International Journal of Engineering and Technology Innovation, vol. 5, no. 3, 2015, pp. 141-155

A Complete Design Flow of a General Purpose Wireless GPS/Inertial Platform for Motion Data Monitoring

Gianluca Borgese[1], Calogero Pace[1,*], Luigi Rizzo[2], Giuseppe Artese[3], Michele Perrelli[3], Roberto Beneduci[4]

[1]Department of Computer Engineering, Modeling, Electronics and Systems Science (DIMES), University of Calabria, Arcavacata di Rende (CS), 87036, Italy.

[2]AIEM s.r.l., Research & Development Area, Zumpano (CS), 87036, Italy.

[3]Department of Town-and-Country Planning, University of Calabria, Arcavacata di Rende (CS), 87036, Italy.

[4]Department of Physics, University of Calabria and INFN Gruppo c. Cosenza, Arcavacata di Rende (CS), 87036, Italy.

Received 03 December 2014; received in revised form 29 July 2015; accepted 30 July 2015

Abstract

This work illustrates a complete design flow of an electronic system developed to support applications in which there are the need to measure motion parameters and transmit them to a remote unit for real-time teleprocessing. In order to be useful in many operative contexts, the system is flexible, compact, and lightweight. It integrates a tri-axial inertial sensor, a GPS module, a wireless transceiver and can drive a pocket camera. Data acquisition and packetization are handled in order to increase data throughput on Radio Bridge and to minimize power consumption. A trajectory reconstruction algorithm, implementing the Kalman-filter technique, allows obtaining real-time body tracking using only inertial sensors. Thanks to a graphical user interface it is possible to remotely control the system operations and to display the motion data.

Keywords: GPS, Kalman filter, MEMS inertial sensors, wireless communication

1. Introduction

Nowadays, in many research fields such as body motion recognition (BMR), fall detection (FD), aerial photogrammetry (AP), inertial navigation (IN), etc, there is the necessity to acquire and wireless transmit all body motion parameters (axial accelerations, angular rates, global position, speed, etc.) to a remote host system for tracking and control purposes. With regard to BMR [1], [2] and FD [3], there are several areas of interest (e.g.: 3D virtual reality, biomedical applications, robotics) in which it is extremely important to detect human body movements, in order to measure, recognize or reproduce them using a robot. AP [4] and IN [5], [6] fields are older than BMR and FD, but a number of different and innovative applications can still be found, such as pedestrian navigation in harsh environments [7], agriculture automated vehicles [8], [9], or animal motion analysis [10]. Although several kinds of similar systems can be found in the market [11], [12], they are usually highly specialized for a particular application and not very flexible. Some systems use high performance and high cost devices, others are not wireless-based or are too heavy. The main purpose of this work was to describe the whole top-down design flow of a low-cost, complete and flexible system which can be customized for several applications. This system should be powerful, compact and lightweight.

*Corresponding author. E-mail: calogero.pace@unical.it

2. System Design Strategy

To reach these features, it is necessary to carefully design the system architecture and to select the components in order to save space and to decrease system weight as much as possible. In the market, there are many kinds of high performance inertial measurement units (IMU), such as the HoneyWell HG900, but they do not fulfil our requirements of small size, low weight and low cost. In our prototype, we chose the ADIS16350 module, a MEMS IMU that integrates a tri-axial accelerometer and a tri-axial gyroscope. This IMU is a strapdown type system which is intrinsically compact, highly integrated and low-cost, even if of limited accuracy. Table 1 shows the ADIS16350 characteristics w.r.t. HG900 ones. We chose to battery-operate the system using two rechargeable NiMh AAA cells. Two high-efficiency switching step-up voltage regulators convert the 2-2.4V input voltage range in the required output voltage levels: 5V and 3.3V. In order to efficiently handle and transmit motion data, it is important to exploit the available wireless transmission band organizing data in packets [13]. In addition to hardware system side, a graphical user interface (GUI) has been developed for the remote PC-based receiver in order to control system operations, set inertial sensor parameters (offset, calibration, alignment, etc), display motion variables progress, track trajectories, drive the camera, etc. A Kalman-based trajectory reconstruction algorithm is implemented in the remote PC software for supporting applications such as inertial navigation or motion parameters detection.

Table 1 Comparison between HG900 and ADIS16350 IMUs

IMU Name	HG900 Honey Well	ADIS16350 Analog Devices
Typology	Laser	MEMS
Gyros Bias (1σ) [°/hr]	<0.003	54
Gyros Random Walk [°/√hr]	<0.002	4.2
Accelers Bias (1σ) [mg]	<0.025	0.7
Accelers Random Walk [m/s/√hr]	0.0143	2.0
Acc_Bias Pos. Error (1hr) [km]	~ 1.59	~44.5
Size [mm]	139.7x162.6x135.6	23.2x22.7x23.3
Weight [kg]	<3.0	<0.016
Power Consumption [W]	<10.0	<0.285
Price [k$]	~100	~0.6

3. System Architecture

The system printed circuit board hosts two subsystems: System Control Block (SCB) and Power Management Block (PMB). SCB manages all control operations, acquiring data from inertial sensor and GPS module, sending data packets to the host pc and receiving command packets from it; PMB provides the two supply voltage levels to SCB. Fig. 1 shows system architecture together with power and communication connections. Fig. 2 is a picture of the system prototype.

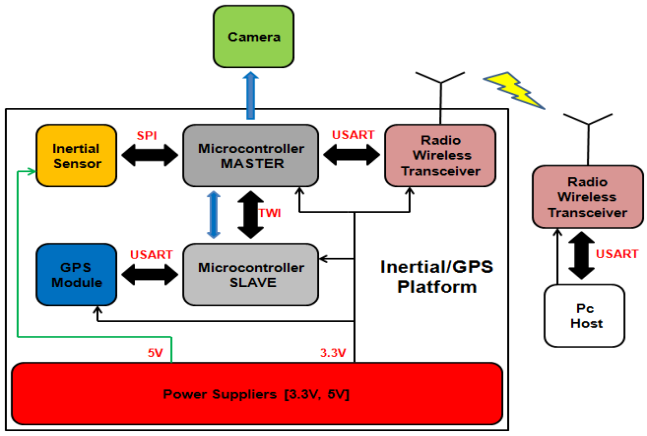


Fig. 1 Complete system block diagram

3.1. Power Management Block

The Power management block is constituted by two step-up converters (Maxim MAX756) which allow to provide both and 3.3V voltage guaranteeing up to 400mA load current and an efficiency of about 85% (input voltage over 2.2V). As said before, the two rechargeable NiMh battery packs have a nominal voltage of 2.4V and a nominal capacity of about 2650mAh. Considering a nominal input energy of about 6.36Wh against a max required load power of about 625mW (worst case), our system has an autonomy of about 9 hours (experimentally verified).

