PC104 Based Low-cost Inertial/GPS Integrated Navigation Platform: Design and Experiments

Di Li, René Jr. Landry and Philippe Lavoie

LACIME, Department of Electrical Engineering
École de Technologie Supérieure (ÉTS), University of Quebec, Montreal, Quebec, Canada, H3C 1K3

Abstract. The integration of Global Positioning System (GPS)/Inertial Navigation System (INS) has become very important in various navigation applications. In the last decade, with the rapid development of Micro Electro Mechanical Sensors (MEMS), great interest has been generated in low cost integrated GPS/INS applications. This paper presents a PC104 based low cost GPS/INS integrated navigation platform. The platform hardware consists of low cost inertial sensors and an assembly of various PC104 compatible peripherals, such as data acquisition card, GPS receiver, Ethernet card, mother board, graphic card, etc. The platform software including inertial/GPS data acquisition, inertial navigation calculation and integrated GPS/INS Kalman filter is implemented with Simulink, which can be directly loaded and processed in the PC104 mother board with the aid of Matlab Real-Time Workshop (RTW) utility. This platform is totally self-embedded and can be applied independently or as part of a system. Simulation and real data experiments have been performed to validate and evaluate the proposed design. A very low cost MEMS inertial sensor was utilized in the experiments. The reference is the navigation solution derived from a tactic grade Inertial Measurement Unit (IMU). Test results show that PC104 navigation platform delivers the integrated navigation solutions comparable to the reference solutions, which were calculated with a conventional laptop computer, however with less power consumptions, less system volume/complexity and much lower over-all costs. Moreover the platform hardware is compatible to various inertial sensors of different grades by configuring the related parameters in the system software.

Keywords: PC104, GPS, MEMS, Kalman filter, Real-time Workshop, xPC Target

1 Introduction

As an independent means of navigation, GPS is capable of delivering position and velocity information with time-independent precision, while the performance becomes unreliable however when the system is exposed to high dynamics, interference from communication equipments and intentional/non-intentional jamming, etc. Compared with GPS, INS providing position, velocity, and attitude information via the measurements from inertial sensors has various advantages, such as totally autonomous, high dynamic response, good short-term accuracy and robust performance when exposed to interference and or jamming. However its usage as a stand-alone navigation system is limited due to time-dependent growth of the inertial sensor bias/noise. Because of the aforementioned complementary characteristics, GPS and INS are commonly coupled by Kalman filter to augment the over-all performance by overcoming the shortcomings of each individual system.

A high precision integrated GPS/INS system requires expensive inertial sensors that have exceptional long term bias stability. The sensor cost limits such kind of integrated navigation systems to very expensive applications (Hayward et al., 1997). Over the past decade low cost MEMS are experiencing rapid improvements in terms of precision, robustness, size, high dynamic response and so on. With the quick growth in demand for low cost navigation systems for general aviation, unmanned automotive vehicles, locating personnel, mobile mapping systems, athletic training and monitoring, and computer games, etc, it has become important to develop low cost integrated navigation systems.

This paper introduces the development of a low cost inertial/GPS integrated navigation platform. The platform hardware is constructed on the basis of a PC104 computer and an assembly of PC104 peripherals, such as data acquisition card, graphic card, Ethernet card, power

