Navigation System for a Mobile Robot Using Kalman Filters

David Vaz
Academia Militar
IST / UL
Lisboa, Portugal
Email: david.ferreira.vaz@ist.utl.pt

António Serralheiro
CINAMIL, Academia Militar
INESC-ID
Lisboa, Portugal
Email: antonio.serralheiro@academiamilitar.pt

José Gerald
IST / UL
INESC-ID
Lisboa, Portugal
Email: jabg@ist.utl.pt

***Abstract*—This paper is intended to be an extended abstract of a MSc Thesis that focuses on the development of a navigation system for a terrestrial mobile robot, aiming at its autonomous movement through a default route. This work gathers information about the State of the Art of Kalman filters and navigation systems. The developed method uses the sensors in the robot (IMU (Inertial Measurement Unit), GPS, and odometers) and processes information in Matlab and C# on a remote PC using a Wireless network for communication with the robot. The information from the odometers is combined to estimate the actual position of the robot and to evaluate the error between this position and the GPS position. This error is filtered with linear Kalman filters and then returned to the estimated position, balancing the data from various sources taking advantage of the predictive capabilities of the Kalman filter. The objective is to obtain an optimal discreet estimation of the robot's actual position. This position is then used for the automatic navigation of the robot, which is based on a demonstration program provided with the robot. However, this program was changed and adjusted to improve the process. Evaluation tests that illustrate the ability of the robot to automatically travel through a predefined path are presented confirming the validity of the work and the operation of the algorithms.**

***Index Terms*—autonomous navigation, estimated position Kalman filters, motion sensors, noise filtering.**

I. INTRODUCTION

AUTONOMOUS navigation systems are important for the scientific community. Its development is easily noticeable with the disclosure of new technologies. Sometimes this may be difficult because:

- There may exist software and hardware integration problems;
- The high level of complexity of the activities that are intended to be done;
- It may be difficult to acquire and process the system data that is important to know the status of the systems. This data may not have the sufficient quality to be useful because the limitations of the sensors, inadequate processing power, or presence of noise.

The developed navigation system described in this paper must use the capability of acquiring guidance data of the robot to determine its position and make it move along a path previously defined with the lowest error / deviation possible. The main objective is the acquisition of information provided by the various robot sensors (accelerometer, gyroscope, GPS receiver, magnetic compass, etc.), its filtering and processing, and delivery of new data to the robot for it to develop its activity autonomously. Since the information of the sensors is associated with an inevitable amount of noise, this control system will be based in information filtering using Kalman filters in order to reduce errors in robot navigation. The development of this filtering is the main focus of this dissertation.

II. STATE OF THE ART REVIEW AND THEME'S SCIENTIFIC BACKGROUND

The concept of automatic navigation consists in the ability of a system to move autonomously in order to meet objectives previously defined, not just wandering erratically. For this, it is necessary to know the state of the system (position, speed, and direction) at successive time instants with the lowest possible error. This knowledge depends only on the capacity of the sensor system to get information about the surrounding environment. After successfully finding the instantaneous actual position, the autonomous system should attempt to travel a predefined path, developing its activity in order to reach the next point of the trajectory from its position. This performance must be controlled considering the errors and noise that exist in the known movement data.

It is intended in this work, the processing of such data using Kalman filters, to obtain a good estimate of the true state of the system.

The use of such filtering method is justified, above all, for its simplicity of implementation and recognized good performance in predicting states and noise reduction. Its ability to determine future values makes the Kalman filters the ideal tool for real time system controlling, avoiding the typical delays introduced by other filtering methods that harms the navigation performance.

This work was developed with a robot owned by the Portuguese Military Academy.

A. Kalman Filtering

The Kalman filter is a tool to estimate the variables of a set of processes, i.e., estimates the states of a linear system. This filter minimizes the estimation error variance and its implementation is relatively simple [1]. It is widely used in applications such as tracking objects, economy, navigation or computer applications such as fusion of radar data, laser scanners, and cameras for speed and depth measurements [2]. The dynamics of the Kalman filter results of consecutive cycles of state prediction and information filtering and these cycles are based on the structure of the Gaussian distribution probability function.

The general application of the Kalman filter indicating graphically the control process of a system using this filter is depicted in Fig. 1.

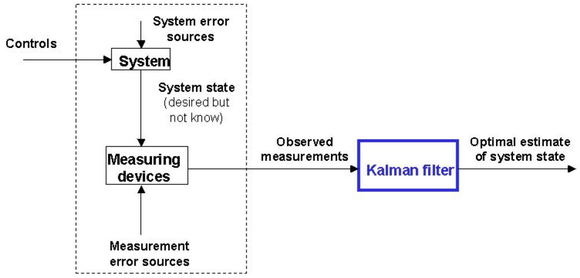

Fig. 1. Generic application of Kalman filter [1].

Actually, the ability of this filter to predict future values makes it the adequate tool to use in this real time controlling system, avoiding the characteristic delays introduced by other filtering methods that are harmful to the navigation performance.