3.2. System Control Block

The system control block is the control core of whole system. The modules on the board are: two 8 bit microcontrollers (Atmel ATMEGA8) (defined as Master and Slave), an inertial sensor (AnalogDevice ADIS16350), a GPS module (Fastrax UP500), and a wireless transceiver (Maxstream XBEE). To support some applications such as aerial or ground photogrammetry, a pocket camera (Canon SX200IS) was interfaced through its USB port. Main features of these devices are:

(1) *ADIS16350*: is a low-power (165mW @ 5V) complete inertial measurement station. It is constituted by one tri-axial accelerometer, one tri-axial gyroscope and a triple thermometer for thermal compensation. It transfers inertial data with 14 bit resolution to the output registers, accessible via a 2MHz SPI interface, at a maximum sample rate of 819.2Hz (350Hz bandwidth). The inertial sensors are precision aligned across axes, and are calibrated for offset and sensitivity.

(2) *UP500*: is a low-power (90mW @ 3V) GPS receiver module with embedded antenna and fix rate up to 5Hz. Communication is based on NMEA protocols, via RS232 link up to 115.2kbps. It supports WAAS/EGNOS correction to improve position resolution up to about 2m.

(3) *XBEE*: is a low-power (165mW@ 3.3V) 2.4GHz transceiver which implements ZigBee™ protocol and has a transmission range of about 80m. Transmission and reception buffers allow efficient data stream packetization, also required to reach the rated communication speed because every data exchange requires the presence of an about 20 bytes long header. It is interfaced through RS232 protocol up to 115.2kbps. As we can see in Fig. 1, Master microcontroller is connected to ADIS16350 through SPI interface, to XBEE through USART interface, to Slave microcontroller through TWI interface and to the high resolution camera by means of a digital output pin. The Slave microcontroller is connected only to UP500 by means of USART interface and to Master microcontroller as said before. A Slave output pin is used to send an interrupt to Master when a new GPS frame is ready. Master and Slave are clocked with two 14.7654MHz quartz.

3.3. Pocket Camera

We used a low-cost 12.1 Mpixels Canon SX200IS camera (5-60mm lens focus, 4X digital zoom, 12X optical zoom, shutter speed 1s -1/3200s). The firmware was updated with an unofficial version in order to acquire full control of the camera functions. In particular, we exploited the possibility to remotely shot photos applying a 3V pulse to the USB port and to store photos in uncompressed format (RAW), as required for photogrammetry applications. For georeferencing each picture, a progressive number, corresponding to the file number on the memory card, is recorded on the inertial data frame.

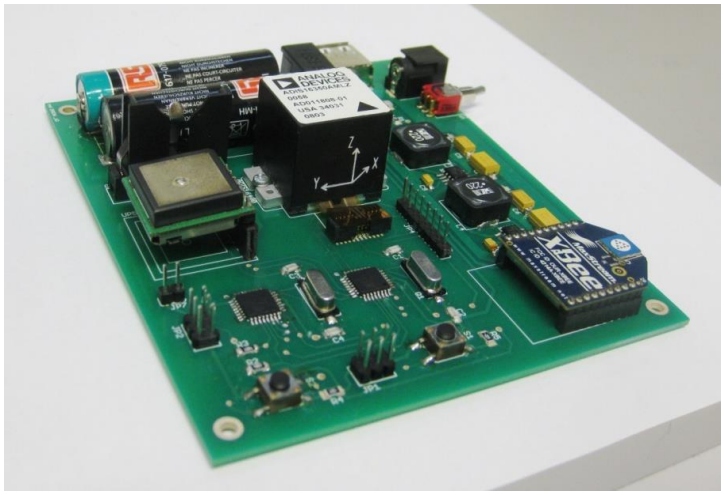


Fig. 2 Picture of the system prototype

4. System Working and Data Protocol

Thanks to a simple but complete remote GUI, the PC-host can start every system operation, as will be explained in next sections. There are three kinds of command packets that can be sent to the system:

(1) Operation request (GPS/Inertial data readout, photo shooting, offset readout);

(2) Configuration setting;

(3) Configuration readout.

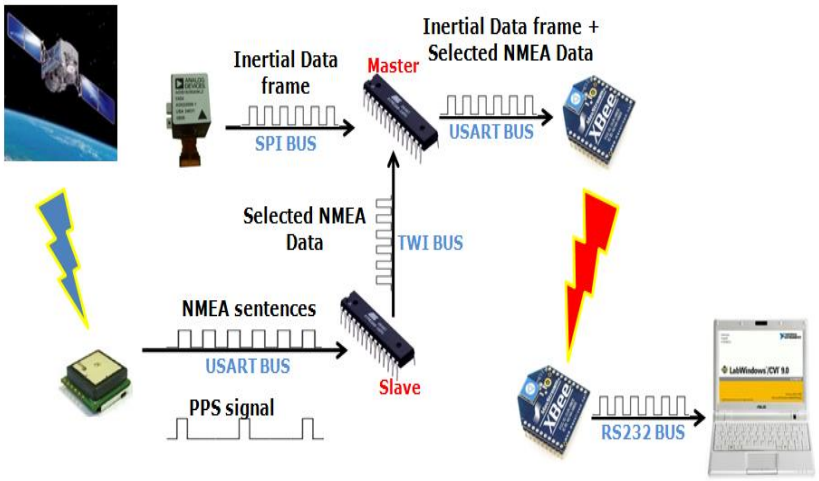


Fig. 3 System operations

Every command packet is identified by means of different opcodes. The system requirement was to transmit synchronized data from inertial sensor, operating at 100Hz, and GPS module operating at 5Hz (Fig. 3). The inertial sensor sample rate is important to get a good position resolution in case of trajectory tracking calculations. Hence the data stream has to contain 20 inertial frames plus one GPS frame every 200ms. The inertial data frame is 20 bytes long and contains the following fields: supply voltage, x/y/z temperatures, x/y/z angular rates, x/y/z linear accelerations. The sensor has to be read by the Master every 10ms and this is guaranteed by a dedicated hardware timer of the microcontroller. A problem is posed by the verboseness of GPS serial message: in fact, NMEA sentences contain hundreds of bytes. So we had to select only the necessary information, in order not to compromise the desired data-rate. To this aim, at start-up, the slave microcontroller initializes the GPS module to send only four sentences:

(1) GGA: Global Positioning System Fix Data;

(2) GSA: GPS DOP and active satellites;

(3) VTG: Track Made Good and Ground Speed;

(4) RMC: Recommended Mininum Navigation Information.

These NMEA sentences contain main information which can be useful for different applications. The Slave creams off the received sentences and stores in RAM only the information to display, i.e. a total of 72 bytes. Even if reduced in this way, the time required to send such information is still too high (about 6.25ms) in order not to compromise the regularity of the inertial sensor reading. So we decided to divide the GPS answer in 8 packets of 9 bytes and to send, every 20 ms, two inertial frames plus a GPS packet. So, in 200ms, we send 8 frames of 51 bytes (frame number, 2 inertial frames, 1 GPS packet, photo number) and last 2 frames of 42 bytes (frame number, 2 inertial frames, photo number) as shown in Fig. 4. Data acquired from PC are reconstructed, displayed and stored in a text file for further elaboration; GPS data are also processed at run-time to display the trajectory. The frame number is used to identify each frame within a second (50 frames/s) and is used for:

(1) Reconstruction of GPS information;

(2) Identification of any frame lost in reception.