supply board and a PC104 compatible GPS receiver, etc. Matlab Simulink's modularity and graphical design make it convenient for point-wise improvements and facilitate the ramp-up knowledge of future contributors (Giroux, 2005). Also one of the Simulink's powerful assets is the possibility to do rapid real-time testing through the RTW and xPC Target. Therefore the system software comprising the data acquisition, strapdown inertial mechanization and integrated Kalman filter is implemented by the Matlab Simulink. This Simulink based platform software can be directly compiled into executable code for PC104 computer by the RTW. The Simulink based design scenario has various advantages, such as rapid real-time prototyping and fast re-designing/debugging which is particularly helpful in the early stage of developing a real-time navigation system. The architecture of PC104 navigation platform is depicted in Fig. 1.
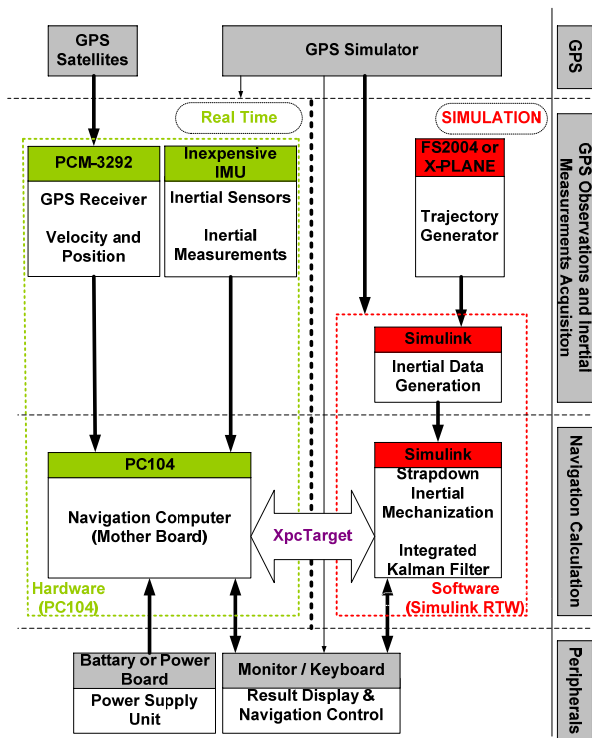

Fig. 1 Architecture of PC104 navigation platform

This paper is organized as follows. Section 2 presents an overview of the platform, the devices utilized, the software used to communicate and the algorithm structure. Section 3 describes the low cost MEMS real time test results followed by a performance evaluation. Finally, future work and potential enhancement of the project are discussed in the conclusions.

2 Hardware platform and software design

2.1 Hardware platform

The hardware platform consists of a PC104 computer and an assembly of PC104 peripherals, such as data acquisition card, graphic card, Ethernet card, power supply board and a PC104 compatible GPS receiver. All the component parts are stacked up together through the PC104 bus. This configuration makes the individual peripheral independent from each others as well as provides a good synchronization for the data transmission. For example, the platform utilizing MEMS inertial sensor is depicted in Fig. 2.
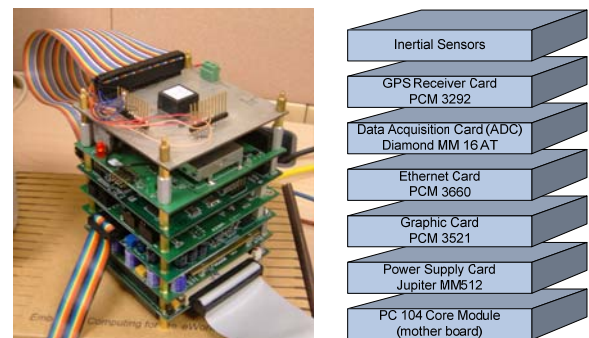

Fig. 2. PC104 navigation platform prototype

On the top layer there is a MEMS inertial sensor which generally is not PC104 compatible but the information is sent via an external bus to the data acquisition card. This layer provides the specific force and angular rate measurements given by the MEMS. On the next layer, there is a PC104 compatible GPS receiver which obtains and transmits the position and velocity observations to the algorithm. Following the GPS receiver, there is a 16-bit data acquisition card which collects and converts the MEMS inertial analog signal to digital data. If the digital inertial data are available in MEMS sensor, e.g. a USB interface is provided by the newly acquired MEMS sensor in the lab (nIMU MEMSense™) or a RS232 interface used in the tactic IMU, this card can be excluded. The digital data should be directly connected to the PC104 computer USB/RS232 interface, under which a PC104 compatible power board provides the power supply to each card. Then, a PC104 compatible network card is added to build a TCP/IP connection to a host computer. This connection is used for carrying out several important tasks: first, to download the programs to the mother board, second to setup the control parameters and record the calculated navigation results, then it enables the platform in the remote-control mode through Internet. This platform is also equipped with a graphic card to display all the data/parameters/results residing in the platform in real time on an additional screen. On the bottom layer, there is the PC104 motherboard where the algorithm is loaded and executed.