A mathematical overview of the Kalman filter process is shown in Fig. 2 and further details about this filter implementation can be found in [3].

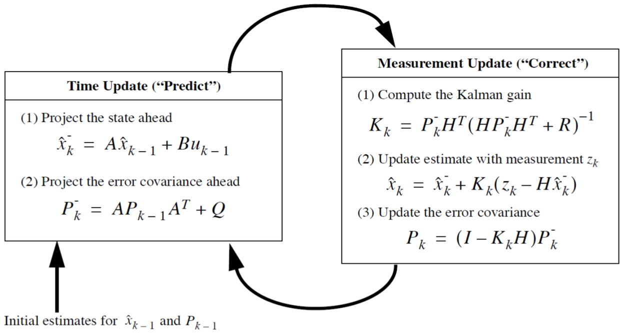

Fig. 2. Kalman filter algorithm [3].

It is well known that the best way to consider all variables and the nonlinearities of the system would be using a nonlinear filter such as the Extended Kalman Filter (EKF). However, in this implementation, this was avoided due to several well-known issues [4], namely: nonlinear filtering can be difficult and complex and it is not as well-understood as linear filtering; to use an EKF one needs a mathematical system representation model; accurate system noise models are needed and more computational resources are necessary. In fact, although a mobile robot is not a linear system, it can be fairly approximated by a linear one and, therefore, it is expected that linear estimation approaches would give good results.

B. Examples of Kalman filter based navigation methods

Self-location methods using external sensors can be divided into two groups [5]: global methods and local methods. In local methods, the robot creates a map of the surrounding area, based on the relative distance to existing objects in the environment and then matches this map with a global map previously saved. Thus, the robot finds its position in the space. In global methods, the robot calculates its position directly on the coordinates of the space in which it operates, using the distance to certain reference points.

Local methods have the advantage of allowing anti-collision movement and environment map reconstruction by using distance and self-location system sensors. However, these methods require a large amount of processing power to create local maps and compare it with global maps stored in the system. Sometimes it is also necessary to interrupt the movement of robots to get location information in order to obtain a correct observation of the environment [6] [7].

On the other hand, global methods can avoid the creation of local maps and comparison processes with global maps. Self-location with these methods is also computationally efficient and fast [8] [9].

An example where the filtering of information is important are the artificial satellites that require reliable, safe, and completely autonomous navigation systems, especially for emergencies in space.

Celestial navigation is a fully autonomous navigation method in satellites. For satellites in close orbit to earth (low Earth orbit - LEO), the direction to the Earth is the most important measure and the precision of the horizon detection is crucial to the accuracy of celestial navigation.

The acquisition of the horizon location can be done directly through sensors or indirectly through the light from celestial bodies refracted by the atmosphere. Since there is complementarity between these methods, Kalman filters may be used to merge information, which enables an improvement in the performance and reliability of the navigation system, as proposed in [10]. In this case, localization is determined using external sensors, instead of dead-reckoning data.

Another example using the Kalman filter for navigation is the robot Pygmalion from École Polytechnique Fédérale de Lausanne [11]. This robot uses a telemetric laser sensor to create a continuous and abstract environmental representation and make a map of the surrounding areas. Its estimated state is represented by a Gaussian distribution, because it uses the Kalman filter in the location algorithm. The value of its average position is represented with a high level of precision, making it possible to obtain its actual location with a small error. This system uses a method of local self-location and may be defined as a system for autonomous navigation of mobile robots using the vision-based recognition.

Another interesting implementation of Kalman filters that can be reported is the autonomous navigation system for robots with position estimate using triangulation of radio signals [12].

This method lies in the use of radio signals along with other sensors in the robot such as compass and odometers, for the robot to perceive its position in space. It is accomplished by doing position triangulation using multiple beacons distributed in space that have radio transceivers with different signal signatures. It will be necessary, however, to know the number of beacons and their location in space. In Fig. 3, it can be seen an operating scheme of this system.

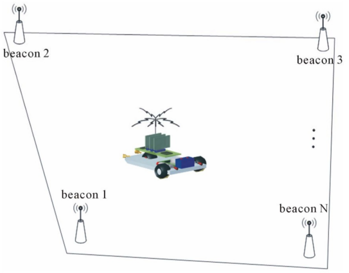

Fig. 3. Beacon distribution and communication system example [12].

This configuration is an example of a global self-locating method, where there are used landmarks (beacons) for the robot to be situated in the space.

With the map of the space and calculating estimations of system states (using the Kalman filter) at time k+1, based on the current robot position, it is possible to have knowledge about the evolution of the system with different commands that are sent to the robot. Fig. 4 illustrates a scheme with the overall operation of the navigation system for this robot. It is possible to check that data is collected by the odometer, compass and radio triangulation (perception) sensors, which are then compared (matching) with the predicted observation.

