MEMS-Based Inertial Measurement

Sebastian Sabou
Electrical, Electronic and Computer Department
Technical University of Cluj Napoca - North University Center of Baia Mare
Baia Mare, Romania
sebastian.sabou@ubm.ro

***Abstract*—Creating an environment for testing and verification of algorithms for navigation and orientation implemented in intelligent embedded systems is an essential step in a project development process. By testing several variants of algorithms, or a combination of the two lead to a reduced numbers of errors, which is one goal of project. A test was performed for navigation algorithm using map matching and a 2-axis accelerometer to determine speed and direction of moving a mobile unit (robot) on a small map.**

***Keywords*— Map matching, IMU, ATMEGA644**

I. INTRODUCTION

The main goal of this project is to achieve a working environment to verify map matching technique for navigation, using Micro-Electro-Mechanical Systems (MEMS) sensors for measurements. For experimental propose a 64cm x 64cm area enclosure, a 5cm x 5cm mini robot containing a minimum numbers of sensors (in this case just accelerations sensor), a wireless communication system, a battery and motors for movement, are used. The processing data module is separated, because of limitation of robot physical dimensions and to obtain a more versatile system, to use of the various computing platforms which is much easier in this case. The schematic of the entire system is presented in figure 1.

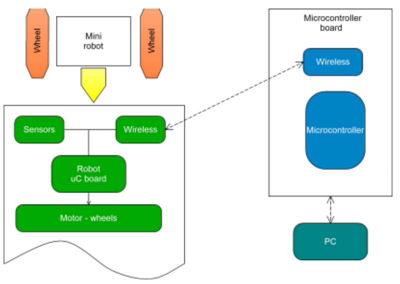

Fig. 1: Schematic of the system.

A. Acceleration

The signal obtained from the sensor can't be used directly to obtain velocity and position requiring processing the input data. To obtain position requires a double integration of the obtaining information on velocity. Presented algorithm can be used for a single axis or two axis, one exception is in the case of the third axis where additional processing is required to cancel the influence of gravity earth.

The acceleration (a) can be considered as the change rates of the object velocity (v), and the velocity is the change rate of the object position (s) as show in (1).

$$\vec{a} = \frac{d\vec{v}}{dt} \text{ and } \vec{v} = \frac{d\vec{s}}{dt} \text{ result } \vec{a} = \frac{d(d\vec{s})}{dt^2} \quad (1)$$

The integration is the opposite of the derivative. In other word, if we known the acceleration of one object we can obtain the velocity with integration of acceleration of that object and the position of the same object with double integration of acceleration or just integration of the object velocity (2).

$$v = \int(\vec{a})dt \text{ and } s = \int(\vec{v})dt = \int(\int(\vec{a})dt)dt \quad (2)$$

To simplify the implementation of this formula will read values from acceleration sensor with a constant frequency. Considering this, the approximate formula for integrations is presented in (3).

$$v = A_n + \frac{A_n - A_{n-1}}{2} \quad (3)$$

where A_n represent acceleration value read at the moment T and the A_{n-1} represent acceleration value at the moment T-1.

The whole idea is based on the equivalence between the integration of function f(x) and the area under the curve f(x), area calculated using "different" method (fig 2).

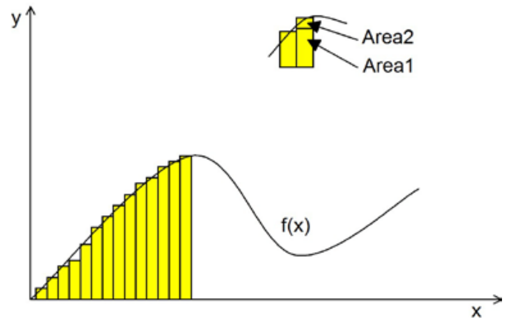

Fig. 2 Integrate function f(x)

Formula (3) is the sum of Area1 and Area2 from Figure 2.

For position we need to repeat the formula (3) changing acceleration with velocity values, realizing in fact, a double integration of acceleration.

B. Map matching

In general, map matching algorithms integrate estimated locations, from positioning sensors such as GPS and DR, with a road network map to identify the correct link on which a vehicle is traveling and to determine the location of a vehicle on that link [8]. Map matching plays a crucial role in a navigation system whose logic is heavily dependent upon the characteristics of the underlying positioning sensors. The success of a navigation application mainly depends on the suitability of its positioning sensors and the map matching algorithm.