2.2 Platform software

The proposed system software package consists of strapdown inertial mechanization, the integrated Kalman filter and the software interface. The scheme of the system software is shown in Fig. 3.
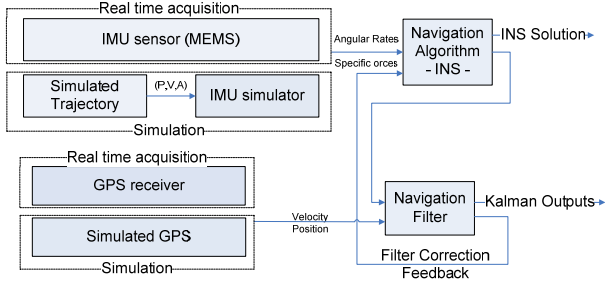

Fig. 3. Scheme of the system software

Strapdown inertial mechanization

There are many approaches to implementing the strapdown mechanization, which are generally divided into two categories, i.e. a multi-speed digital design including accurate coning, sculling, scrolling compensations for attitude/velocity calculation and a simplified single speed continuous design without any attitude, velocity or position compensation algorithm (Savage, 2000). Both algorithmic designs have been

implemented and investigated by Li et al. (2007). By utilizing Matlab Simulink's capabilities to directly evaluate the differential equations in the continuous mode, a simpler single high speed inertial calculation algorithm structure based on the INS analytically continuous differential equations is implemented in this study. The attitude, velocity and position solutions are derived by evaluating the rate equations as follows:

$$\dot{C}_B^L = C_B^L(\omega_{IB}^B\times) - (\omega_{IL}^L\times)C_B^L \qquad \omega_{IL}^L = C_N^L(\omega_{IE}^N + \omega_{EN}^N)$$
$$\dot{v}^N = a_{SF}^N + g_P^N - (\omega_{EN}^N + 2\omega_{IE}^N)\times v^N \qquad g_P = g - \omega_{IE}\times(\omega_{IE}\times R) \qquad (1)$$
$$\dot{C}_N^E = C_N^E(\omega_{EN}^N\times) \qquad \dot{h} = v_\perp^N$$

where C_B^L is the attitude matrix; $\omega_{IB}^B\times$ is the skew-symmetric matrix of the angular rate vector in the body frame (B-frame), $\omega_{IL}^L\times$ is the skew-symmetric matrix of the angular rate vector caused by the translational motion in the L frame, ω_{EN}^N is the angular rate of N Frame relative to E Frame, v^N is the velocity vector, a_{SF}^N is the specific force vector, g_P is the plumb-bob gravity, g is the standard gravity, R is the position location vector from the earth centre. For example the Matlab implementation of the attitude solution is depicted in Fig. 4.
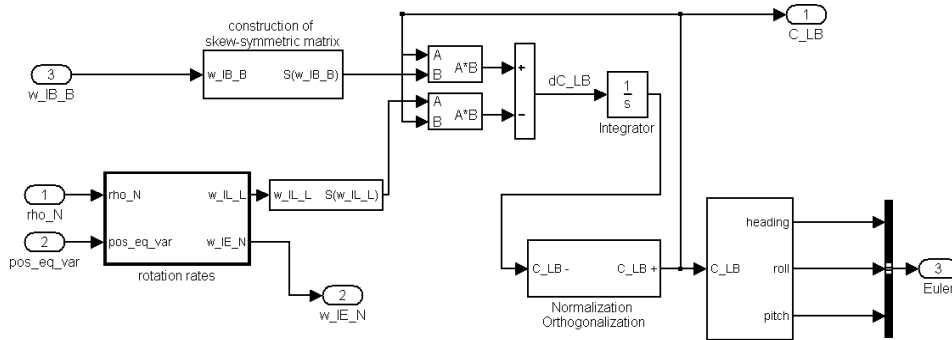

Fig. 4. Attitude implementation in Simulink

Integrated Kalman filter

According to the aforementioned advantages, the Kalman filtering is applied to combine the inertial/GPS data in this study. The roles of the Kalman filter in our application are to estimate/correct the errors in navigation parameters, e.g. position, velocity and attitude, and also to estimate the inertial sensor bias/drift which enables the in-motion calibration of inertial sensor raw measurements.