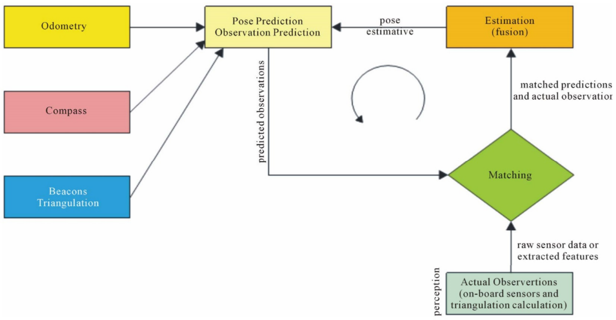

Fig. 4. Algorithm diagram for the location of the robot using the Kalman filter [12].

With this information it is calculated the following position estimation (estimation / estimative pose) which is used to estimate the next observation to be compared with the next observation time instant.

The work of Soo-Yeong Yi and Byoung-Wook Choi [5] is an example that has similarities with the previous one but approaches the method developed for this MSc Thesis. It uses a set of four or more fixed ultrasonic sensors on referenced positions that represent pseudo-global coordinates (similar to GPS) as illustrated in Fig. 5. In this case the coordinates are related to a defined space inside a building, but the system used in the MSc Thesis uses (among others) the GPS system in order to obtain outdoor location information.

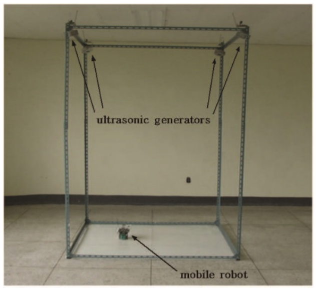

Fig. 5. Autonomous navigation system scheme using a global ultrasonic system [5].

Thus, the work of Soo-Yeong Yi and Byoung-Wook Choi [5], is based on distance measurements between the ultrasonic transmitters and receivers on the robot and distance data fusion algorithms. It is then possible to determine the position of the robot in the pseudo-global coordinate system.

As in the previous example, the distance between the robot and the transmitters is based on the time delay between the signal sent and received between the robot and the emitter. The application of extended Kalman filter to the results obtained by odometry and the ultrasonic sensors enables increased accuracy of the particular location. Fig. 6 presents some results of this work demonstrate the importance of the data fusion from an internal and external location system using Kalman filter.

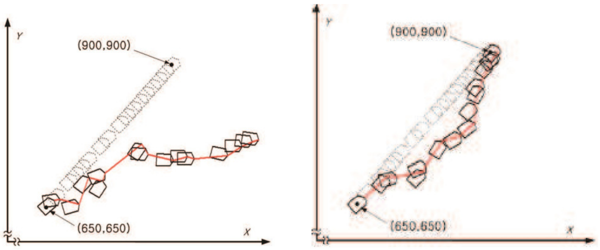

Fig. 6. Paths followed by the robot considering different forms of self-location – Dead reckoning only on the left and data fusion with Kalman filter of dead reckoning and ultrasonic sensors on the right [5].

The last example is the work made in the MSc Thesis developed in [13], whose objective was also the development of an autonomous navigation system for the robot Jaguar, using the linear Kalman filter to directly filter the received data from accelerometer and GPS. In Fig. 7 is depicted the result of two navigation tests and the deviation from the desired trajectory.

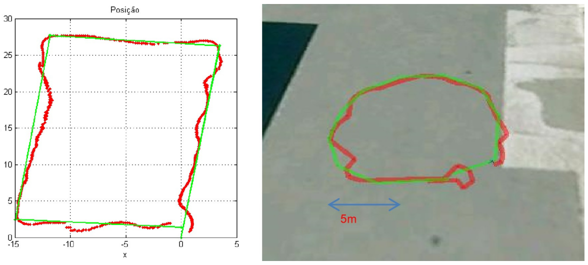

Fig. 7. Tests to the Jaguar robot navigation with linear Kalman filter in a rectangular path (15mx25m) on the left and a circular path (5m radius) on the right [13].

C. *Robot Jaguar 4x4 Wheel*

The robot used in this work is the Jaguar 4x4 Wheel manufactured by the Canadian company Dr Robot (Fig. 8). This robot is designed to operate inside and outside of buildings, in rocky terrain in activities that require quick operation and control.

Fig. 8. Robot Jaguar 4x4 Wheel [14].

It is equipped with an 80W motor for each one of the four wheels, reaching a maximum speed of 11km/h. It features a tough construction, weighs about 20kg, and is water resistant. The robot is powered by a lithium polymer battery (Lipo) of 22,2V with a capacity of 10Ah, which allows its operation for about 2 hours. The communication with the device is performed via wireless (IEEE 802.11g standard).

The robot includes a GPS module and a set of sensors (gyroscope, compass and accelerometers) that allows the performance of autonomous navigation. It also performs video and sound capture. Although there is a program to control the robot, it is possible to develop software for this purpose [15].