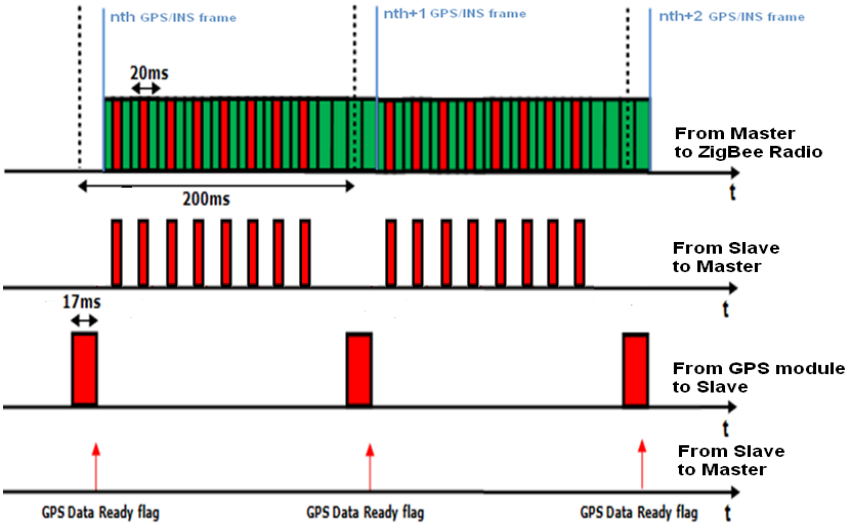


Fig. 4 GPS/Inertial data timing (in red the GPS data, in green the inertial data)

Finally, the photo number allows for the association of picture files in the SD card with time, position and attitude of the camera. The complete system protocol is better explained in the flow chart of Fig. 5. After the reception of a data request from host pc, master microcontroller sends a GPS data request to slave microcontroller and waits for response checking the GPS data ready flag. When slave acquires and creams off a GPS frame, it sets the GPS data ready flag so that master starts a 10ms timer up and acquires an inertial frame storing it on RAM. Then master asks slave a single GPS packet which is received on TWI line and immediately stored on RAM. When 10ms timer stops, master acquires a second inertial frame storing it on RAM. In the end master sends the two inertial frames and a GPS packet to XBEE module which sent them to host-pc. When these operations are over, master restarts 10ms timer and begin a new operation cycle. If there is an interruption of GPS operations, master continues to send to host-pc only inertial frames respecting the 10ms timing.

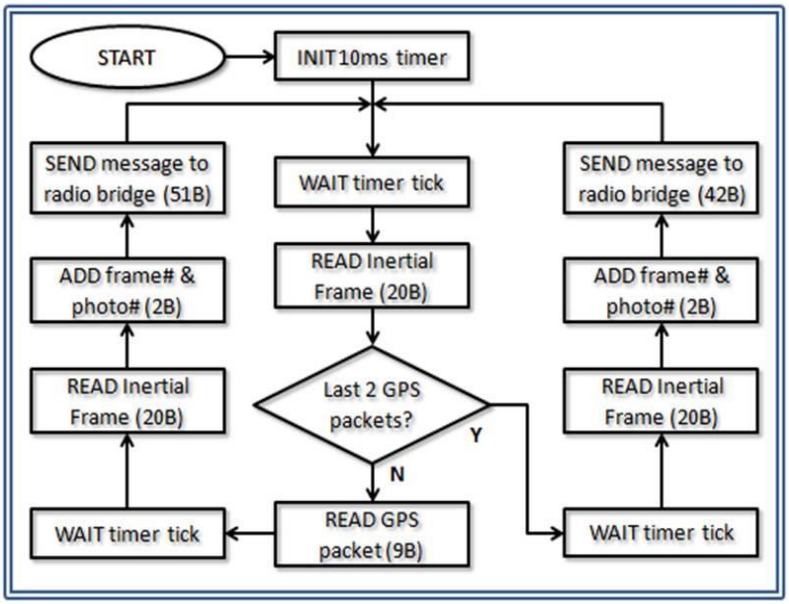


Fig. 5 Data protocol flow chart

5. Inertial Data Elaboration

Data acquired from the inertial sensor can be processed to obtain position and orientation of a body and to track a three dimensional trajectory. This technique is called inertial navigation and it is used in a wide range of applications.

Inertial data are processed following the scheme [14] in Fig. 6. where:

(1) *U_acc*: signals from accelerometers;

(2) *U_omega*: signals from gyroscopes;

(3) *a*: linear acceleration;

(4) *v*: linear velocity;

(5) ω: angular velocity;

(6) *C*: rotation matrix.

The subscripts *b* denote the body coordinate system (that is the navigation system's reference frame) while the subscripts *n* denote the local coordinate system (in which the body move). The first step of trajectory reconstruction algorithm is the correction of accelerometers and gyroscopes signals. The correction of errors on signals is the most important step of algorithm, because errors influence overall system performance [15]. In particular, propagation of orientation errors caused by noise, perturbing gyroscope signals, is identified as the critical cause of a body position drift. The main cause of errors are: scale factor, bias, drift, temperature, non-orthogonality. In order to compensate them it is necessary to perform a procedure of calibration. A first coarse calibration was executed using the automatic calibration of ADIS16350 managed from remote GUI software. Then a finer calibration was conducted manually. Among all calibration methods proposed in literature, the most appropriate calibration technique for low-cost sensors is the "*modified multi-position calibration method*" [16-17]. Its aim is to find all calibration parameters (bias, scale factor, non-orthogonality, etc) of sensors. It consists in laying out sensors in different linearly independent positions in order to define a system of linearly independent equations which outnumbers the set of calibration

parameters to find.

