Sensor Fusion: An Application to Localization and Obstacle Avoidance in Robotics Using Multiple IR Sensors

Rahul Sharma, Honc Daniel, and František Dušek

Department of Process control, Faculty of Electrical Engineering and Informatics,
University of Pardubice, Pardubice, Czech Republic,
rahul.sharma@student.upce.cz,
{daniel.honc,frantisek.dusek}@upce.cz

Abstract. Sensor fusion brings the advantage of combining data from various sensors and there by generating a more accurate prediction or estimation of data. Over dependency of sensor and estimation from unreliable data are the most challenging tasks in mobile robotics. In this paper, a framework of sensor fusion technique is presented. The data from the multiple sensors are fused together and the parameters and crash time are estimated. The experiment results show that the sensor fusion technique provides solution to over dependency of sensor and problems with estimation of data from unreliable data. The technique finds application in obstacle avoidance and localization of mobile robots.

1 Introduction

For an autonomous mobile robot, the most challenging task is the perception of the environment using sensors. In most of the cases the robots tend to depend heavily on dedicated sensors, which often tend to be unreliable. So, for autonomous robots in fault tolerant applications, sensor fusion techniques help to perceive the environment in a better way.

Sensor fusion is a technique by which the data from multiple sensors are fused together to get an exact information. The fusion of sensor data can be from redundant sensors or complementary sensors. The simplest case of sensor fusion is to combine the data from the sensors, by averaging the sensor reading, if all the sensors have same belief. If the sensors have different belief, a weighted average would help to some extent. However, this simple combining of sensor data would not work, when the system is complex or requires a precise data.

A range of sensor fusion techniques are reviewed in [1,4,6]. Over the years, many techniques of sensor fusion have been emerged. Kalman Filter and Extended Kalman Filter are the most researched technique in sensor fusion for robotic navigation [2,3].

This paper focuses on a simple sensor fusion technique of redundant sensors (IR range finder), in which data from multiple sensors are fused together to determine the three parameters, namely perpendicular distance from center of robot to the wall, distance to the wall and angle between the horizontal axis of robot and obstacle wall. Another important parameter which needs to be considered in applications like ob-

I. Zelinka et al. (eds.), *Nostradamus 2014: Prediction, Modeling and Analysis of Complex Systems*, Advances in Intelligent Systems and Computing 289,
DOI: 10.1007/978-3-319-07401-6_38, © Springer International Publishing Switzerland 2014

stacle avoidance is the crash time, the time left before crashing in to the wall or obstacle. The crash time can be predicted from the current velocity and acceleration of robot and distance to the wall from the robot. Some application of the proposed method is also presented.

The paper is organized as follows: section 2 describes the calibration of the IR range finder, followed by mathematical modelling of the system in section 3. The design of the proposed system is stated in section 4, followed by experimental results in section 5 and conclusion in section 6.

2 Characterization of IR Sensor

SHARP IR sensor (GP2Y0A02) is chosen as the range sensor. The output voltage of the sensor is a nonlinear function of the distance between the object and the receiver. The distance value depends on a non-linear way from the sensor analogical value. The best function to fit the sensor curve is given by the expression:

$$y = \frac{Ax}{x+B} + C \tag{1}$$

Where x is the analog voltage output from the sensor, y is the distance to the object from the sensor.

Fig. 1 shows the real acquired values and the result of best fit that makes it possible to compute the distance, measuring the analogical Sharp's value. The best relation was obtained optimizing, on MATLAB, the sum of the quadratic error between the real acquired values and the fitting curve. The parameters A,B and C are optimised by MATLAB *lsqcurvefit* function.

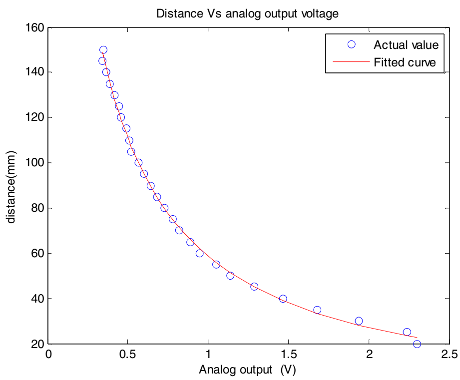

Fig. 1. Plot of distance Vs ADC volt after curve fitting by *lsqcurvefit* function in MATLAB (Comparison between actual data and fit data)

3 Mathematical Modelling

x_k and y_k are the points on the line (wall), where the IR sensor beam gets reflected. The parameters of interest are d, $\emptyset$ and d_k, where d is the distance from the robot to the wall, d_k, is the perpendicular distance from the origin of the robot to the wall and $\emptyset$ is the angle between horizontal axis of robot and axis parallel to the wall.

