Multiple Sensor Fusion in Mobile Robot Localization

Karthick Srinivasan[1]
[1]Electrical & Computer Engineering, Dalhousie University, Canada
Karthick@dal.ca

Jason Gu[1,2]
[2]Shenzhen Institute of Advance Technology, Chinese Academy of sciences, China
Jason.gu@dal.ca

***Abstract*— The fusion of multi-sensory information plays a key role in driving a mobile robot over a fixed lane, object recognition, obstacle avoidance, self localization and path planning. To learn the environment using multi-sensory information, we need both an accurate sensor model and a reasonable sensor fusion methodology. In this paper a novel technique is explained combining data's from Ultrasonic sensor, Encoder and Gyroscope. Encoder is often utilized for the position estimation by accumulating the number of times the wheel rotates. Since the B21r robot relies only on encoder data information for localization, the motivation behind this technique is to reduce the wheel-drift that occurs in encoder due to slippage error and bumps on the path that causes the robot to move in a elliptical path when intended to move in a circular path. The B21r mobile robot has forty eight ultrasonic sensors, twenty four at the base and twenty four at the body of the robot. The ultrasonic sensors are used to develop an obstacle avoidance algorithm based on Virtual Force Field (VFF) technique. The algorithm is then combined with a rule based algorithm for the inertial sensors namely encoder and gyroscope, which switches the control back and forth between the encoder and the gyroscope depending on the slippage error caused in the encoder. The simulation results show the path of the robot with the conventional encoder data alone and then with the algorithm implemented. The results and future expansion of the study and the merits of the algorithm are discussed.**

Keywords- Mobility, Gyroscope, Encoder, Ultrasonic sensor, Virtual Force Field

I. INTRODUCTION

Sensor fusion is a process of integration and extraction of desired information from two or more sensors. In other words it's a process of combining multiple sensors and to provide more useful information than the sum of individual sensors. Integration or fusion of data from multiple sensors improves the accuracy of applications ranging from target tracking and battlefield surveillance to non-defense applications such as industrial process monitoring and medical diagnosis [1]. Fused sensor data's from various sensors offers several advantages when compared to data from a single sensor.

- First advantage of combining data's from several identical individual sensors will result in improved estimate of the target position and velocity.
- Secondly it improves the relative position or motion of the sensors for observation.
- Third advantage by using multiple sensors include improvement is system observability.
- And finally Sensor fusion increases the system's stability, reliability and compensates for the inaccuracies and limited operation of individual sensors.

II. EASE OF USE

The classification of sensor fusion in mobile robotics is based on the applications and the purpose they are used in. The two main classifications are based on 1. Based on fusion of data's, 2. Based on type of sensors used in fusion.

The classification based on the fusion of data's is sub-divided in to three categories namely,

i. Direct level fusion of data.

In this method the information obtained from different sensors measuring same physical quantity like two or more visual sensors or acoustic sensors are combined directly and the information is used for further process. If the information obtained from the sensors are non-commensurate then the data must be fused by any one of the following two techniques. Fig 1.1 shows the schematic view of Direct level fusion.

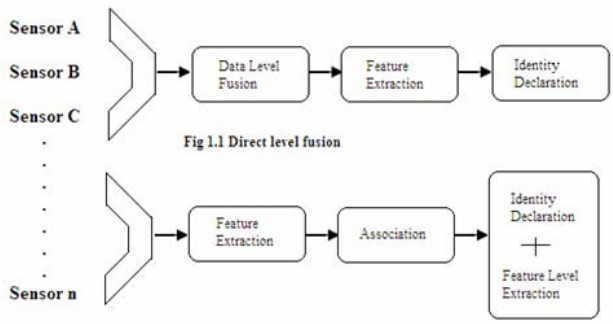

Figure 1. Feature level Extraction

ii. Feature level fusion of data.

This technique involves the extraction of representative features from the sensor data like object recognition. In feature level sensor fusion, features are extracted from the information obtained from the sensors; these data's are then fused or concatenated to one single feature. This one single feature is then further processed using various techniques like objects recognition, pattern recognition to extract the

image. Fig 1.2 shows the block diagram of Feature level fusion of data.

iii. Decision level fusion of data.

The data's obtained from individual sensors are processed to obtain their location, identity and attributes. Then each data's are processed to obtain high level interference which is subsequently combined.

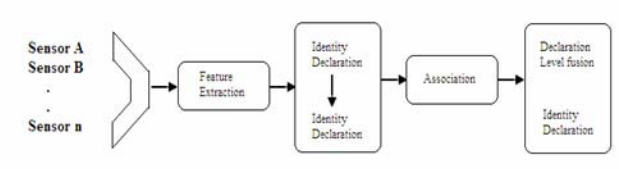

Figure 2. Decision Level Sensor Fusion

The classification based on the types of sensors used is 1.Inertial sensor fusion 2. External sensor based fusion. The inertial sensor fusion is based on sensors like odometer, gyroscope and compass. These sensors provide the position and orientation of the robot based on the movement. The external sensor based fusion depends on the infrared, ultrasonic, laser etc. These sensors measure the position of the robot based on the distance from the obstacle or the object near by. These sensors are mostly used in navigation technique.