Current map matching algorithms can be divided into three main approaches, geometric map matching, topological map matching and advanced map matching [9]. Geometric map matching consists of point-to-point map matching, point-to-curve map matching and curve-to-curve map matching. Topological map matching utilizes both geometrical and topological data to make matching decisions.
Advanced map matching applies models, such as probability theory, Kalman filter, and fuzzy logic, to either geometrical map matching or topological map matching, so the algorithm obtains better matching results with an increase in the complexity of the computations required.

II. IMPLEMENTATION

A. Sensor

For acceleration measurement an Analog Device accelerometer ADIS 16209 [7] sensor was used. The sensor has 0.244mg resolution on measuring acceleration and the range is +/- 1.7g. Functional block diagram is presented in Fig. 3.

The sensor output is digital, the acceleration is stored in 14 bits in two's complement format and use SPI protocol for data transfer. ADIS 16209 is a dual axis accelerometer and inclinometer. For testing the experimental platform a dual axis accelerometer was considered enough.

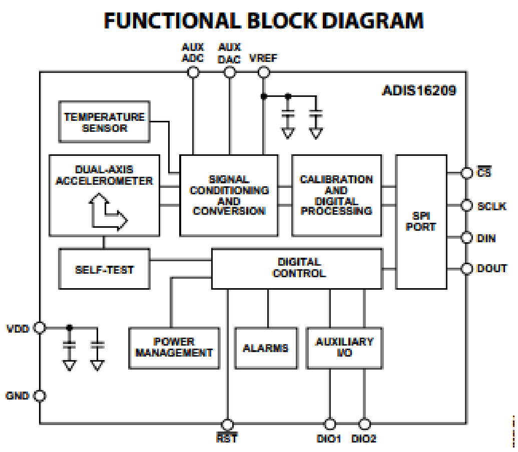

Fig. 3 Functional block diagram ADIS 16209[9]

B. Robot

The hardware is composed of two units: a mobile unit and a base unit. The reason behind this configuration is to move the processing unit out of mobile unit, in this way can be choose a different platform for processing without change on mobile unit. The only change is to implement the RF communications with the mobile unit to read sensors and to send command for move the unit.

C. Mobile unit

The mobile unit contains minimal hardware parts necessary to read sensors, transmits output data, receive commands, sends commands to servo and monitoring the battery voltage. For main board is used an Arduino Mini board, communication is made with nRF24L01 2.4Ghz radio module and for move is used two modified RC servos attached on two wheels.

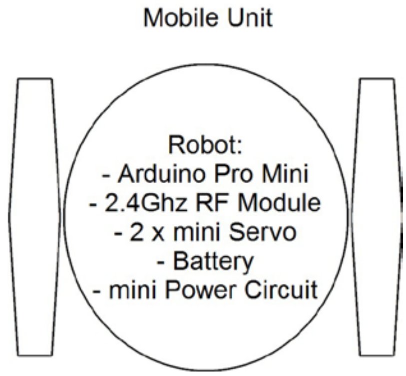

Fig. 4 Mobile unit components

The sensors are not mentioned in Figure 4 because is designed to use a wide range of sensors. The weight of entire mobile unit is as low as possible with a small battery but big enough to ensure longer operating as possible.

D. Base unit

The base unit is designed in a way that can be changed very easy and can be replaced with another computing platform to implement different algorithms. The only step that is needed to be made is to implement the communications with radio unit and with mobile base. A minimal commands and protocol needed to be implemented and the entire system is ready for experiments.

In Figure 5 are presented the actual configuration for this experiment.

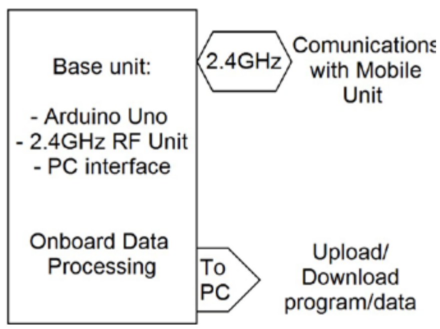

Fig. 5 The Base Unit Schematics

III. EXPERIMENTS

For use with other accelerometer sensors, a calibrating function was implemented. This function read about 1024 samples and made their average, all samples are read in no movement conditions (static acceleration). The average value is considered as zero acceleration value.

For input data was used a simple filter, 64 accelerations values are read and the average value are used as input for other formulas.

When the sensor is in no movement state, the sensor output some small value due to minor error from accelerometer, and these small values are interpreted as constant velocity, being summed all of this value as shown in formula (3). For these reason it was necessary to differentiate the valid value from the others, a discriminator function was implemented (Fig. 6).