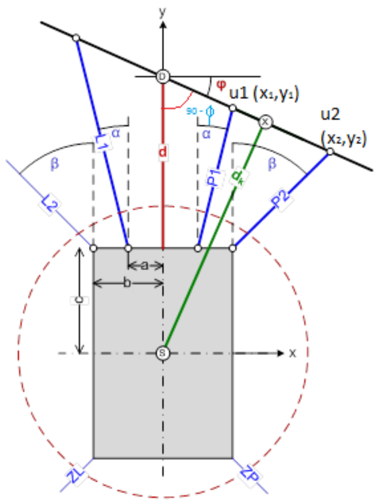

Fig. 2. Layout of sensor deployment

α and β are the angle at which the IR sensors are mounted on the robot. p_k is the depth to the wall from the robot measured by sensor *k*. *a,b* and *c* are the distances from the axes of robot to the sensors as shown in Fig. 2.The parameters can be derived from the following expressions.

$$\begin{bmatrix} x_i \\ x_j \\ y_i \\ y_j \end{bmatrix} = \begin{bmatrix} sin\alpha & 0 & 0 & 0 \\ 0 & sin\beta & 0 & 0 \\ 0 & 0 & cosa & 0 \\ 0 & 0 & 0 & cos\beta \end{bmatrix} * \begin{bmatrix} p_i \\ p_j \\ p_i \\ p_j \end{bmatrix} + \begin{bmatrix} a \\ b \\ c \\ c \end{bmatrix} \tag{2}$$

$$\begin{bmatrix} y_i \\ y_j \end{bmatrix} = \begin{bmatrix} x_i & 1 \\ x_j & 1 \end{bmatrix} * \begin{bmatrix} m \\ n \end{bmatrix} \tag{3}$$

This is in the form of,

$$Y = F * X \,, \tag{4}$$

Applying least square estimation (LSE) method to eq. (4)

$$X = (F^T F)^{-1} F^T Y \tag{5}$$

$$d = n - c \tag{6}$$

$$\emptyset = \tan^{-1}(m) \tag{7}$$

$$d_k = (d + c) * \sin(\Pi/2 - \emptyset) \tag{8}$$

From each pair of sensors the corresponding parameters can be calculated and these values can be represented as p_{ijk}, where *i,j,k* represents the parameter (d, d_k or $\emptyset$), first sensor and second sensor respectively. The variables *m* and *n* represents slope and intercept respectively.

4 Design of Sensor Fusion System

Fig. 3 shows the basic design of sensor fusion system adopted in the paper. The first step is preprocessing of the signal acquired from the sensor. In this stage, a series of sample are averaged to get an initial rough data. Using this data (p_i), the parameters (p_{ijk}) are calculated using eq. (1) – (8). These set of parameter have to be validated, for knowing the reliability of the sensors. At the data validation stage, the data is validated and a weighing factor v_{ijk} is calculated. The following sections will explain the stages in detail.

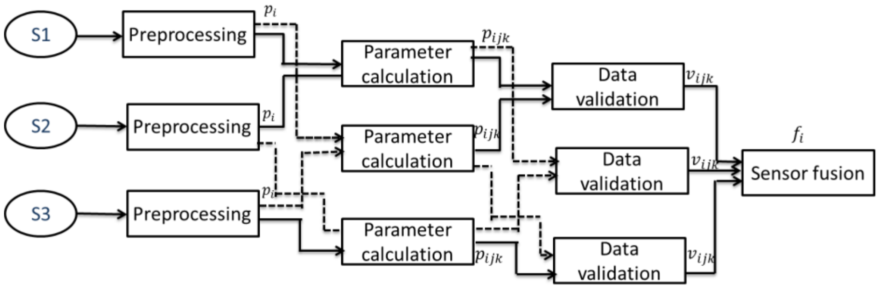

Fig. 3. Sensor fusion system

4.1 Data Validation

Data from the sensors has to be validated before proceeding to sensor fusion. The data from the sensor is only allowed to fuse, if it confirms certain criteria. The criteria are correlation coefficient and closeness coefficient. Correlation coefficient is given by the expression,