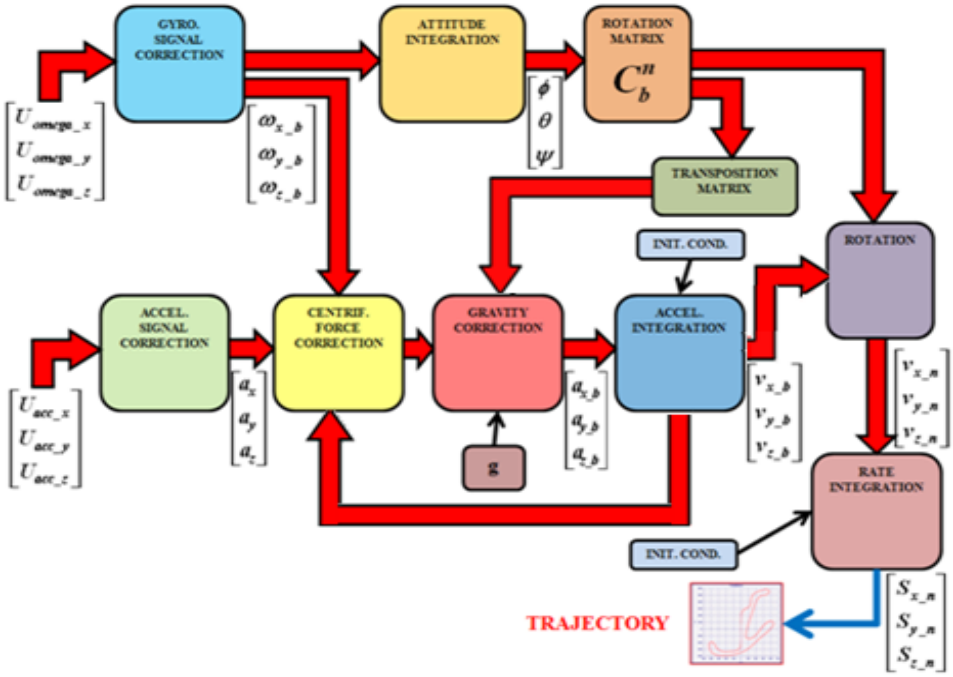


Fig. 6 Block diagram of the trajectory reconstruction algorithm

The linear acceleration and angular velocity error can be modeled as Eqs. (1)-(2):

$$\delta a = b_{acc} + \lambda_{acc} a + c_{acc\_T}\left(T - T_0\right) + \nu_{acc} \tag{1}$$

$$\delta \omega = b_{gyro} + \lambda_{gyro} \omega + c_{gyro\_T}\left(T - T_0\right) + \nu_{gyro} \tag{2}$$

where: b_{acc} and b_{gyro} are the sensor bias; λ_{acc} and λ_{gyro} are the sensor scale factors; c_{acc_T} and c_{gyro_T} are the sensor thermal constants; ν_{acc} and ν_{gyro} are the sensor measurement noises, $\nu_{acc} = \sigma_{acc} * \sqrt{sample\_rate}$, $\nu_{gyro} = \sigma_{gyro} * \sqrt{sample\_rate}$ and, σ_{acc} and σ_{gyro} are noise density, and T and T_0 are the temperatures during the measurement and at sensor start-up respectively.

In Table 2, there are the calibration parameters obtained according to refs. [16] and [18].

Table 2 Calibration parameters

Parameter	Value
b_{acc_x}	0.012133g
b_{acc_y}	0.023295g
b_{acc_z}	-0.03593g
b_{gyro_x}	0.3766°/s
b_{gyro_y}	0.1963°/s
b_{gyro_z}	0.6270°/s
λ_{acc_x}	0.00775
λ_{acc_y}	0.008838
λ_{acc_z}	0.008041

Table 2 Calibration parameters (cont.)

Parameter	Value
λ_{gyro_x}	0.004818
λ_{gyro_y}	0.004042
λ_{gyro_z}	0.009385
c_{acc_T}	4mg/℃
c_{gyro_T}	0.1°/s/℃
v_{acc}	1.85mg $\sqrt{Hz}$
v_{gyro}	0.05°/s $\sqrt{Hz}$

After the calibration phase, it is necessary to compensate the centrifugal acceleration and the acceleration of gravity effects obtaining accelerations in body coordinate system .The former is compensated subtracting the vector product between angular velocities (from gyroscopes) and linear velocities (from numerical integration of accelerations), the latter is compensated adding the scalar product between transposed rotation matrix and the gravity acceleration. After a numerical integration, velocities in body coordinate system are obtained. In order to pass to local coordinate system, the linear velocities are multiplied by the rotation matrix and then are integrated to have body trajectory. The angular velocities are also integrated, obtaining the information about the orientation (Euler angles) (3-5) and the rotation matrix (for transformation from b-frame to n-frame) (6), (Table 3). The equations to integrate and the rotation matrix A are:

$$\dot{\phi} = \left(\omega_{y\_b}\sin\phi + \omega_{z\_b}\cos\phi\right)\tan\theta + \omega_{x\_b} \tag{3}$$

$$\dot{\theta} = \left(\omega_{y\_b}\cos\phi - \omega_{z\_b}\sin\phi\right) \tag{4}$$

$$\dot{\psi} = \left(\omega_{y\_b}\sin\phi + \omega_{z\_b}\cos\phi\right)\sec\theta \tag{5}$$

Table 3 Rotation matrix elements

Matrix Element	Value
A_{11}	$\cos\theta\cos\psi$
A_{12}	$-\cos\theta\sin\psi + \sin\phi\sin\theta\cos\psi$
A_{13}	$\sin\theta\sin\psi + \cos\phi\sin\theta\cos\psi$
A_{21}	$\cos\theta\sin\psi$
A_{22}	$\cos\theta\cos\psi + \sin\phi\sin\theta\sin\psi$
A_{23}	$-\sin\phi\cos\psi + \cos\phi\sin\theta\sin\psi$
A_{31}	$-\sin\theta$
A_{32}	$\sin\phi\cos\theta$
A_{33}	$\cos\phi\cos\theta$

$$A = \begin{pmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{pmatrix} \tag{6}$$

where transformation from reference axes to a new frame is expressed as: (1) Rotation through angle ψ about reference z-axis; (2) Rotation through angle θ about new y-axis; (3) Rotation through angle ϕ about new x-axis.

However, also with a tight correction of errors, it is not possible to obtain great position accuracy for long time using only MEMS IMU, but it is necessary to include information from GPS module, integrated in our system. Inertial and GPS modules are complementary: the former is characterized by high measurement frequency, but short-term accuracy while the second by long-term accuracy but low measurement frequency. The main idea is to be able, in the further system versions, to precisely reconstruct the trajectory by means of the high-rate inertial data acquired between two GPS acquisitions and then to correct accumulated errors using the low-rate but accurate information coming from the GPS module. The Kalman filter is the most used algorithm for this purpose. In literature, there are several implementations of Kalman filter depending on the features of devices [19]-[20]. To be able to correctly integrate Inertial and GPS data, it is important to have high synchronization between data acquisitions. In the present prototype, limited to the gyroscopes and accelerometers, the implementation of a typical Kalman filtering is included into remote PC and its variance, and noise parameters are chosen on the basis of empirical measurements and calibration.