The communication between the inside components (motor controllers, sensors, motherboards, etc.) is performed via Ethernet forming a network between the components. This network is a Local Area Network (LAN) that also includes a wireless router that establishes wireless communication with the computer making it possible to remotely control the device.

- GPS Sensor

The GPS system will be the main reference for the robot to determine its location and therefore to the navigation. Basically, this sensor uses data from the GPS satellites orbiting the earth to determine the robot position.

The GPS sensor used in the robot is the Garmin GPS 18x 5Hz (Fig. 9). This is a high sensitivity sensor for use in monitoring applications and orientation in machines that require speed and position reports with the frequency of 5 Hz which allows a good accuracy in location determination [16].

Fig. 9. Garmin GPS 18x 5Hz [16].

However, the data obtained by this sensor is also affected by noise and measurement errors, particularly if the measurements are obtained in areas with obstructions in the space between the sensor and the satellites.

These obstructions may be, for example, buildings or trees, and their effects can be the reduction of the satellite signal power, or reception of multipath signals leading to information corruption.

- Inertial Measurement Unit (IMU) Sensor

These sensors are an accelerometer (ADXL345), a gyroscope (ITG-3200), and a magnetic compass (HMC5843). They are included in a single integrated circuit (SEN-10125). The three sensors have the ability to collect measurements in the X, Y, and Z axes (Fig. 10), which are combined and sent to the robot communication system. Thus, the accelerometer measures the acceleration occurring in each of the three axes XYZ, the gyroscope measures angular velocities recorded around each axis, and the magnetic compass converts Earth magnetic fields into voltages on each axis. In Fig. 11 is shown the board that integrates the sensors.

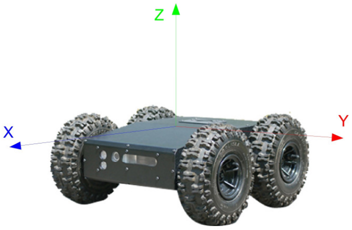

Fig. 10. Robot axes scheme.

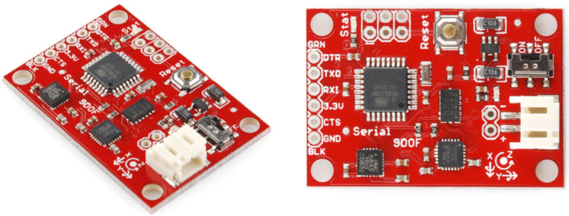

Fig. 11. Sparkfun 9DOF-Razor SEN-10125 [17]

- Wheel encoders

The third type of sensors that can be used in robot navigation are the encoders in the wheels. These sensors check the position of the wheels by adding the number of steps that each wheel moves during the robot movement, i.e., the total angle of the wheel (360º) is divided into small spaces that are recorded during the circular motion of each wheel. This sensor has a resolution of 150 counts per wheel revolution corresponding to a travelled distance by the robot of about 0.57 cm between each count (considering the wheel diameter of 27 cm). The reading is done with events created from robot's control system at a frequency of 20 Hz.

The stated control system is a special version of the Dr. Robot PMS5005 board and it is a platform that makes readings of motion sensors and performs engine control. This card also receives the data from the wheel encoders, the temperature of the motors and controls its movement. In addition, the battery voltage is read by this board.

III. DEVELOPED WORK AND IMPLEMENTATION

The work for this MSc thesis was the development of an autonomous navigation system for a mobile robot based on the Kalman filtering of information provided by the sensors of this robot (IMU, GPS, and wheel encoders).

A. *Remote computer connection*

It was possible to test the original navigation program provided with the device. This application is named Jaguar Navigation Demo (Fig. 12) and it allows to navigate with the robot in a user-defined route. This application uses a very basic navigation method that is strongly based in the GPS data and is greatly affected by its associated errors.

Target Latitude: 0.00000000
Target Longitude 0.00000000
Target Heading: 0.00
Step: 1
Plan Path Step: 0
☑ GE En
FileName:
Dis2Path: 0.00
Dis2Target: 0.00m
Direction Error: 0.00
PresetYaw 0
Latitude: 40.77014667
Longitude: -7.07200267
Yaw: 67.43
GPS COG: 0.00
GPS VOG: 0.00
☑ EnableGPS
SetPoint Latitude: 38.7266654463993
SetPoint Longitude: -9.13893756146665
☑ Joystick
☑ GPS Display
IMU Sensor IP: 192.168.0.61
IMU Sensor Port: 10001
GPS IP: 192.168.0.61
GPS Port: 10002
GPS Quality: DGPS fix avaiable
IMU Raw Data: $208,2,-2,241,22,-13,-29,0,0,0#
GPS Raw Data: $GPRMC,232821.6,A,4046.20880,N,00704.32016,W,000.00

Fig. 12. Jaguar Navigation Demo graphical interface.

This program will be used as the basis for the development of the Kalman filter navigation algorithm (subject of this work) because it already includes all the communication platform with the robot and allows to focus the effort in the development of that algorithm.