$$Cor(j,\mathrm{k}) = \frac{N\sum_0^N P_{ijk}P_{ikj} - \sum_0^N P_{ijk}\ \sum_0^N P_{ikj}}{[N\sum_0^N P_{ijk}{}^2 - (\sum_0^N P_{ijk})^2]^{\frac{1}{2}}\ [N\sum_0^N P_{ikj}{}^2 - (\sum_0^N P_{ikj})^2]^{\frac{1}{2}}} \tag{9}$$

$Cor(j,\mathrm{k})$ is the correlation of parameter P_i , calculated from sensor j and sensor k and N is the number of samples. Closeness coefficient [5] is given by the expression,

$$\gamma = 1 - \frac{|P_{ij} - P_{ji}|}{Closeness} \tag{10}$$

Where $Closeness$ is the parameter which allows choosing the data which are closer.

When the new data (parameters) falls in a range, say magnitude of correlation coefficient is above 0.75 and closeness coefficient is less than 0.8, the data is selected. Otherwise the data is rejected and not chosen for data fusion. This allows the system to choose only reliable sensor data by comparing with all the other sources. A weighing factor, v_{ijk} gives the weightage of each parameter, so that the corresponding parameter can either selected or rejected. This factor is calculated by the following expression

$$v_{ijk} = \begin{cases} 0, & if\ any\ of\ the\ criteria\ fails \\ 1, & if\ both\ the\ criteria\ are\ satisfied \end{cases}$$

v_{ijk}, will help to choose the most reliable parameters for fusion, calculated from different sensors.

4.2 Sensor Fusion

The parameters which confirms the criteria of data validation are selected and fused together [7] by the following expression

$$f_i = \sum_{i=0,j=0,k=0}^{i=3,j=n,k=n} \frac{\sigma_{kj}^2}{\sigma_{jk}^2 + \sigma_{kj}^2} p_{ijk} * v_{ijk} \tag{11}$$

Where is the final parameter, σ_{kj}^2 is the variance and p_{ijk} is the parameter i, calculated from sensor j and k.

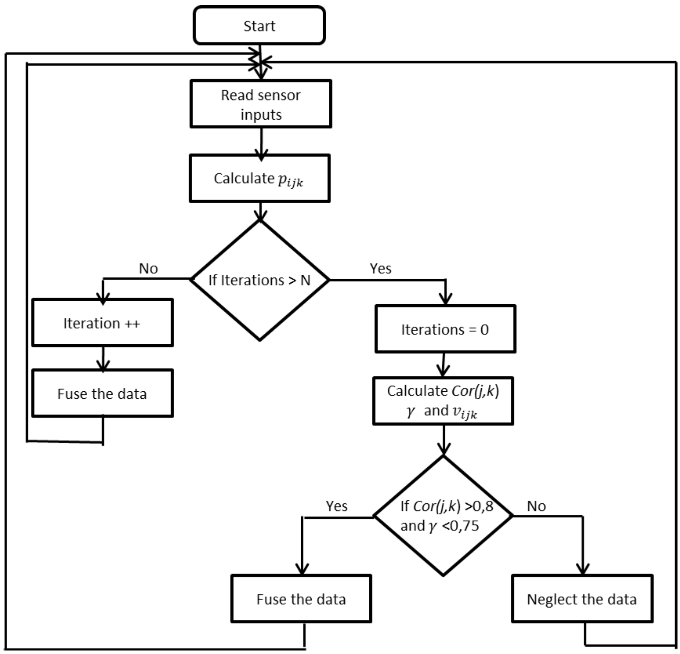

Fig. 4. Flow chart of sensor fusion system

Fig. 4 illustrates the process, step by step. For making the system faster, a variable N (period), is introduced to validate data on regular intervals, thereby avoiding the sensor validation process in each iteration.

4.3 Prediction of Crash Time

For autonomous robots, the time to crash is an important criterion. If robot can predict the crash time, it could avoid the obstacle by making the appropriate decision. The crash time can be predicted from the parameters estimated by sensor fusion. The crash time T, derived from basic laws of motion, is given by the expression

$$T = \frac{-v \pm \sqrt{v^2 - 2a(\frac{Z}{2} * \tan \emptyset + d)}}{a} \quad (12)$$

Where *v*, *a* and z represents velocity of robot, acceleration of robot and width of the robot respectively.