6. GPS Data Handling

In order to plot a GPS trajectory in a two dimensional graph, it is necessary at first to transform GPS geodetic coordinates (longitude λ, latitude ϕ, height h) to ECEF (Earth-Centered-Earth-Fixed) coordinates (X_e, Y_e, Z_e) and then to NED (North-East-Down) coordinates (x_n, y_n, z_n) according to (7-9) equations [21]. $N(\phi)$ is the grand normal (the largest radius of curvature of ellipsoid defined at a specified latitude) that is the distance from the surface to the Z-axis along the ellipsoid normal. a is the lenght of the semi-major ellipsoid axis and e is the first numerical ellipsoid eccentricity. $R_{n/e}$ is a transformation matrix (12) from ECEF to NED coordinates. X_{er}, Y_{er}, Z_{er} are ECEF reference coordinates.

$$X_e = \left[N(\phi) + h\right]\cos\phi\cos\lambda \tag{7}$$

$$Y_e = \left[N(\phi) + h\right]\cos\phi\sin\lambda \tag{8}$$

$$Z_e = \left[N(\phi)\left(1-e^2\right) + h\right]\cos\phi\cos\lambda \tag{9}$$

$$N(\phi) = \frac{a}{\sqrt{1-e^2\sin\phi}} \tag{10}$$

$$\begin{pmatrix} x_n \\ y_n \\ z_n \end{pmatrix} = R_{n/e}\begin{pmatrix} X_e - X_{er} \\ Y_e - Y_{er} \\ Z_e - Z_{er} \end{pmatrix} \tag{11}$$

$$R_{n/e} = \begin{pmatrix} -\sin\phi\cos\lambda & -\sin\phi\sin\lambda & \cos\phi \\ -\sin\lambda & \cos\lambda & 0 \\ -\cos\phi\cos\lambda & -\cos\phi\sin\lambda & -\sin\phi \end{pmatrix} \tag{12}$$

7. Remote GUI

The Remote GUI is developed using LabWindows© development environment based on C language. The GUI allows managing every system operation. As seen in Fig. 7 in the window, there are three main sections: a graph section to display GPS trajectory, angular velocity and linear acceleration; a boxes section to show inertial sensor parameters (supply voltage, x-y-z

linear accelerations, angular velocities and temperatures) and GPS parameters (time, latitude, longitude, altitude above mean sea level, height of geoid above WGS84 ellipsoid, speed, heading and PDOP; a command section to initializing XBEE radio-bridge, to start/stop system operations and to shoot photos. It is also possible to save data into a text file for offline analysis. In the top of window, there is a menu in which user can access inertial sensor setting mode and manually change gyroscope dynamics, number of tapes of Bartlett FIR digital filter, sample rate, accelerometer and gyroscope offset or use automatic procedures of axial alignment, offset compensation, calibration (Fig. 8, 9). The numerical integration algorithm, the Kalman filter and the coordinate transformation are integrated into the GUI as already said.

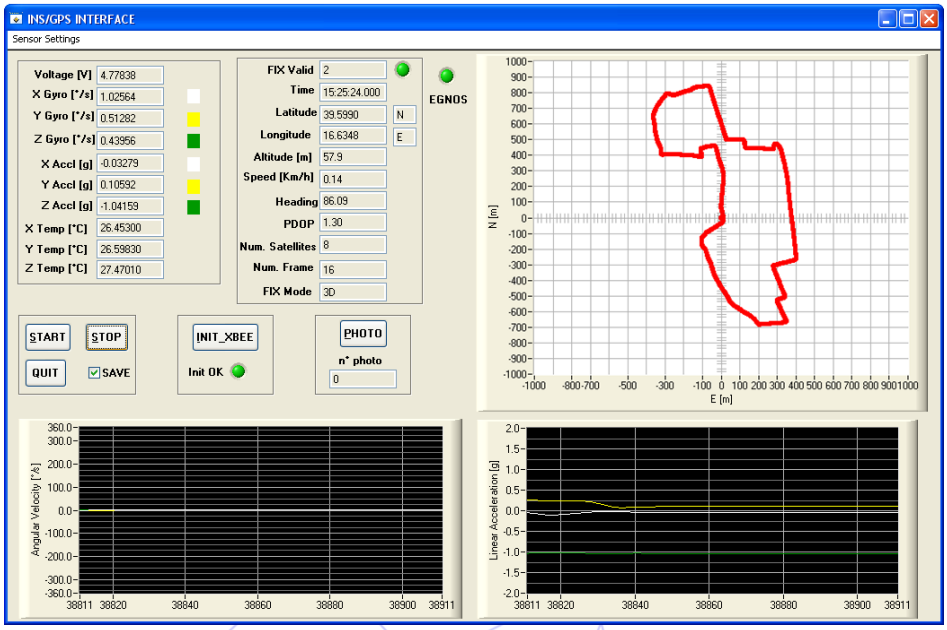


Fig. 7 System GUI with an example of GPS trajectory

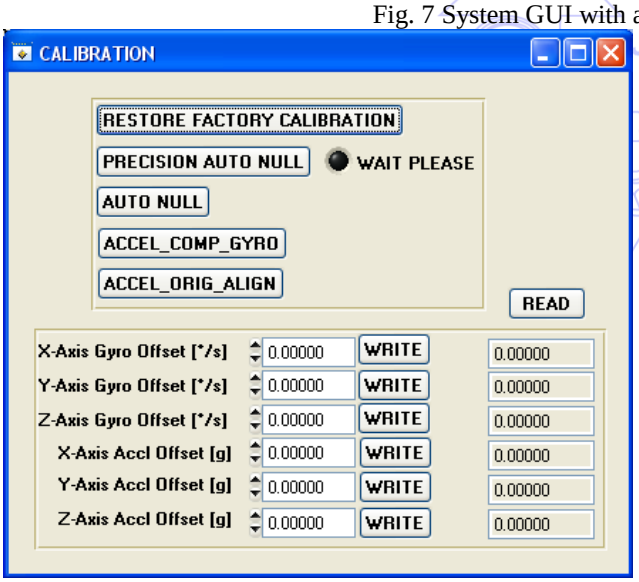


Fig. 8 Calibration sub-window

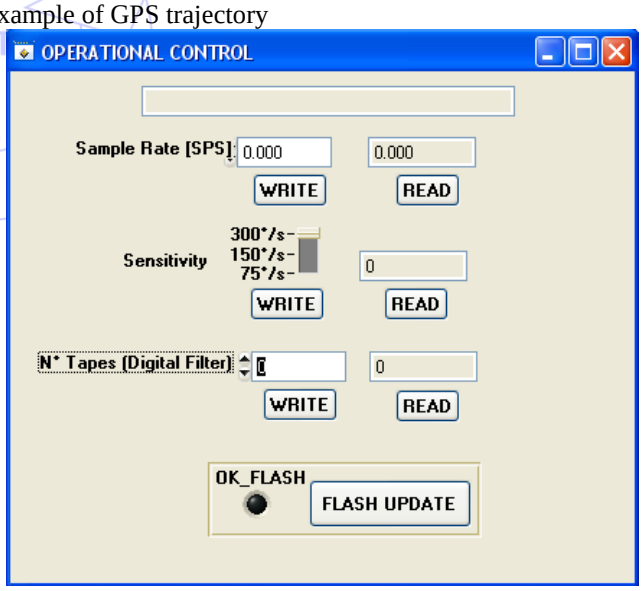


Fig. 9 Operation control sub-window

8. System Testing

In order to verify proper working of system many kind of tests are conducted on system modules.

8.1. Accelerometers/Gyroscopes Test

To test accelerometers and gyroscopes, two type of tests were conducted. In the first test, the system was placed on a strobe speed-controlled turntable with velocity of 33 rpm and 45 rpm, to evaluate biases and the correct angular velocity measured by gyroscopes; in the second test, system was placed on a radio-controlled toy car and various movements were performed to test

the performance of the whole inertial system (Fig. 10, 11, 12).