The measurements in the proposed Kalman filter are formed from the comparison between the INS calculated and GPS receiver derived position/velocity data. Such measurements are derived using the estimated attitude, velocity, position and MEMS sensor error states. The

design of the Kalman filter dynamic model is based on the INS error model. In our system, the so-called psi-angle error model is applied, which defines the errors in attitude, velocity and position parameters ($\Psi, \delta V, \delta R$) (Savage, 2000):

$$\dot{\Psi}^N = -C_B^N\delta\omega_{IB}^B - \omega_{IN}^N\times\Psi^N$$
$$\dot{\delta V}^N = C_B^N\delta a_{SF}^B + a_{SF}^N\times\Psi^N + \delta g_{Mdl}^N - (2\omega_{IE}^N + \omega_{EN}^N)\times\delta V^N \qquad (2)$$
$$\dot{\delta R}^N = \delta V^N - \omega_{EN}^N\times\delta R^N$$

where, $\Psi, \delta V, \delta R$ are the errors in attitude, velocity and position parameters, C_B^N is the attitude matrix expressed in the navigation frame (N-frame), $\delta\omega_{IB}^B$ is the angular-rate error vector in the B-frame, $a_{SF}^N, \delta a_{SF}^B$ are the specific force vector in the N-frame and the specific force error

vector in the B-frame, δg_{Mdl}^{N} is the plump-bob gravity error, $\omega_{IE}^{N}, \omega_{EN}^{N}$ are the earth rotation rate vector and the transport rate vector in the N-frame, respectively; ω_{IN}^{N} the N-frame rotation rate in the inertial frame (I-frame). The error parameters in attitude, velocity and position are represented as the error states which are propagated in the Kalman filter through the dynamic model. As in the application of low cost inertial sensors, the sensor noises are the dominant terms causing attitude, velocity and position errors, i.e. many error terms in the INS error model are negligible when compared with raw measurements errors δa_{SF}^{B} and $\delta \omega_{IB}^{B}$. Therefore those terms can be removed from the error model which in turn reduces the number of the Kalman filter error

states, remarkably decreasing the computing load (Li et al., 2007). The simplified error equation is given as:

$$
\begin{aligned}
\dot{\Psi}^{N} &= -C_{B}^{N} \delta \omega_{IB}^{B} \\
\delta \dot{V}^{N} &= C_{B}^{N} \delta a_{SF}^{B} + a_{SF}^{N} \times \Psi^{N} \\
\delta \dot{R}^{N} &= \delta V^{N}
\end{aligned}
\tag{3}
$$

This continuous Kalman dynamic model should be discretized to build Kalman state transition matrix (Phi-Matrix) and the integrated dynamic noise matrix (Q matrix). The discretizing rate is chosen as the available maximum sensor data rate, which is sensor dependent, typically varying from 100Hz to 200Hz. The Matlab implementation of the Kalman Dynamic model is depicted in Fig. 5.
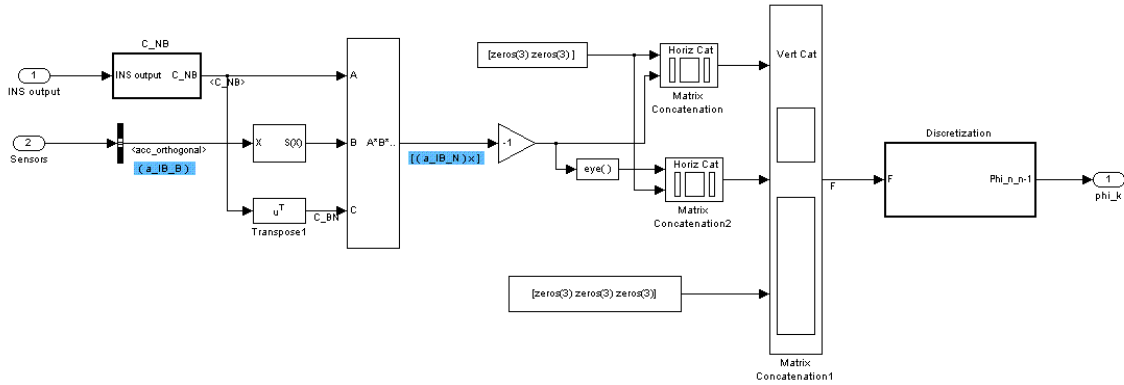