5 Experimental Results

Experiments were conducted using Arduino Due board and Sharp GP2Y0A02, IR range finders. Three IR range sensors were fixed on the front side of the robot with a=0, b=10, c=10, α=0 and β=5̊. Three sets of parameters were calculated from the sensor readings. As a first step the IR sensor data were preprocessed by averaging a sample set of 10. This data was then validated by using the correlation coefficient threshold as 0.75 and closeness coefficient as 0.8. If the data fall in the range of the criteria, it was selected to fusion. Three sets of data were then fused together as explained in section sensor fusion.

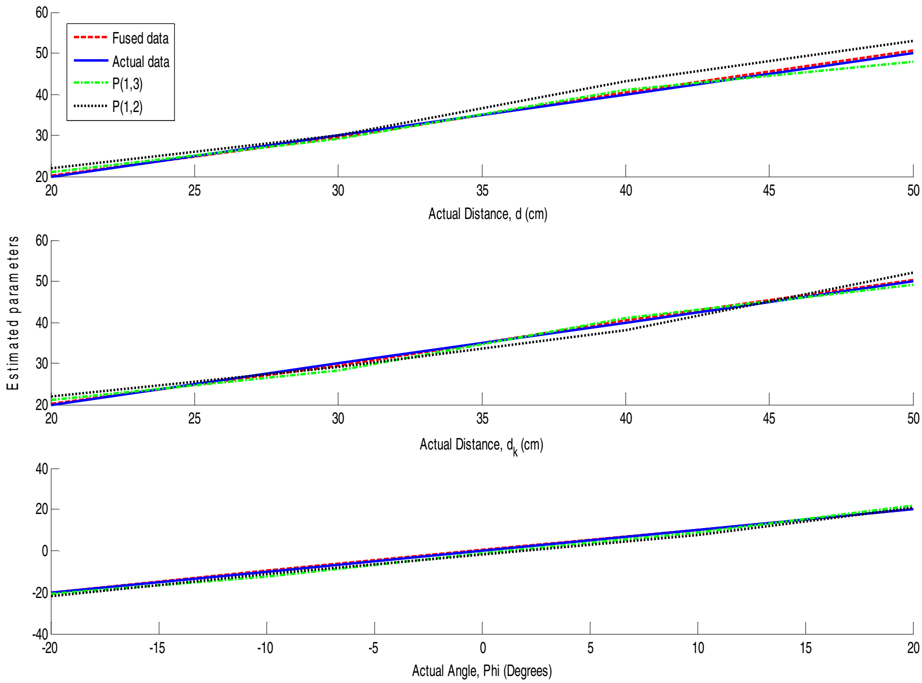

Fig. 5. Comparison of fused and non-fused data a) parameter d, b) parameter dk, c) parameter ∅, where P(1,2) is the parameters calculated from sensor 1 and sensor 2, P(1,3) is the parameters calculated from sensor 1 and sensor 3

Several experiments were conducted by varying distances and angles. Fig. 5 shows the variation from actual data to fused and non-fused data from a series of experiments. The fused data tends to follow more closely to actual data, compared to non-fused data P(1,2) and P(1,3), which had larger variations.