Thus, since the source code is available, the program will be adapted to be able to exchange data with Matlab that has a great calculation capacity and important tools in mathematical data manipulating. In general, the implemented algorithm in Matlab have the function of receiving raw data from the sensors (IMU, GPS, and wheel encoders) and return a filtered estimate of the instantaneous position of the robot, all this using the navigation program as the communication platform. After receiving the filtered position, the navigation procedures will be accomplished by this navigation program.

Since the Jaguar Navigation Demo is an application developed in object-oriented C# language, the adaptation and development of the source code was done in Microsoft Visual Studio 2015 environment.

B. *Navigation method*

The process of following a defined path requires the use of algorithms and methods for calculating the instantaneous position of the robot and to determine the evolution of its movement. Thus, the system should accurately estimate the position in which the robot is in every moment from the available data, must be aware of the position to achieve, and must then determine the actions to take to reach the desired position. This set of procedures is achieved by the navigation program as follows:

- Determination of the instant position of the robot: To determine the instant position of the robot, the raw data received from it is send to the Matlab developed algorithm to be processed. This algorithm will then return the filtered estimated instant position.

- Determination of the path to follow: To reach the target point from the position where the robot is, the program is mainly based on the calculation of two orientations. One of these orientations is defined from the point where it is estimated that the robot is located to the target point, the second orientation is defined by its instantaneous movement. When the two directions differ more than 5º (defined by default but this value can be changed) the program initiates corrective measures to turn the robot until the difference becomes less than this value (Fig. 13). This procedure is repeated until the estimated position of the robot lies within a certain distance of the destination point (1 meter by default). In this situation, the program will consider the following point as the destination point.

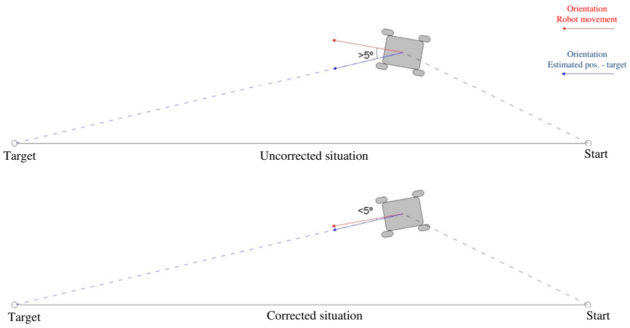

Fig. 13. Demonstration of the correction made by the navigation method to reach the target point.

In this process, the navigation program sends instructions to the robot that allow it to change the route, through the differentiation of the power sent to the wheels on each side, which allows the robot to make the curves.

The need for accurate determination of the two above mentioned directions becomes apparent with the understanding of this navigation method:

- It is important that the estimation of the robot instantaneous position is obtained with the minimal possible error, be free of noise, and with the smallest possible delay, to obtain a correct orientation of the instantaneous position to the destination point.
- The second orientation, in which the robot moves at every moment, is also very important. It is what tells if the device is moving towards the intended direction. In the original program, it is computed from the GPS and gyroscope data fusion. Thus, with the evolution of GPS coordinates the program determines the robot travelling direction. The gyro data helps in situations of sudden turns that GPS does not have the necessary resolution.

This combination of data may have some problems, especially noise that can be picked up by the robot in rough terrain. Also, in situations where the GPS signal may be affected (reflections, obstacles) and there are variations in the received positions, the instantaneous orientation of the robot will have wrong values and will be noisy.

C. *Generated raw data analysis*

In order to justify the developed method in the position estimation Matlab algorithm, it will be presented the most relevant aspects about the information generated by the sensors in the robot and described the most important characteristics of this data.

A default route was made up manually to be possible to compare the raw data sent by the robot with the movement actually performed. This route, approximately rectangular, was held in a flat ground (road) and is illustrated in Fig. 14. It begins at point A (0.0) and finished in B. This scheme was obtained with the conversion of GPS coordinates received during the movement to distance in meters between the points for these coordinates.

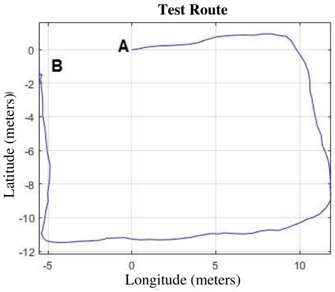

Fig. 14. Test route performed manually to observe the data sent by the sensors.

1) *Accelerometer*

Acceleration data provide information about the robot speed variation in the X, Y, and Z axes. Only the X axis is considered because it is the one that matches with robot motion and more directly reflects the velocity variations in normal conditions, considering the most likely cases that the robot can move (leveled flat surfaces without slippage). The data for the Y-axis give information about the lateral acceleration and the Z-axis gives the vertical acceleration.

The output for this sensor shows a noisy data with peak values without any meaning as observed in Fig. 15.