III. SENSORS

A. Encoder

Encoder is used by most of the mobile robotic systems to navigate and to position. Encoder is a device used for measuring the distance traveled by the robot or any other vehicle. Navigation technique developed using encoders are widely used in relative position system, since encoders are inexpensive, easy to implement and provides good short-term accuracy.

Encoders and odometer are not very exact techniques to calculate the position and heading. Encoder errors influence both heading and position errors. In encoder navigation system even a small error will result in a serious position error. The errors in encoders can be classified as systematic errors and non-systematic errors.

Systematic errors are caused due to the inaccurate wheel size or improper positioning of the wheels or the distance between the wheels, limited encoder resolution and limited encoder sampling rate. Non-systematic errors are due to the surroundings like uneven floor surface, wheel slippage, drift.

B. Ultrasonic Sensor

Ultrasonic sensors are mainly used for range detection because it has advantages of less expensive, good range information based on time of flight, compatible, easy to install and it is an external sensing systems for mobile robot locations. These sensors have a measuring range of 0.3-10.7m [2].

Ultrasonic sensors work based on simple principle: the time elapsed between the transmission and reception of the ultrasonic wave is proportional to the distance from the sensor to the obstacle. The measurements by these ultrasonic sensors suffer from drawbacks which limit the usefulness of these sensors [3]. Wide beam- width which yields uncertainties in the measurement [4], variation in speed of propagation.

The B21r robot used to experiment in this robot has 48 ultrasonic sensors spaced at an angle of 15 degrees between them. The obstacle avoidance algorithm developed as a portion of this research uses 24 ultrasonic sensor at the base of the robot.

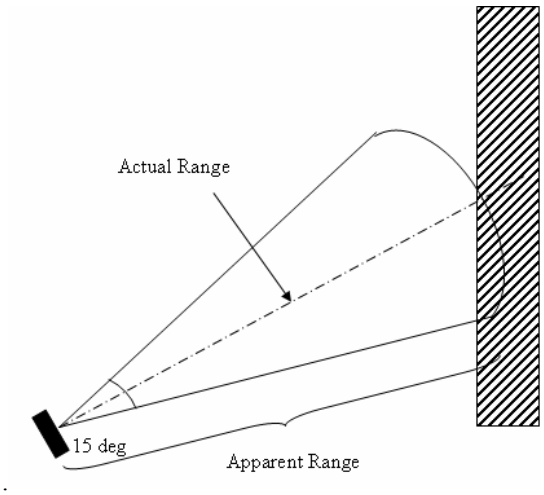

Figure 3. Formation of cone shaped sonar arc

C. Gyroscope

Gyroscope is a device used for measuring orientation, based on the principle of angular momentum. The use of gyroscope has become more attractive for mobile robot applications as it measures the angular rate without contacting any surface or parts of the robot.

The gyroscope used in this research is IMU 300CC from Crossbow Technologies. The IMU 300CC is an intelligent six-degree-of-freedom (6DOF) Inertial Measurement Unit (IMU). The device uses a high performance Digital signal processor to provide outputs that are compensated for deterministic error sources within the unit. Internal compensation includes offset, scale factor and alignment. The IMU measures linear acceleration along with three orthogonal axes and rotation rates around three orthogonal axes. The three angular rate sensors consist of vibrating ceramic plates that utilize the corolis force to output angular rate independently of acceleration. The serial interface is standard RS232, 38400 baud, 8 data bits, 1 start bit, and 1stop but, no parity and no flow control.

IV. ALGORITHM

The algorithm developed for this technique is carried out in steps as explained. First the obstacle avoidance algorithm is developed which is based on Virtual Force Field technique. The next step is to include the encoder in the algorithm, an algorithm is developed using velocity approach. The final step in developing the algorithm is developing an algorithm that reads only the RAW –axis data from the Gyroscope. After the algorithm for individual

sensors are developed the sensor- algorithms are fused and a rule-based algorithm is coded, which enforces the encoder readings to shift to gyroscope and back when needed.

A. *First Step – Obstacle Avoidance*

Many robotic applications require the robot to navigate safely through the environments which are known or unknown. In this method the obstacle exerts forces onto the mobile robot as a result the forces tries to steer the robot away from the obstacles. The Virtual Force Field is a combination of potential field method with a certainty grid. In this method the robot work area is represented by a 2-D square matrix in which the robot is assumed to be in center at any given time.