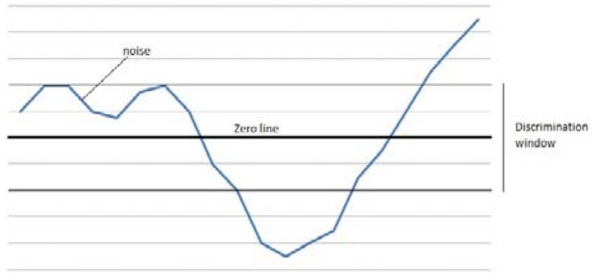

Fig. 6 Discriminator window

Another function implemented, resulted from experiments, is the NO MOVE. As shown in Figure 7, after initial acceleration (the upper area – sample 1-10) is follow with deceleration (the low area – sample 10-19).

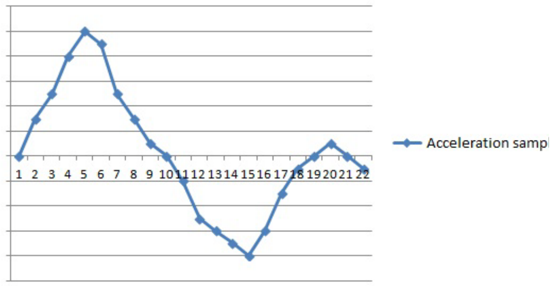

Fig. 7 "NO MOVE" function

From the theoretical point of view, after the initial acceleration and subsequent deceleration, final velocity is zero, which has the effect of change the position of object and a new stable position is achieved. In real experiment, the area from samples 1-10 is not equally with the area from samples 10-19 and the result never reach zero velocity. To obtain zero velocity, a new function is needed to verify constantly accelerations values and compare with zero. If this condition exists during a certain number of samples, velocity is simply returned to zero.

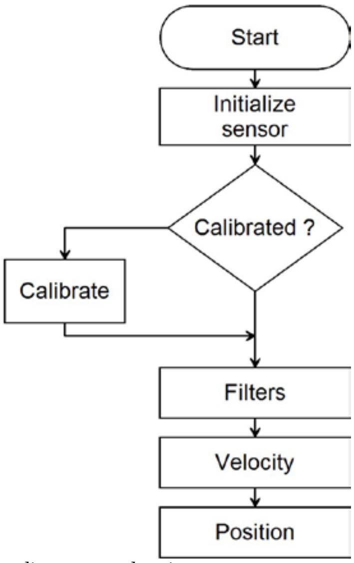

Fig. 8 Flow diagram acceleration

The flow diagram for calculating acceleration is show in Figure 8.

Because this MEMS sensor don't have output for angle of move the only way to compute the direction of move is from acceleration from X axis and from Y axis with trigonometric function. This is complicated to implement in an ordinary 8 bit microcontroller, so this is not implemented in this setup. In future a gyroscope will be used to help in finding directions of movement.

The focus was on algorithm to compute velocity and position of the mobile unit with linear movement, and to move the mobile unit for a determined distance.

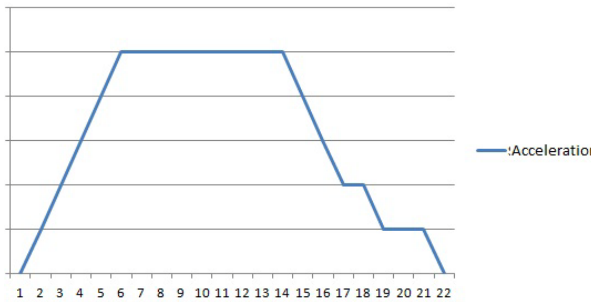

Fig. 9 Acceleration profile

In Figure 9 is presented the profile for acceleration that was used to make a move on a fixed distance. Depending of distance an acceleration growth rate is chosen, for short distance the acceleration rate is small and for longer distance the rate can be increase. This is chosen according to mechanical inertial of mobile unit.

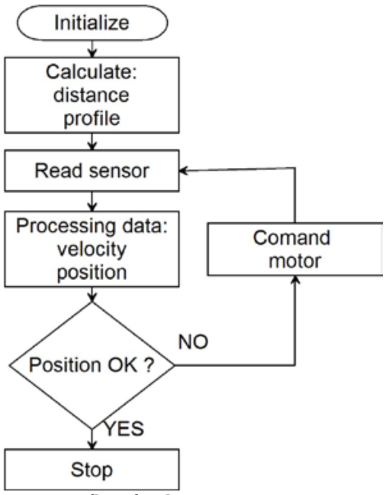

Fig. 10 Program flow for distance

The flow diagram for calculating distance is show in Figure 10.