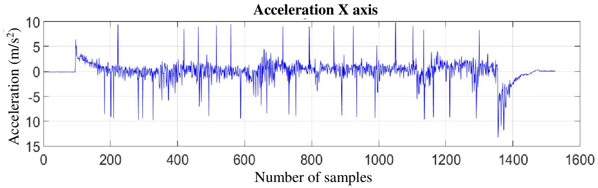

Fig. 15. Raw data for acceleration in X axis relative to the path of Fig. 14.

It was decided that the accelerometer data would not be used due to the need of double integrate the values to obtain the position estimation. This method propagates and adds errors that severely damage the calculated information.

2) *Gyroscope*

As the acceleration sensor, the gyroscope also returns values relative to the referred axis system. These values are a measure of angular velocity around each one of these axes. In this case, the most important measure is the angular velocity around the Z axis, because an orientation change in the trajectory of the robot is reflected this axis. Again, the integration of the values (angular velocity) is necessary to be able to determine the change in direction (angle).

The output is indicated in Fig. 16 where it can be seen that, in this case, the data is very clean and sharp without peak values or noise.

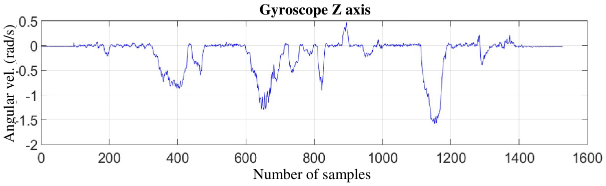

Fig. 16. Raw data for gyroscope in Z axis relative to the path of Fig. 14.

3) *Magnetic Compass*

The compass works by converting the magnetic fields in each axis into voltages. It is then possible to know the relative orientation of each one to a reference point - the Earth magnetic North. This is especially important because it helps to place the robot movement in a well-defined referenced plane, allowing further combination of information from other sensors without reference (encoders and IMU) with GPS.

However, after the analysis of the generated data by the sensor, the conclusion is that it could not be used because it is very noisy, as can be seen in Fig. 17, regarding the route of Fig. 14.

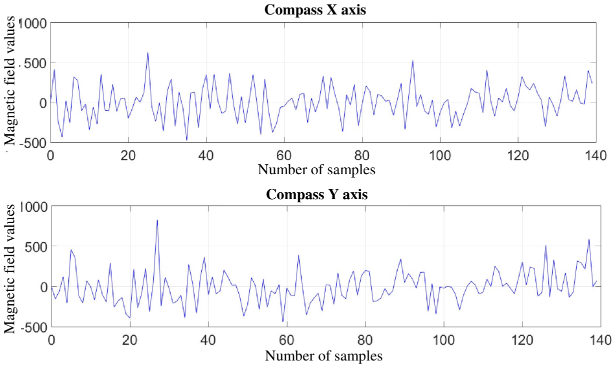

Fig. 17. Raw data for magnetic compass in X and Y axes relative to the path of Fig. 14.

In this context, since the robot circulated in 3 main directions, it was expected that data readings, at least in the X and Y axis, to be approximated of three different values, which means those 3 most significant changes of direction. Excluding the possibility of a malfunction in the sensor, this set of data errors may be due to the fact that the interior of the robot is filled with elements that can generate parasitic magnetic fields and corrupt the readings that should be relative to the Earth's magnetic field. Some of the components that may cause reading errors are the electric motors of the wheels, electric power cables, battery and wireless modem.

4) *Wheel Encoders*

This data contains the values of the encoders of the left and right front wheels. With them, it can be known how much each wheel has turned between each verification. Thus, it is given a value between 0 and 32767 which will be added the number of steps walked by the encoders between verifications. Knowing that one full turn of a wheel corresponds to 150 steps [15] and that the wheels have a diameter of 27 cm, it is possible to determine the traveled distance in meters.

The main problem related with the use of encoders is the wheel sliding verified during the motion of the robot. In several tests, it was confirmed that this slip can be significant, especially when the robot changes the direction of travel and performs curves. This is due to the high torque which is characteristic of electric motors, the terrain characteristics, and the robot weight that is not sufficient to make a convenient tire adherence to the ground. This slip will severely compromise the calculation of distance traveled and the definition of orientation changes.

5) GPS sensor

The GPS sensor data is very important for position estimation performed by the algorithm since it is the reference that place the robot travel on a referenced ground. This data is based on the NMEA 0183 protocol and the string starting with "$GPRMC" is one of the structures defined by the protocol and is the one that is relevant for the algorithm, because it contains the data obtained by the GPS position of the robot.

With the acquisition of new GPS coordinates it is also possible to determine the direction of motion of the robot relative to the Earth reference, i.e., it is possible to know whether the robot is moving. As mentioned before, the data obtained by the GPS sensor is affected by several factors that are likely to compromise the location based on this information. The determination of the instant direction of the robot depends directly on the received GPS sensor data, and suffers from the possible lack of accuracy of it. If the accuracy of GPS sensor data is affected, the obtained position could be moved further few meters of real and even "jump" to different locations around the true position, which will cause very sharp changes in the orientation of the vehicle movement even if it is stopped. Because of these conditions, it is necessary to find a balance between the data provided by dead reckoning sensors and GPS that is the one which gives the reference points necessary to frame the remaining information. The use of Kalman filters will be decisive in the definition of this balance, where merging the data from multiple sensors is necessary.