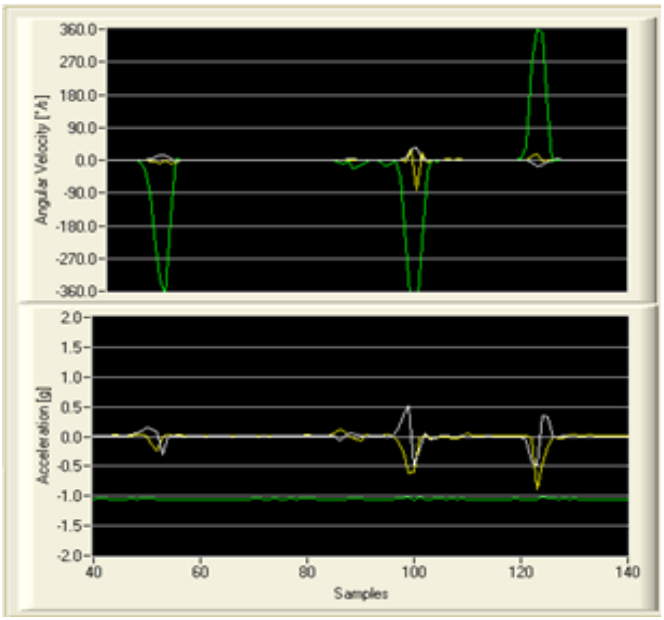


Fig. 10 Sensor responses for slewing rounds movement performed

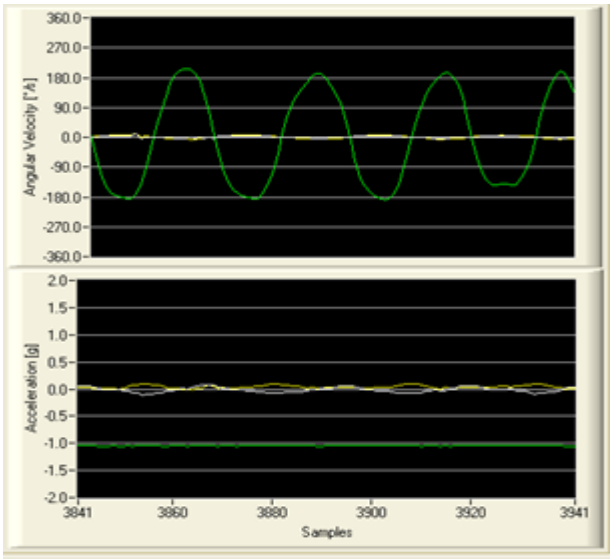


Fig. 11 Sensor responses for spins movement performed

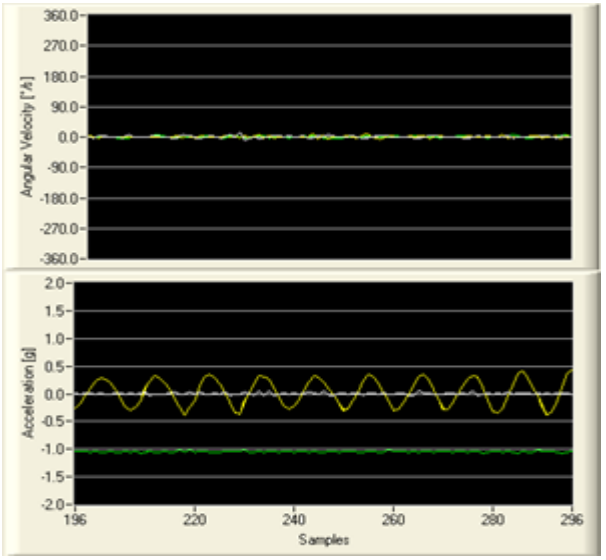


Fig. 12 Sensor responses for back/forth movement performed

8.2. GPS Module Test

Moreover, the system was mounted on a car in order to verify GPS module operations and the coordinate transformation algorithm using GPS data (Fig. 7). Another kind of test allows to analyze the proper working of GPS module along a closed path and comparing results with a high accuracy differential GPS module.

The reference GPS receiver is a Leica VIVA GNSS, used in combination with the permanent reference GNSS station at the University of Calabria. The reference station is a node of the "ITALPOS" national reference network, and it is positioned close to the path followed during the test (100 to 400 m). The data processing has been performed by using the software Leica Geo Office. A standard deviation less than 1 cm has been obtained for both planimetric and height position. Thus, the results of DGPS have been assumed as reference. From this test we valued position errors along x, y and z axis using a statistical analysis. In Fig. 14 the trajectory comparison between our GPS SBAS module and the reference differential GPS module is shown. As it is possible to see, the two trajectories (red and blue) are almost undistinguishable, due to the scale of the mapIn Table 4, there are the error distribution parameters. The mean position error is lower than 1m for x and y axis with a standard deviation lower

than 2m, only for the z axis the mean position error is of about 5m.

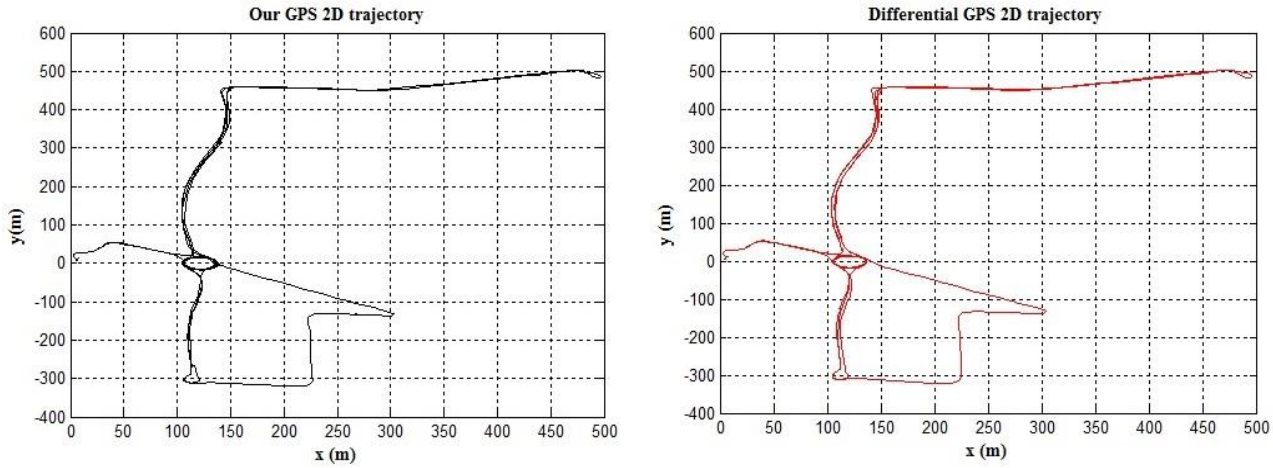


Fig. 14 Results comparison between our GPS (left) and differential GPS (right); the two trajectories are almost totally overlapped and undistinguishable

Table 4 Error distribution parameters

X-error Mean	0.573m
Y-error Mean	-0.143m
Z-error Mean	4.267m
X-error σ	1.825m
Y-error σ	1.480m
Z-error σ	1.997m

8.3. Inertial-based Trajectory Reconstruction Test

After the GPS trajectory reconstruction test, we conducted an Inertial-based trajectory reconstruction test to verify the quality of trajectory reconstruction algorithm and of Kalman filtering. For this test the strobe speed-controlled turntable was used. As it can be seen in Fig. 15 using just the reconstruction algorithm, after about 25 loop at 33rpm, there is an increasing of offset and bias which deforms the circular trajectory with a spiral divergence; with Kalman filtering the trajectory is very stable and it is evident the decreasing of x/y error as shown in Table 5 comparing to values in Table 6. The x/y position error fluctuations are shown in Figs. 16 - 17.