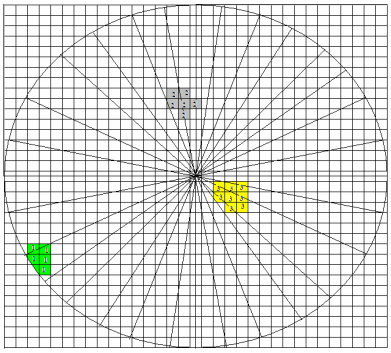

Figure 4. Grid formation of the square matrix, values shown in the figure represents the value generated based on the presence of obstacle

A 2-D histogram grid is used as a model to represent the obstacle, the work area moves with the robot with square region of cells in the grid. Each cell exerts a value based on the size, shape and the presence of obstacle. Each occupied cell creates a repulsive force to the robot making it steer away from the cell. The force exerted by the obstacle varies depending on the position of the obstacle; if the obstacle is close to robot it exerts huge force to drive the robot away from it as shown in the figure. The magnitude of the force generated is directly proportional to the value that is been generated in the cell as shown in the figure The size of cell depends on the sensors, if the sensors are highly precise and the speed of propagation is high then the size of the cell can be smaller if not the cells should be larger to have better resolution. As a result the force exerted by the cell is given by the equation 4.1, the equation also implies that the force is inversely proportional to the square of distance between the obstacle and the robot

$$F_{(x,y)} = \frac{F_k C_{(x,y)}}{d^2_{(x,y)}}\left[F_{rx} + F_{ry}\right] \quad 4.1$$

$$F_{rx} = X_{grid} - X_{current} \quad 4.2$$

$$F_{ry} = Y_{grid} - Y_{current} \quad 4.3$$

Where,

$F_{(x,y)}$ = Repelling force.

$d_{(x,y)}$ = distance between the robot and the obstacle.

$X_{current}$, $Y_{current}$ = Current position of the robot.

X_{grid}, Y_{grid} = Co-ordinates of the grid.

B. *Second Step – Encoder Fusion method*

The B21r robot has in-built four raw encoders at the base and does motion control based on encoder feedback [5]. The controller combines the encoders into two virtual axes namely, translate axis and rotational axis which are sent to Mobility. Mobility is an object oriented tool used as integration software in B21r robot. The information obtained from translate axis and rotational axis is converted to meters/second and radians/second by Mobility.

The algorithm for B21r robot is developed using C++ Mobility Class frame work model. This Class frame work is used to develop Mobility Core object Model which is defined using the CORBA interface definition language. Different object models are used for the encoder program like An ActutatorState object, TransformState object, FVectorState object, PostionState object. The ActutatorState object represents the B21r drive system which represents the state of actuator, The FVectorState object is used to convert the sensor data into meters is used to represent the velocity of the B21r robot and the TransformState object provides information the current location of the robot. The discrete form of inertial-encoder navigation equation is

$$X_{k+1} = \sum_{i=1}^{k}(X_i + \Delta X_{i-1}) + IND * \cos(\theta_i + \theta_0)/2 \quad 4.4$$

$$Y_{k+1} = \sum_{i=1}^{k}(Yi + \Delta Yi - 1) + IND * \cos(\theta_i + \theta_0)/2 \quad 4.5$$

$$\theta_{k+1} = \sum_{i=1}^{k}(\theta_i + \Delta\theta) \quad 4.6$$

C. *Third Step – Gyroscope*

Gyroscopes are of particular importance to mobile robot positioning as they help compensate for the foremost weakness of odometry: in an odometry-based positioning method, any small momentary orientation error will cause a constantly growing lateral position error [6].

The observation measurement model for the gyroscope is calculated using the equation

$$\varpi(t) = \omega(t) + e(t) + w_k \quad 4.7$$

$$\varpi_k = \sum_{i=1}^{k}\omega_i + e(t) + w_k \quad 4.8$$

V. FUSION METHOD

The information obtained from different sensors is then fused based on the rule based algorithm that's been created. The Obstacle avoidance algorithm that's been developed for the Ultrasonic sensor is made to run by default until specific input is given to the controller.

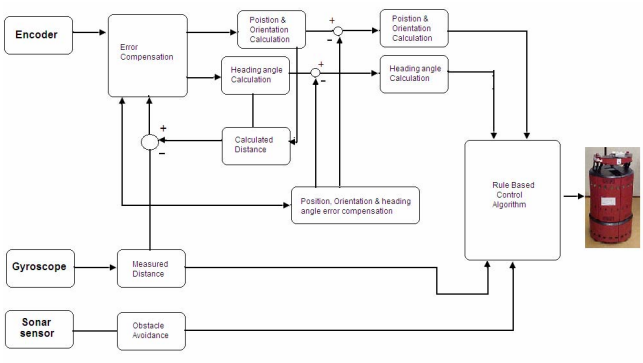

Figure 5. The Block Diagram of control logic

The figure shows the information obtained from the encoder are corrected based on the calculated value and the real value and then fed to the rule based controller, similarly the information from gyroscope is fed to the controller and then shared with the encoder compensation to provide the position and orientation information when the robot controller switches the controller back to encoder.

