendations.

II. A review of Probabilistic data fusion techniques

The Bayes estimator set the base for all of the probabilistic data fusion techniques, which enables fusion of pieces of data, hence the name Bayesian fusion. Also we have the well known Kalman filter which is actually a especial case of the Bayes filter with an exact analytic solution due to it enforce the simplifying of constraints on the system dynamics to be linear-Gaussian[1]. But yet those filters deals only with linear systems so when it comes to nonlinear systems we use different approximation techniques.

One of those techniques is the Extended KF [2] and Unscented KF [3]. Both of them are extensions of the Kalman filter and they are applicable to deal with non-linear systems.Also the Grid-based methods [4] provide another approach for approximating non-linear probability density functions, although they rapidly become computationally difficult to deal with in high dimensions.

The Monte Carlo simulation-based techniques such as Sequential Monte Carlo(SMC) [5] and Markov Chain Monte Carlo (MCMC) [6] are within the most powerful and used methods of approximating probabilities. Also they do not make any assumptions regarding the probability densities which is been approximated and this actually make it very flexible.Particle filters are a recursive implementation of the SMC algorithm [7]. It can substitute the Kalman filter when dealing with non-Gaussian noise and nonlinear systems.

III. Methodology

Our aim in this study is to be able to develop data fusion algorithms which can overcome the problem of Cross talk, reduce the uncertainty and eliminate the noise caused by the cross talk phenomena from the sensor's readings. In this section we will introduce the sensor used in our study, the data fusion techniques used and finally our proposed approach.

A. Sensors

Different types of sensors are used in mobile robots for them to be able to recognize the surrounding environment and to know their status. Sensors are the first step in order for the robot to gain perception. The sensor type can be described by 2 main characteristics, it can be proprioceptive or exteroceptive and active or passive. Proprioceptive sensors are the sensors used to measure the internal values of the robot and determine its internal status like motor speed and internal temperature of the components. On the other side exteroceptive sensors are the ones used to gather information from the surrounding enviroment of the robot as the distance from a certain object or the light intensity of the ambient environment. The other characteristic which define the sensor has to do something with the energy received or transmitted by the sensor. Active sensors emit energy to the surrounding environment and measures values according to the reaction of the environment. Examples of the active sensors are Ultrasonic and Laser range finder sensors. Unfortunately specially for Ultrasonic sensors we face what we call the Cross talk phenomena. Cross talk means that a sensor pick up a signal sent from another sensors due to they are near to each other by being on the same robot or even on different robots and this cause the sensor to give false and very noisy readings. Meanwhile passive sensors measure the energy which they receive from the environment for example thermometers and microphones.

1) Ultrasonic sensor: The basic idea of the Ultrasonic sensor is that it sends an ultrasonic waves and wait for those waves to come back and measure the time in between and through this we can calculate the distance between the sensor and the object which cause the waves to reflect [8].

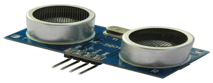

Fig. 1. The ultrasonic sensor used

$$d = \frac{c \times t}{2} \tag{1}$$

d = distance from the object.
c = speed of sound.
t = time taken for the waves to reflect.

$$c = \sqrt{\gamma RT} \tag{2}$$

γ = ratio of specific heats.
R = gas constant.
T = temperature in degrees Kelvin.

B. Used data fusion techniques

When we came to choose which data fusion techniques we will be using in our project. And because it is uncertainty which we needed to eliminate we chose the probabilistic data fusion and you can see why we chose the probabilistic data fusion in figure 2.The 2 probabilistic data fusion techniques which we decided to choose are the Kalman filter and the Bayes filter.