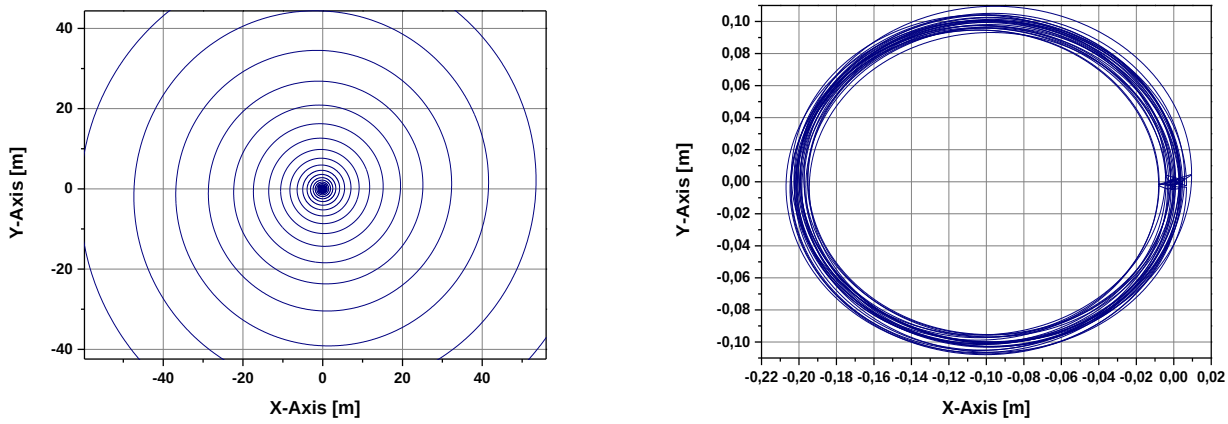
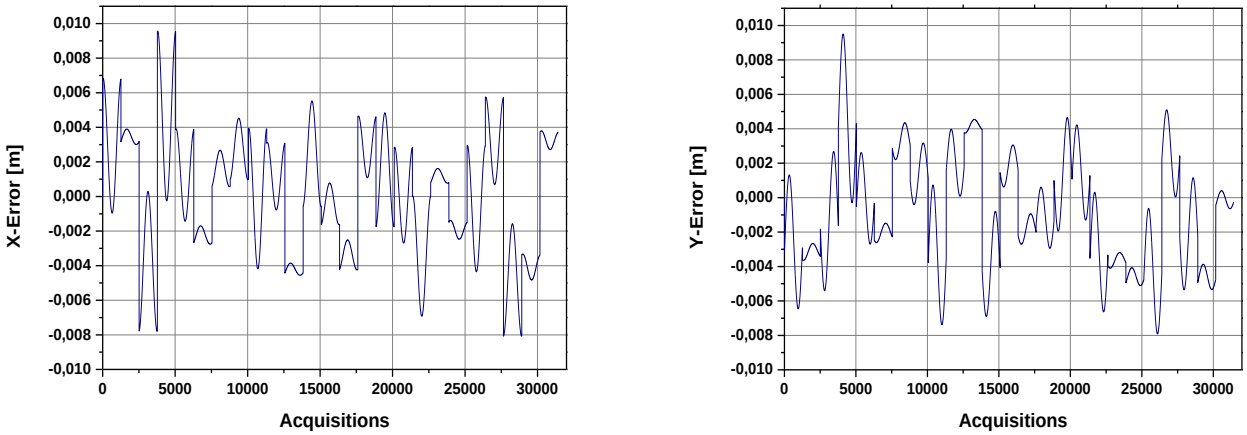
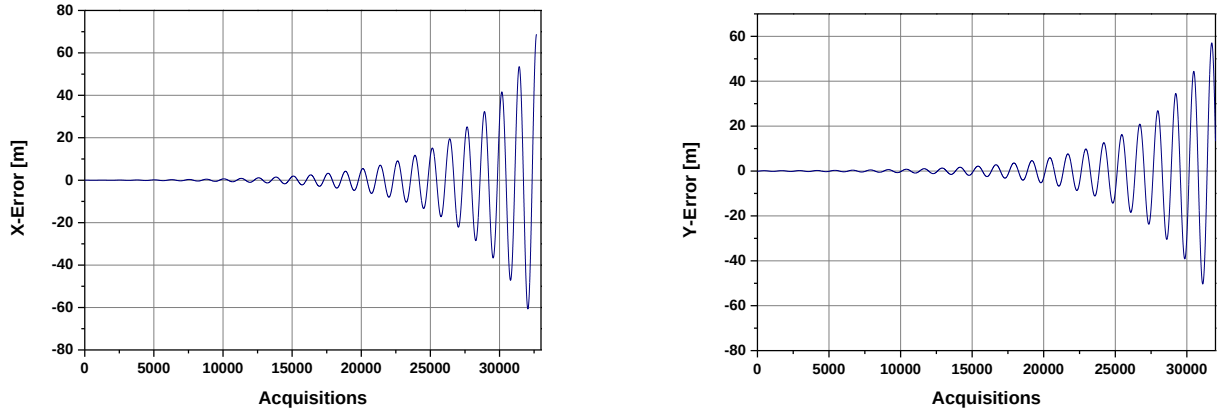


Fig. 15 Trajectory reconstruction without (left) and with (right) Kalman filtering



(a) X axis (b) Y axis

Fig. 16 X/Y axis error fluctuations with Kalman filtering



(a) X axis (b) Y axis

Fig. 17 X/Y axis error fluctuations without Kalman filtering

Table 5 Distribution parameters (with Kalman filter)

Abs Max X-error	0.0066m
Abs Max Y-error	0.0084m
X-MSE (mean square error)	3.30e-3m^2
Y-MSE (mean square error)	6.50e-3m^2

Table 6 Distribution parameters (without Kalman filter)

Abs Max X-error	59.68m
Abs Max Y-error	60.57m
X-MSE (mean square error)	1.53e+5m^2
Y-MSE (mean square error)	1.49e+5m^2