For testing purposes and for testing a few different algorithms a matrix space was chosen. Dimension of this working space is 64 x 64 cells, a total of 4096 cells. If each cell occupies 1 bit result a total of 512 bytes. “0” means a free cell and “1” an obstacle cell. All the space is a matrix with 64x64 cells with each cell can have value of 0 or 1.

IV. Conclusions

A functional system which allows verification of algorithms used for autonomous navigation and guidance it was obtained. The proposed system is modular and allows adding or changing several types of sensors, designed to find the optimal configuration for a particular algorithm.

Digital sensors are a good option for use in a smart navigation system, thus eliminating signal processing on the robot side and reduced physical dimensions for very elaborate systems. By using external computing system in this robot design, in this phase of project development, allows them to adapt it without any restrictions regarding to dimension of the robot.

For data processing an ATMEGA 328 microcontroller it was used. Its choice was imposed by the 2KB SRAM, necessary for map storing. The map is stored in a bitmap format with a resolution of 1 cm, with size of 64 x 64 representing the working space which occupied 4096 bits. For the first trial, 512 bytes of memory is enough, considering that each bit corresponds to a square centimeter. If set to logic 0 corresponds to a free space, one corresponding to a space occupied by an object. The remaining SRAM is provided as a safety measure, if desired workspace expansion or increase resolution (more bits for each square). In Figure 11 is presented the map used to test the map matching algorithm. The black line represents the obstacles (walls), 1 cm thick, and the robot trying to avoid them. In this case the wall is used just for try to not reach them, to keep a safe distance from them.

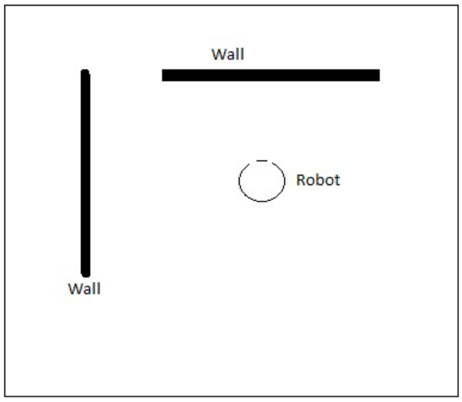

Fig. 11 Testing map with obstacles

References

[1] S Oniga, P Pop-Sitar, Application Possibilities of Hardware Implemented Hybrid Neural Networks to Support Independent Life of Elderly People, Hybrid Artificial Intelligent Systems, Springer Berlin Heidelberg, 520-529, 2013

[2] S. Oniga, J. Vegh, I. Orha, "Intelligent Human-Machine Interface Using Hand Gestures Recognition", Automation Quality and Testing Robotics (AQTR), 2012 IEEE International Conference on ,pp. 559 - 563, 2012.

[3] Miguel Pinto, Héber Sobreira, A. Paulo Moreira, Hélio Mendonça, Aníbal Matos, Self-localisation of indoor mobile robots using multi-hypotheses and a matching algorithm, Mechatronics, Volume 23, Issue 6, September 2013, Pages 727-737

[4] **S. Sabou**, I. Orha , C. Lung, Inertial navigation modular system, SIITME 2012

[5] **S. Sabou**, I. Orha , C. Lung, An. Buchman, Autonomous navigation system – map matching, Carpathian Journal of Electronic and Computer Engineering, ISSN: 1844-9689, 2012

[6] **S. SABOU**, Claudiu LUNG, Ioan ORHA Title : Reference for Indoor Location Systems, NOVICE INSIGHTS Issue 11, 2011

[7] Jared B. Bancroft, Department of Geomatics Engineering, University of Calgary, Multiple Inertial Measurement Unit Integration for Pedestrian Navigation , UCGE Reports, Number 20320, Dec 2010

[8] Tisan, S. Oniga, A. Buchman, C. Gavrincea, “Architecture and Algorithms for Synthesizable Neural Networks with On-Chip Learning”, International Symposium on Signals, Circuits and Systems, ISSCS 2007, July 12-13, 2007, Iasi, Romania, vol.1, pp 265 – 268

[9] Analog Device, ADIS16209 datasheet.

[10] AN3397 application notes, Sensors - Freescale Semiconductor

[11] Quddus, M. A., High integrity map matching algorithms for advanced transport telematics Applications, A thesis of the University of London, 2006.