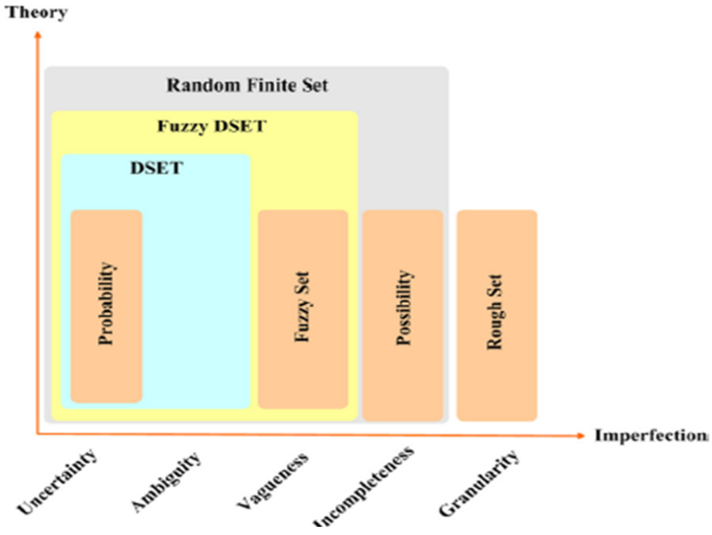

Fig. 2. Data fusion methods handling the data fusion problems [9]

1) Bayes filter: The Bayesian method set the base for all the probabilistic data fusion techniques. The way the Bayes filter work is as follows

$$p(X_K|Z_K) = \frac{p(Z_K|X_K)p(X_K|Z_{K-1})}{p(Z_K|Z_{K-1})} \tag{3}$$

X_k is the state X at the time k
Z_k is the measurement of the state at time k
p($X_k|\mathrm{Z}_{k-1}$) is the prior distribution
p($X_k|\mathrm{Z}_k$) is the posterior distribution
p($Z_k|\mathrm{X}_k$) is the likelihood function
p($Z_k|\mathrm{Z}_{k-1}$) is the normalizing term

So first of all we have to define our parameters. Our state in this project is the distance of the obstacle from the sensor. The way this works is that first of all we have to assume a prior distribution to start with. In our project we assumed an equal distribution for every possible point that we have which in our case can be any way from 2 to 200 cm.

Then we calculate the likelihood function using the coming equation. The likelihood function basically represent how likely this given reading from the sensor can be true taking into consideration the given state of the system and used the Gaussian distribution to calculate the likelihood function as in the coming equation.

$$p(Z_K|X_K) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(X_K - Z_X)^2}{2\sigma^2}} \tag{4}$$

σ = error co variance of the sensor which can be measured

For our last term which is the normalization term. We actually use this term to normalize the equation and make sure that the distribution integrate to 1 which make sense actually because the distribution is basically setting a probability for each possible point to happen depending on the input reading from the sensors and because it's probability so the summation of probabilities which are for the same event have to be equal to 1. And the way we normalize the distribution is by dividing by the sum of all the elements in the distribution by each element in the distribution. The final step that we do is that we set the posterior distribution which we have just calculated to the prior distribution and repeat the steps again.

In the case where we will have more than one sensor to deal with we need to fuse the readings of the sensors together into one reading and you can see in the coming equation how we managed to do that using the error co variances of each sensor.

$$Z_f = \frac{\sigma_2^2}{\sigma_1^2 + \sigma_2^2} z_1 + \frac{\sigma_1^2}{\sigma_1^2 + \sigma_2^2} z_2 \tag{5}$$

The last step in order to fuse the sensors together is to calculate a fused co-variance for the 2 sensors in order to use it to calculate the likelihood function. So we used equation 6 to calculate the fused value of the co-variance then we can use the same steps as before to get the distribution then after we finish we take the point with the higher probability and use it as the sensor reading. And this is how our Bayes filter work

$$\sigma_f^2 = \frac{1}{\sigma_1^{-2} + \sigma_2^{-2}} \tag{6}$$

2) Kalman filter: How the Kalman filter work is actually pretty simple there are 2 phases of the Kalman filter the prediction phase and the correction phase as shown in figure 3