Table 7 Main technical features

Dimensions [cm]	10.5x12.5
Weight [g]	155 (395 with camera)
Maximun transm. range [m]	80 (outdoor)
Inertial frame transm. rate [Hz]	100
GPS channels	32
GPS sensitivity (Track, Nav) [dBm]	-159
GPS frame transm. rate [Hz]	5
Position resolution with EGNOS [m]	< 5 (experimental)
Accmeters dynamic range [g]	±10
Accmeters sensitivity [mg]	2.522
Accmeters axis non-orthogonality [°]	±0.25
Accmeters temp. coefficient [ppm/℃]	100
Gyros dynamic range [°/s]	±300, ±150, ±75
Gyros sensitivity [°/s]	0.07326, 0.03663, 0.01832
Gyros axis non-orthogonality [°]	±0.05
Gyros temp. coefficient [ppm/℃]	600
Nominal input voltage [V]	2.4
Nominal input energy [Wh]	6.36
Max power consumption [mW]	625
Max battery autonomy [hr]	9 (working continuously)

9. Conclusions

A flexible and low-cost wireless GPS/Inertial system which can be used for many kinds of applications is presented. The main features of prototype are low weight, high compactness, high autonomy, fast remote data managing and elaboration (Table 7). The future developments will be the GPS/Inertial data fusion, the replacement of MEMS sensor station with the new model which integrates a tri-axial magnetometer and an automatic thermal compensation, the replacement of the ZigBee module with the new model having a transmission range up to 1km, the using of a single microcontroller device with an integrated I/O (USART) DMA in place of the two microcontrollers and assembling all new modules. In addition, the remote system GUI will be modified to manage data elaboration for various applications such as fall detection, body motion recognition, inertial navigation, etc. Much kind of tests in several scenarios will be conducted in order to demonstrate flexibility and general purpose capability of platform. This paper can be used also as a simple didactic reference which proposes introducing a complete top-down system design flow, touching clearly all steps of a system design: hardware components selection, power supply section designing, microcontroller firmware programming, PCB board designing, GUI software programming, high-level data elaboration and system testing.

References

[1] L. Cheng and S. Hailes, "On-body wireless inertial sensing foot control applications, "Proc. IEEE 19th International Symp. Personal Indoor and Mobile Radio Communications (PIMRC 08), IEEE published, Sept. 2008, pp. 1-5.

[2] M. D. Cooney, C. Becker-Asano, T. Kanda, A. Alissandrakis, and H. Ishiguro, "Full-body gesture recognition using inertial sensors for playful interaction with small humanoid robot, "Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 10), IEEE published, Oct. 2010, pp. 2276-2282.

[3] Q. Li, J. A. Stankovic, M. A. Hanson, A. T. Barth, J. Lach, and G. Zhou, "Accurate fast fall detection using gyroscopes and accelerometer-derived posture information," Proc. 6th International Workshop on Wearable and Implantable Body Sensor Networks (BSN 09), IEEE published, Jun. 2009, pp. 138-143.

[4] J. Skaloud, "Direct georeferencing in aerial photogrammetric mapping, "Photogrammetric Engineering and Remote Sensing, vol. 68, pp. 207-210, 2002.
[5] S. Z. Jamal, "Tightly coupled GPS/INS airborne navigation system, "Aerospace and Electronic Systems Magazine, vol. 27, no. 4, pp. 39-42, April 2012.
[6] H. Che, P. Liu, F. Zhang, and Q. Wang, "A DEEPLY COUPLED GPS/INS INTEGRATED NAVIGATION SYSTEM SUITABLE FOR HIGH DYNAMIC ENVIRONMENTS, "Proc. 3rd China Satellite Navigation Conference (CSNS 12), Springer Berlin Heidelberg, Apr. 2012, pp. 617-626.
[7] S. Godha, G. Lachapelle, and M. E. Cannon, "Integrated GPS/INS system for pedestrial navigation in a signal degraded environment, "Proc. 19th International Technical Meeting Institute Navigation Satellite Division, Fort Worth TX (USA), 2006.
[8] Y. Li, M. Efatmaneshnik, and A. G. Dempster, "Attitude determination by integration of MEMS inertial sensors and GPS for autonomous agriculture applications, "GPS Solutions, vol. 16, no. 1, pp. 41-52, 2012.
[9] T. Oksanen, M. Linja, and A. Visala, "Low Cost Positioning System for Agricultural Vehicles, "Proc. IEEE International Symp. Computational Intelligence Robotics and Automation, Espoo (Finland), 2005, pp. 297-302.
[10] S. M. Warner, T. O. Koch, and T. Pfau, "Inertial sensors for assessment of back movement in horses during locomotion over ground, "Equine Veterinary Journal, vol. 42, pp. 417-424, 2010.
[11] SBG Systems s.a.s., "IG-500N: GPS aided miniature INS," http://www.sbg-systems.com/docs/IG-500N-Leaflet.pdf, 2013
[12] Xsens Technologies B.V., "MTi-G-700 GPS/INS," http://www.xsens.com/images/stories/products/PDF_Brochures/mti%20100-series%20leaflet.pdf, 2013
[13] G. Borgese, L. Rizzo, C. Pace, and G. Artese, "Compact wireless GPS/inertial system," 4th Annual Caneus Fly by Wireless Workshop (FBW 11), IEEE published, Jun. 2011, pp. 1-4.
[14] R. Dorobantu and B. Zebhauser, "Field evaluation of a low-cost strapdown IMU by means GPS," Ortung und Navigation, vol. 1, pp. 51-65, 1999.
[15] O. J. Woodman, "An introduction to inertial navigation, "University of Cambridge: Cambridge, Computer Laboratory, UCAM-CL-TR-696, UK, 2007.
[16] Z. F. Syed, P. Aggarwal, C. Goodall, X. Niu, and N. El-Sheimy, "A new multi-position calibration method for MEMS inertial navigation systems, "Measurements Science and Technology, no. 18, pp. 1897-1907, 2007.
[17] G. Artese, M. Gencarelli, A. Trecroci, G. Borgese, and C. Pace, "Metodologie di calibrazione delle strumentazioni inerziali: il modified multi-position calibration method per la calibrazione dei giroscopi, "Boll. SIFET, no. 1, pp. 171-183, 2010.
[18] Analog Devices Inc., "ADIS16350/ADIS16355 - High precision tri-axis inertial sensor," datasheet rev. B, 2007-2009.
[19] A. Brown, "GPS/INS uses Low-Cost MEMS IMU," IEEE Aerospace and Electronic Systems Magazine, vol. 20, no. 9, pp. 3-10, 2005
[20] D. H. Titterton and J. L. Weston, Strapdown inertial navigation technology, 2nd ed. London: Peter Peregrinus Ltd., 2004.
[21] W. E. Featherstone and S. J. Claessens, "Closed-form transformation between geodetic and ellipsoidal coordinates, "Stud. Geophys. Geod., vol. 52, no. 1, pp. 1-18, 2008.