VI. RESULTS & CONCLUSION

The figure 6.1 shows the raw encoder data i.e when the robot is given the input to take a circular path, the robot drifts approximately 50cm from its starting position and the line in blue dots shows the obstacle avoidance and the encoder algorithm fused.

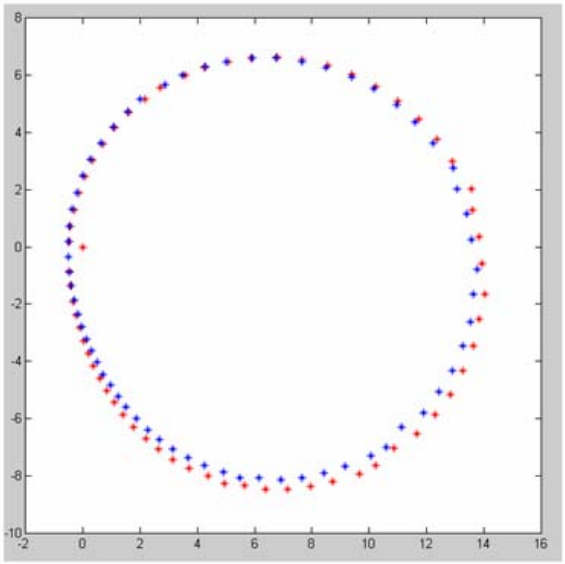

Figure 6. Represents the Error Correction with Fusing Ultrasonic and Encoder Data

The simulation results show the correction model developed for the encoder, gyroscope & Ultrasonic sensor. The algorithm developed & described by this technique was tested on a single degree of freedom Mobile robot –B21r. The algorithm was built over the objects and classes defined in Mobility & CORBA using C++. The results & simulations as discussed in this paper shows significant amount of accuracy compared to the conventional method used in the B21 robots.

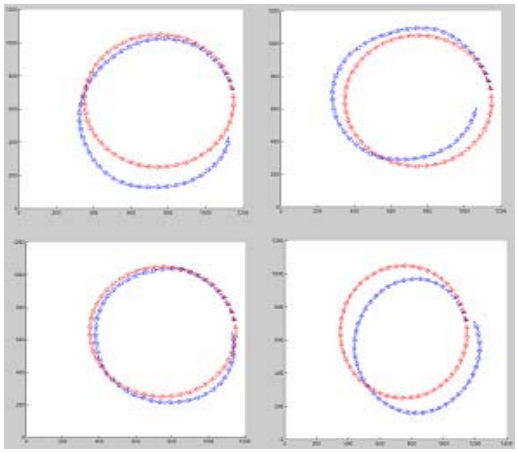

Figure 7. Simulation Reults

The Future expansion of the robot can be done by developing the algorithm on a common platform like C++/ C or Java, this would widen the application of the algorithm. The Mobility Integration software which is used in B21r robot is obsolete; the algorithm can be tried on a new integration software developed for B21r robot.

VII. ACKNOWLEDGEMENT

This work is supported by National Science of Engineering and Research Council.

REFERENCES

[1] [1] David L.Hall and James Llinas , *HANDBOOK OF MULTI SENSOR DATA FUSION:* CRC Press,2001.

[2] [2] Kai-Tai Song and Charles C. Chang, "Ultrasonic Sensor Data Fusion for Environment Recognition," *proceedings of the 1993 IEEEJRSJ International Conference on Intelligent Robots and Systems, Japan July, 1993.*

[3] [3] Johann Borenstein and Yoram Koren, "Obstacle Avoidance with Ultrasonic Sensor," *IEE Journal of Robotics and Automation, Vol. 4, No.2, April 1998.*

[4] [4] Jason Gu, Max Meng, AI Cook, Peter X. Liu, "Sensor Fusion in Mobile Robot: Some Perspectives," *Proceedings of the 4' World Congress on Intelligent Control and Automation June, 2002.*

[5] [5] Irobot, B21r Mobile robot User's Guide and Mobility Robot Integration Software User's Guide.

[6] [6] J. Borenstein1, H.R. Everett, L. Feng, and D. Wehe, "Mobile Robot Positioning &Sensors and Techniques," *Invited paper for the Journal of Robotic Systems, Special Issue on Mobile Robots. Vol. 14 No. 4, pp. 231 – 249..*

[7] [7] Haeyong Yang, Kyucheol Park and Jang Gyu Lee, "A Rotating Sonar and a Differential Encoder Data Fusion for Map-Based Dynamic Positioning*" Journal of Intelligent and Robotic Systems* ***29:*** *211–232, 2000.*