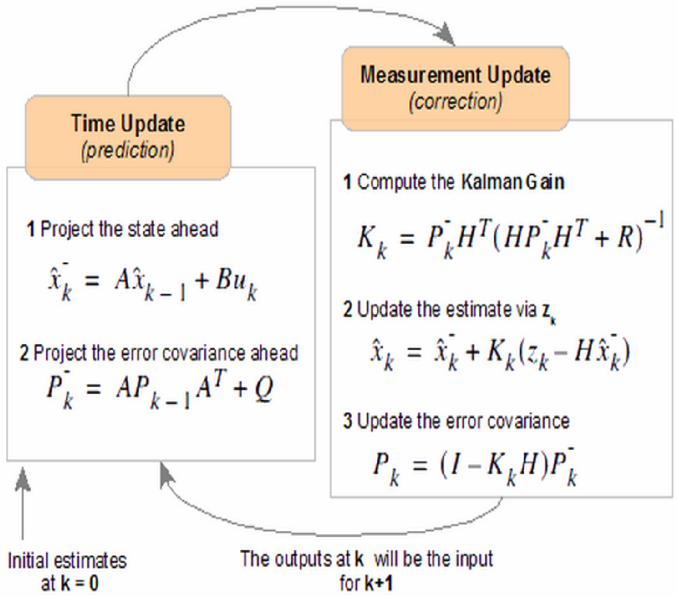
Measurement Update
(correction)
Time Update
(prediction)
1 Compute the Kalman Gain
$K_k = P_k^- H^T (H P_k^- H^T + R)^{-1}$
1 Project the state ahead
$\hat{x}_k^- = A\hat{x}_{k-1} + Bu_k$
2 Update the estimate via z_k
$\hat{x}_k = \hat{x}_k^- + K_k(z_k - H\hat{x}_k^-)$
2 Project the error covariance ahead
$P_k^- = AP_{k-1}A^T + Q$
3 Update the error covariance
$P_k = (I - K_k H)P_k^-$
Initial estimates at k = 0
The outputs at k will be the input for k+1

Fig. 3. Kalman filter cycle

So we start by the prediction phase. First we define our state space model of our system. X_k is our state which we already mentioned in the past section. The A is our state vector with size nxn where n is the number of states which in our case is one so A is 1x1 matrix. The B matrix is called the input matrix which is mxn matrix where m is the number of inputs which is u_k and because we do not have inputs to our system so u_k is equal to zero and the B matrix will not exist. Then we come to the second equation which define the error co-variance of the system P_k is the error co-variance of the system and the Q is what we call the error co-variance in the process and it can be because any error during our process and it is actually very hard to be calculated because it can be due to any error in the environment which we can't measure. So as you can see there are 2 things which we need to give an initial condition for which are X_{k-1} and P_{k-1}.

Now for the correction phase. So we proceed as follows first we calculate the Kalman gain K_k using the system error co-variance P_k, The sensor error co-variance matrix R which can be easily calculated if we know the error in our sensor reading and finally H which is Sx1 matrix where S is the number of sensors used. Then we calculate the new estimated state X_k and the new system error co-variance P_k. Finally we re-plug the new estimated state and the new system error covariance into the prediction state and keep iterating as you can see in fig 3 To fuse the data from more than one sensor using kalman filter we used 2 approaches. Basically we first have to take the readings from each sensor to get 1 reading and this is because we can't trust just one reading we have to collect more than one reading to be able to get more accurate results from our fusion. The first approach which we used is that we divide the readings into groups. So for example if we have 10 readings from each sensor we divide them into 10 groups where the first group will have the first reading from each sensor so if i am using 2 sensors each group will consist from 2 points. Then we apply Kalman filter for each group and we take the last point from each group and bring them all together and so we will have 10 new points which represent our fusion.

The second approach which we used to fuse the data using Kalman filter is actually different from the first approach because in the first approach we don't change the dimensions of the system meanwhile in our second approach we will increase the dimensions of some of our variables. To start with we don't change our State matrix X_k because it's the same state which we are measuring and due to that the P_k, A and the Q matrices will not change their dimensions also. Regarding the correction phase here is where we will have the change of dimensions. First of all our sensor error co-variance matrix will change it's dimension to be a square matrix and it's size is equal to the number of sensors and it will change to be a diagonal matrix which consists of the co-variances of the sensors and you can see how the R will look like if we have 2 sensors in equation 10

$$R = \begin{bmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \end{bmatrix} \tag{7}$$

Also the H will change it's dimension to be Sx1 where S is the number of sensors used which in our case will look like this

$$H = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \tag{8}$$

Due to the change of dimensions of both the R and the H the K_k will change it's dimension to be Sx1 to define 2 kalman gains 1 for each sensor

$$K_k = \begin{bmatrix} K_{k1} \\ K_{k2} \end{bmatrix} \tag{9}$$

Finally the Z_k which will be again Sx1 matrix with the readings from the sensors at the time k.

$$Z_k = \begin{bmatrix} Z_{k1} \\ Z_{k2} \end{bmatrix} \tag{10}$$

C. Proposed approach

Our proposed approach was to use pre-filtering in combination of data fusion so we take the data from the sensors we filter each sensor alone then we fuse the filtered data together. We had 3 sets of experiments:

- Kalman filter for the noise filtering and Kalman filter for the data fusion.
- Only Bayes filter for the data fusion with no filtering.
- Kalman filter for the noise filtering and Bayes filter for the data fusion.