D. *Developed Matlab Algorithm*

The base model for the development of this algorithm is the one described in [18]. This paper proposes the use of linear Kalman filter for filtering the error between the odometer data (INS - Inertial Navigation System) and GPS data and feedback this filtered error in the odometric location data. In Fig. 18 it can be seen a scheme of that algorithm.

This direct feedback methodology is applied in this algorithm to estimate the instantaneous position of the Jaguar robot.

Depending on the type of data that calls the Matlab algorithm (data received from GPS or data from encoders), it will be held the corresponding processing method.

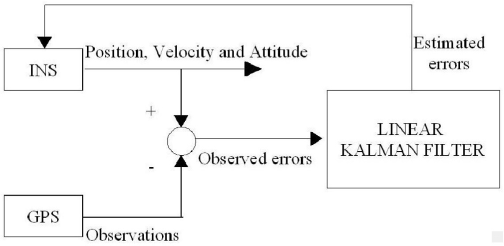

Fig. 18. Direct feedback model proposed in [18] for GPS / IMU data merging.

1) Wheel encoders data processing

Whenever the algorithm receives the encoder information, it computes how much the wheels have moved since the last measurement received and uses the orientation determined by the GPS to differentiate the offset in longitude and latitude. This allows the addition of information to a 2D representation of the path traveled by the wheels. This approach is used if the algorithm has obtained GPS data in the previous iteration, what cannot always happen because the data from the encoders is obtained with greater frequency than the GPS. If the information obtained in the previous iteration has not been GPS, it is likely that the orientation is outdated. In this case, are used the values of the encoders of both wheels to determine the difference in distance traveled by each one, and know the variation of robot orientation. After determination of the distance variation in meters covered by the robot in latitude and longitude, the coordinates are converted to decimal degrees and returned to the navigation program.

2) GPS data processing

Whenever GPS data is received, the algorithm compares the last determined position (obtained by the encoder) with the now received GPS position. It is then calculated the latitude and longitude difference (error) of the two positions and these errors are filtered with a 1D Kalman filter. The latitude and longitude filtered errors are now subtracted from the latitude and longitude of the estimated current position values and this gives a pair of coordinates that contains the filtered and updated information about the current robot position.

The algorithm ends with the return of the upgraded estimated position of the coordinate pair to the navigation program.

E. *Experimental Results*

The tests were performed at 45% of maximum power, which limits the maximum speed of the robot to about 6 km/h, on flat and hard ground (road). This allows to save battery power during testing and allows to follow the robot during the course navigation and quickly stop it if needed. It also prevents excessive heating of the wheel motors, and provides more time for information processing.

Test 1

In this test, illustrated in Fig. 19, is traveled a distance of about 220 m. The green line indicates the route previously defined, that the robot should follow. The blue is the position determined directly by the GPS (with its associated errors) and in red is the position estimated by the algorithm. The route starts at the position marked A and ends in position B.

Fig. 19. Performed Route for test 1.

It can be seen that the robot was able to accomplish this route without deviating more than 1 m from the desired path during the entire route. The major disorders in the followed route (such as the curve at the bottom of the image) are associated with disturbances in the GPS signal due to the trees that exist above the path. The rest of the course has sharp variations, which are due to the subtractions of the filtered error between the GPS obtained positions and those calculated with the encoders.

Note: the position indicated on the maps does not correspond exactly to that position in the ground, the traveled route is always in the center of the tracks and not at the extremes, as represented.

Test 2

In this test, the case of the test 1 is repeated but with an angle difference margin of 3º (Fig 13 – instead of 1º in test 1), giving to the robot a bigger error margin in curve performing. In Fig. 20 it can be seen the followed path.

Given the changes made, the effect reflects especially in the reduction of the amount of steering corrections that the robot performs during the journey. However, in certain situations, a higher deviation may occur on the path.

Fig. 20. Performed Route for test 2.

Test 3

This time, as illustrated in Fig. 21, the robot is tested in a significantly longer path than in previous tests. This route is approximately 620m long and was accomplished in about 10 minutes starting at A and ended in B.

Fig. 21. Performed Route for test 3.

It can be seen that the robot could make the entire route without excessive deviation from the desired path. This test demonstrates the ability of the navigation algorithm to successfully cross places with GPS signal shadow (caused by the zones of trees and buildings) and handle errors and noise in the information from the sensors. Also it was possible to maintain the system stability and operation all the time.

F. Conclusions

The tests allowed to verify the proper functioning of the work. Nevertheless, it is not error-free nor can be considered ideal, so there is always something more that can be done or improved. However, the purpose of this work has been fully achieved, since the results are quite satisfactory.