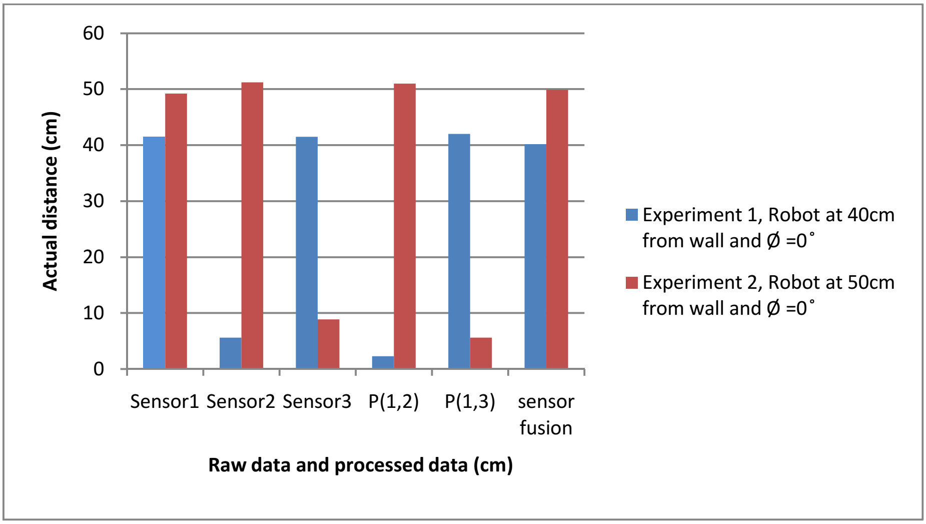

Fig. 6. Experiments with faulty sensors (Actual vs Observed and calculated variables)

Experiments were also conducted by making one or more sensors faulty intentionally. Fig. 6 shows that even if sensor data is unreliable, the sensor fusion system is still able to estimate an accurate data. For example, in experiment 1, sensor 2 was made faulty and by fusing the data from all the three sensors, the system could reject the data from the sensor, identifying the reliability of sensor through data validation process, and estimate the data which was closer to the actual data. Crash time was also predicted by measuring current velocity and acceleration of robot.

It was noticed that, the proposed system performed better even in case of unreliable data from faulty sensors. Furthermore, the treat of over dependency of dedicated sensors was also avoided to some extent, by making use of multiple sensors instead of dedicated sensor. This will allow the robots to perform better when working in unknown dangerous environments. In another words the system works better where there is a requirement of fault tolerant system.

The proposed system finds two applications in robotics. Firstly, in case of decision making during obstacle avoidance, where the parameter $\emptyset$, can be used to take a right decision. For example, if the $\emptyset$ is negative, the robot must take a LEFT turn and vice versa. The second application is localization of robot. The $\emptyset$ will give a better orientation estimate with the help of multiple sensor input. The approximation from multiple sensor input will give more accurate estimate of orientation of the robot.

6 Conclusion

The proposed system is modeled, simulated and tested in various environments. The results show that the system performs better by making use of multiple sensor inputs. The fusion of multiple sensor inputs allows the robot to work in fault tolerant applications. The system finds applications in obstacle avoidance decision making and localization (orientation estimate) of robots. When combined with absolute positioning sensors (GPS) or dead-reckoning sensors (IMU), the can be further extended to real world applications in robotics.

Acknowledgments. This research was supported by Institutional support of The Ministry of Education, Youth and Sports of the Czech Republic and by the project "Support for internships and professional activities by innovation of tertiary education at DFJP and FEI University of Pardubice, Reg. No: CZ.1.07/2.4.00/17.0107" in Modern control methods team - development and application of predictive control methods using artificial intelligence.

References

1. Elmenreich, W.: A review on system architectures for sensor fusion applications. In: Obermaisser, R., et al. (eds.) SEUS 2007. LNCS, vol. 4761, pp. 547–559. Springer, Heidelberg (2007)
2. Foxlin, E.: Inertial head-tracker sensor fusion by a complementary separate-bias Kalman filter. In: Proceedings of the IEEE 1996 Virtual Reality Annual International Symposium, pp. 185–194. IEEE (March 1996)
3. Jetto, L., Longhi, S., Venturini, G.: Development and experimental validation of an adaptive extended Kalman filter for the localization of mobile robots. IEEE Transactions on Robotics and Automation 15(2), 219–229 (1999)
4. Kam, M., Zhu, X., Kalata, P.: Sensor fusion for mobile robot navigation. Proceedings of the IEEE 85(1), 108–119 (1997)
5. Naji, H.R., Weir, J., Wells, B.E.: Applying the multi-agent paradigm to reconfigurable hardware: a sensor fusion example. In: Second International Workshop on Intelligent Systems Design and Application, pp. 207–212. Dynamic Publishers, Inc. (January 2002)
6. Nicosevici, T., Garcia, R., Carreras, M., Villanueva, M.: A review of sensor fusion techniques for underwater vehicle navigation. In: OCEANS 2004, MTTS/IEEE TECHNO-OCEAN 2004, vol. 3, pp. 1600–1605. IEEE (2004)
7. Panich, S., Afzulpurkar, N.: Sensor Fusion Techniques in Navigation Application for Mobile Robot. In: Sensor Fusion-Foundation and Applications, pp. 101–120 (2011)