IV. Tests and Simulations

In this section we will discuss the simulations and results from our system and compare the results between each test and simulation. The coming sections shows how we used those data fusion techniques. We first used Kalman filter for filtering the data coming from the sensors and then we used the Bayes filter alone to fuse the data from both sensors and finally we used both the Kalman filter and the Bayes filter. The Kalman filter for filtering the noise and the Bayes filter to fuse the data. Due to we were using regular surfaces in the simulations we decided also to test how the sensors will react when we use irregular surfaces which is not a good reflector of the ultrasonic waves.

A. Kalman filter

Here we used the Kalman filter for noise filtering and Data fusion.

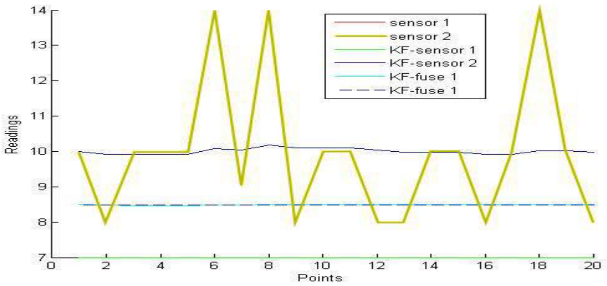

Fig. 4. KF test 1

In this figure you can see the reading from the sensors but you can't see the readings from sensor 1 because it was overlapped by it's Kalman filter. And you can see the Kalman filter of each sensor and the fusion of both sensor. So basically what we did here is that we filter the noise from each sensor alone as you can see in fig 4 then we use the filtered data to fuse them together and we used the 2 approaches of data fusion which we mentioned in the last section. Here we can see how the sensors can react you will see one sensor which is very stable in it's reading and the other is actually very noisy and this is mainly because of the Cross talk effect because the sensors were fixed very near to each other. Given the fact that each sensor work in a better way when it comes to the noise when they are alone. The ground truth was 8.

B. Bayes filter

In the coming figures you can see the readings that we took from each sensor in a figure then in the following figure you can see the Bayes filter of sensor1, sensor2 and the fusion between the 2. The true value is 8.

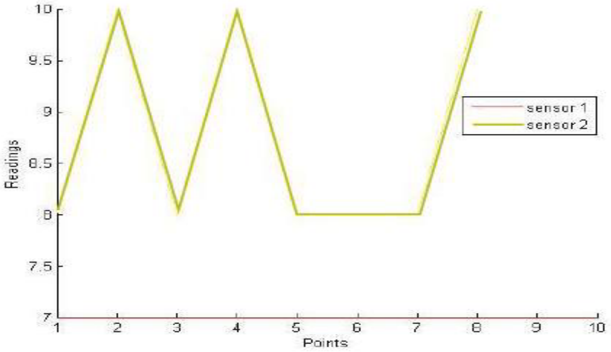

Fig. 5. BF test 2 measurements

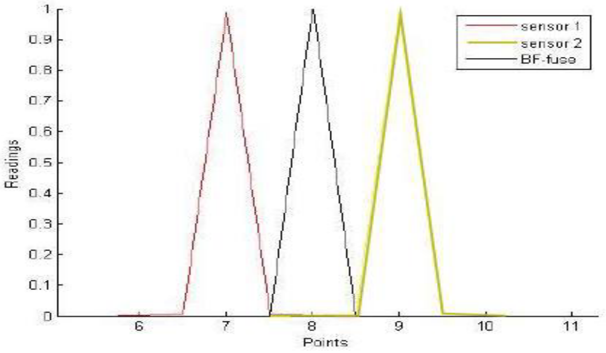

Fig. 6. BF test 2

C. Kalman and Bayes filters together

In this section we used both the Kalman and the Bayes filters together so we used the Kalman filter first to filter the noise from the sensors then the Bayes filter to fuse the filtered data.The true value is 8.