The inability to use the magnetic compass data was crucial, because they would have been very useful in the navigation process. Initially it was thought that this data could be used, so, it was necessary to spend some time attempting to properly get and process the information.

As expected, the amount of data involved and the high number of adjustable parameters in the algorithms made the process of adaptation and adjustment of the navigation system slower requiring a lot of tests and revisions to the developed processes.

Also the availability of the Military Academy space was very important for the achievement of numerous tests required, which could not have been carried out in another public space by the risk involved for the robot and other users of this space. The robustness of the robot and quality / stability of the sensors was also decisive, since there was no need to solve additional problems (possibly intermittent) related to hardware or instability in sensors, enabling better observation of the changes for a particular parameter without being confused with any other detail. Highlights also to the importance of program Google Earth that allows the registration of robot movements and navigation results and the program Matlab that greatly helps in the manipulation of numerical data and management of the vast information generated.

Future work

As future work it is suggested the adaptation of an external compass to the original robot system that is free of magnetic interference.

If it were possible to frame the motion of the robot with magnetic compass data, it is possible to estimate its position without dependence on GPS, which was only necessary to update the position coordinates. In the case of this work, the calculated orientations had lack of precision because they are always dependent on the GPS that has no sufficient resolution to correctly interpret the small movements of the robot. Another interesting change require the development of the algorithm implemented in Matlab in an executable format (Dynamic-link library (DLL), for example), allowing the use of the algorithm independently of the Matlab environment and enabling faster execution of instructions and the reduction of requirements for the computer. Thus it would be closer the possibility of implementing the entire system in a small control unit (System on Chip) that could be loaded onto the robot making it completely autonomous and independent of a computer permanently located within range of the wireless network.

REFERENCES

[1] M. I. Ribeiro, *Kalman and extended kalman filters: Concept, derivation and properties*, Institute for Systems and Robotics, 2004, 43.

[2] Cornell, *Kalman filter tank filling, subject MI63*, 2008.

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

[4] D. Simon, “Using nonlinear kalman filtering to estimate signals,” *Embedded Systems Design vol. 19 no. 7*, 2006, p. 38.

[5] S. Y. Yi and B. W. Choi, “Autonomous Navigation of Indoor Mobile Robot Using Global Ultrasonic System”, *Mobile Robots: Perception and Navigation,* 2007.

[6] J. J. Leonard and H. F. Durrant-Whyte, *Directed Sonar Sensing for Mobile Robot Navigation,* Kluwer Academic Publishers, 1992.

[7] J. H. Ko, W. J. Kim, and M. J. Chung, “A method of acoustic landmark extraction for mobile robot navigation”, *IEEE Trans. Robotics and Automation,* 12(3), 478–485, 1996.

[8] J. J. Leonard and H. F. Durrant-Whyte, “Mobile robot localization by tracking geometric beacons” *IEEE Trans. Robotics and Automation*, 7(3), pp. 376–382, 1991.

[9] S. Hernández, J. M. Torres, C. A. Morales, and L. Acosta, “A New Low Cost System for Autonomous Robot Heading and Position Localization in a Closed Area”, *Autonomous Robots*, 15(2), pp. 99–110, 2003.

[10] Ning, Xiaolin, & Fang, Jiancheng. 2008. Spacecraft autonomous navigation using unscented particle filter-based celestial/Doppler information fusion. Measurement Science and Technology, 19(9), 095203.

[11] R. Siegwart and I. R. Nourbakhsh, “Introduction to Autonomous Mobile Robots”, *The MIT Press,* 2004.

[12] L. F. Melo and A. F. Junior. "Mobile robot indoor autonomous navigation with position estimation using rf signal triangulation", 2013.

[13] J. Filipe, “Piloto Automático Sistema de Navegação Autónoma” M.S. thesis, Instituto Superior Técnico, Lisboa, Portugal, 2013.

[14] Dr Robot Inc. (2016, October 3). WiFi 802.11 robot, Network-based Robot, robotic, robot kit, humanoid robot, OEM solution [Online]. Available: http://jaguar.drrobot.com/images/jaguar_4x4w_photo_1280_09.jpg

[15] Dr Robot Inc. (2016, October 3). WiFi 802.11 robot, Network-based Robot, robotic, robot kit, humanoid robot, OEM solution [Online]. Available: http://jaguar.drrobot.com/Download.asp

[16] Garmin (2016, October 3). GPS 18x 5Hz™ | Garmin [Online]. Available: https://buy.garmin.com/en-US/digital/oem/sensors-and-boards/gps-18x-5hz-/prod13195.html

[17] SparkFun Electronics (2016, October 3). 9 Degrees of Freedom - Razor IMU - AHRS compatible - SEN-10125 [Online]. Available: https://www.sparkfun.com/products/retired/10125

[18] C. Cappelle, D. Pomorski, and Y. Yang, “GPS/INS data fusion for land vehicle localization”, *Proceedings of the IMACS Multiconference on Computational Engineering in Systems Applications*, Beijing, China, vol. 46, pp 2127, 2006.