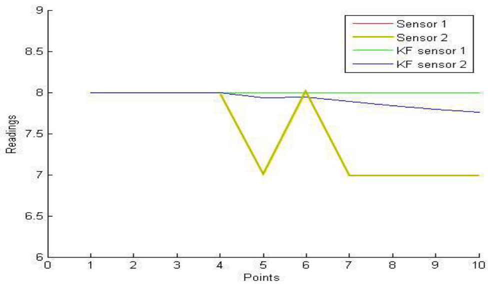

Fig. 7. KF noise filtering

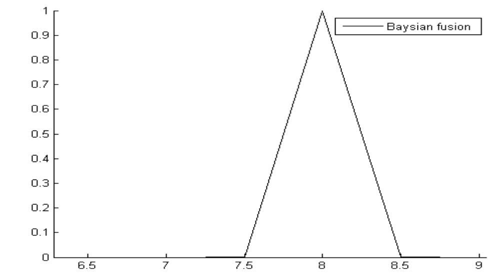

Fig. 8. BF fusion

D. Irregular surfaces

In the last figures the readings was taken with the sensors measuring the distance from a regular rectangular surface which make it good reflection surface. But in the coming figures you can see how the sensors act when we put instead of a regular surface a rolled surface you can see how the readings of both of the sensors are far away from each other and the noise at both of the sensors increased. Again we applied the past combinations of data fusion and noise filtering techniques.

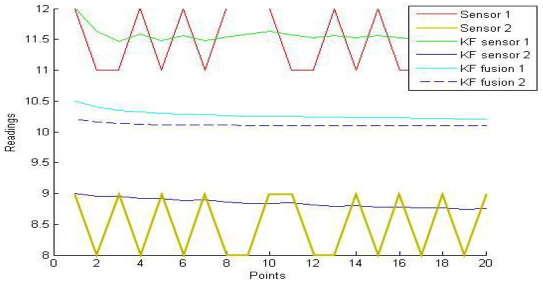

Fig. 9. KF rounded surface

In this figure we used the Kalman filter to be our noise filter and our data fusion method. The ground truth is 8.

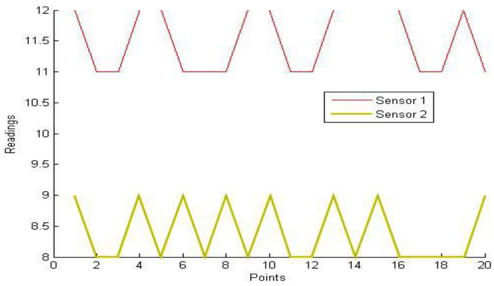

Fig. 10. BF rounded surface measurements

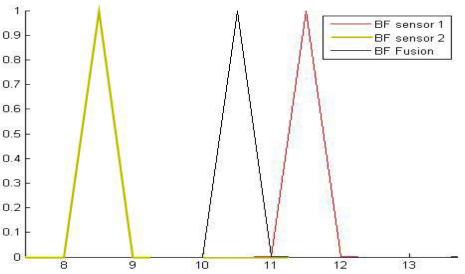

Fig. 11. BF rounded surface

Here you can see the Results when we used the Bayes filter again it is far away from the true value.

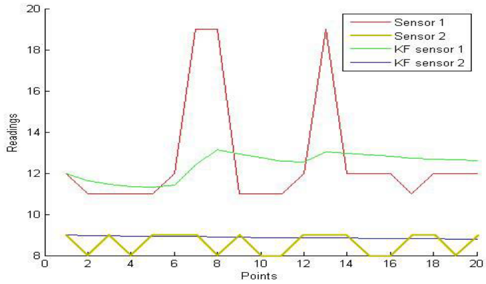

Fig. 12. KF rounded surface noise filtering

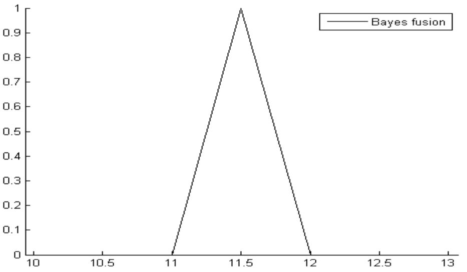

Fig. 13. BF rounded surface fusion

Here you can see that the Kalman filter of the first sensor was a little bit struggling to stabilize the data coming from sensor 1 due to that the readings as very unstable and again the value from the Bayes filter is far from the true value which is 8.

V. RESULTS

As you can see in the past section The Kalman filter was very successful in filtering the noise from the sensors readings and both the Kalman and Bayes filters where able to achieve good results when it come to Data fusion of the 2 sensors. So we can see that the use of more than 1 sensor to measure the same value and then use data fusion techniques to fuse the data from the 2 sensors together gets more accurate and precise data. So thanks to GOD we were able to achieve our goals. We were able to reduce the uncertainty in the readings coming from the sensors, reduce or even eliminate the sensor noise in some cases and finally we were able to reduce the effect of the Cross Talk on the sensors readings.

Also we can easily modify how much we trust the readings coming from each sensor as for example some of the graphs sensor 2 was giving noisy readings most of the time so we can choose to trust sensor 1 more than sensor 2 to get more accurate results but we choose in this study to trust the 2 sensors equally.

But as you can see when it comes to irregular surfaces the sensors start to give readings which is actually very far from the true value. Also the probabilistic data fusion techniques in general are very sensitive to outliers which is caused by the environment like the one which we have here which is dealing with surface with minimum reflecting capabilities as the rolling surface. And as you can see in our tests the data fusion techniques which we were using were able to give very precise data but not accurate at all.

VI. Conclusion and Future Work

A. Conclusion

To sum up what we did in this study. We first defined our problem which is the uncertainty and noise in the sensors readings this is where we got our motivation to start this study. Then we searched for the types of data fusion which can be suitable for our aim. And we ended up choosing the probabilistic data fusion because it was the most suitable to use in order to reduce the data uncertainty. And then we get to choose which types of probabilistic data fusion to work with and we decided to choose more than one way in order to compare the results. So we chose the Kalman filter and the Bayes filter. Then we modeled our system and used the data fusion methods in a different arrangements and also we used those method not only for data fusion but also for noise filtering. We used the Kalman filter alone first then the Bayes filter and finally we used both the Kalman filter and the Bayes filter. The Kalman filter for the noise filtering and the Bayes filter for the data fusion. We used a regular rectangular surface and irregular rolled surface to be our obstacle which we measure the distance from and we compared the results between them. The results which we achieved were very promising. We were able to filter the noise from the readings successfully and our data fusion techniques were able to successfully fuse the data from the sensors and achieve good results which we discussed in the last section. Also we talked about the limitations that we faced and the tests which we used to see the effect of those limitations. And the limitations of the of the data fusion method used itself like it being very sensitive to outliers. Finally all the 3 arrangements used to test the data fusion techniques proved to work very well and you can choose to work with any one of them depending on your system.

B. Future work

My recommendations for future work on this topic is

- Use other types of sensors like the laser range finder sensor
- Use different types of data fusion like the fuzzy logic and compare the results to mine
- Try to use different reflection surfaces with different shapes and materials
- Use more than 2 sensors at the same time which from my opinion will improve the results significantly
- Study the computational cost of those algorithms and try to improve it

Acknowledgments

We thank GOD for giving us the strength to complete this project. And we thank all the RAS lab family for the continuous support.

References

[1] R. P. Mahler, *Statistical multisource-multitarget information fusion*. Artech House, Inc., 2007.

[2] G. Welch and G. Bishop, “An introduction to the kalman filter. 2006,” *University of North Carolina: Chapel Hill, North Carolina, US*.

[3] S. J. Julier and J. K. Uhlmann, “New extension of the kalman filter to nonlinear systems,” in *AeroSense’97*. International Society for Optics and Photonics, 1997, pp. 182–193.

[4] L. D. Stone, T. L. Corwin, and C. A. Barlow, “Bayesian multiple target tracking,” 1999.

[5] A. Doucet, N. De Freitas, and N. Gordon, *An introduction to sequential Monte Carlo methods*. Springer, 2001.

[6] B. A. Berg, “Markov chain monte carlo simulations and their statistical analysis,” 2004.

[7] D. Crisan and A. Doucet, “A survey of convergence results on particle filtering methods for practitioners,” *Signal Processing, IEEE Transactions on*, vol. 50, no. 3, pp. 736–746, 2002.

[8] R. Siegwart, I. R. Nourbakhsh, and D. Scaramuzza, *Introduction to autonomous mobile robots*. MIT press, 2011.

[9] B. Khaleghi, A. Khamis, F. O. Karray, and S. N. Razavi, “Multisensor data fusion: A review of the state-of-the-art,” *Information Fusion*, vol. 14, no. 1, pp. 28–44, 2013.