Fig. 5. Simulink implementation of Kalman state transition matrix

Software interface

For real time processing, the Simulink implemented navigation algorithm can be re-written in C, complied as executable codes and then downloaded to the hardware platform by the RTW. Simulink's xPC Target Library provides the necessary drivers for the different peripherals. The connection to the monitor, keyboard and the host computer to display, read and command the platform in the real time is created by the drivers from xPC Target Library. For example, the data acquisition of the raw inertial measurement implemented in Simulink by xPC Target is depicted in Fig. 7 . In this study, the Diamond MM 16 AT card is utilized by the platform. The configuration of the Diamond MM16AT Acquisition Driver Module is shown in Fig. 6.
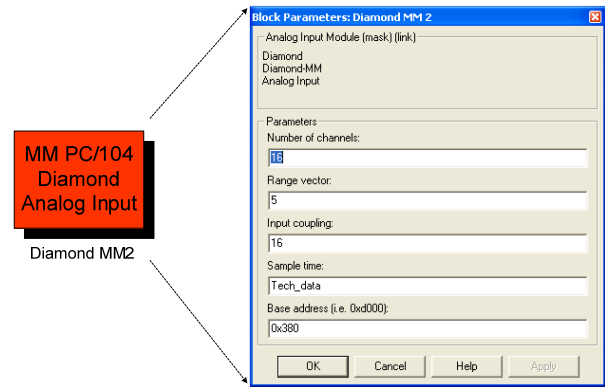
The settings for this module are Number of Channels: Number of different outputs; Range Vector: Maximum Amplitude of the Output; Input Coupling: Number of input for the ADC; Sample Time: Output Sampling Time; Base Address: Output Address. In this platform, the data are encoded with 16 bits precision for the 5 volts range. Therefore the resolution is 7.63×10^{-5} V/bit which means a resolution of 76.3 µg for the accelerometers as the sensitivity is 1000 mV/g and a resolution of 6.1×10^{-3} °/sec for the gyroscopes as the sensitivity is 12.5 mV/°/sec.

Fig. 6. Configuration of data acquisition driver

3 Experiments

Simulation and real data experiments were performed to validate and evaluate the proposed design. The simulation test was to validate the designed navigation algorithm. Following the simulation, the real data experiment tests the platform hardware and software. A low cost MEMS and a low cost tactic IMU were utilized in the real data experiments.

3.1 Simulation test

Fig. 8 depicts the trajectory processed in the simulation. The red curve is the reference trajectory generated by the Flight Simulator (Microsoft FS2004) and the blue one corresponds to the solution calculated by the algorithm.
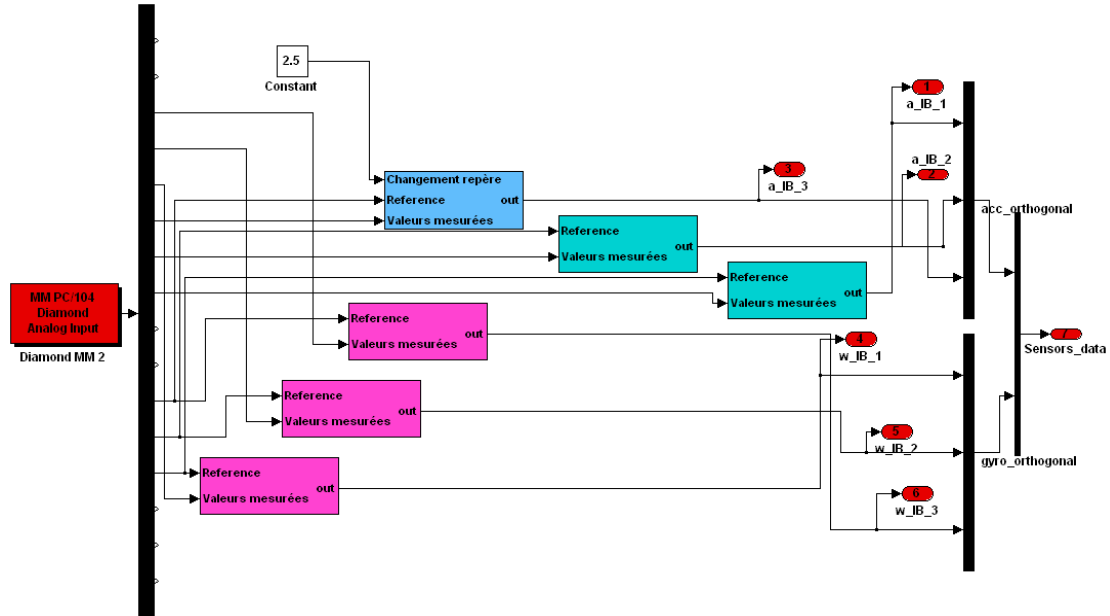

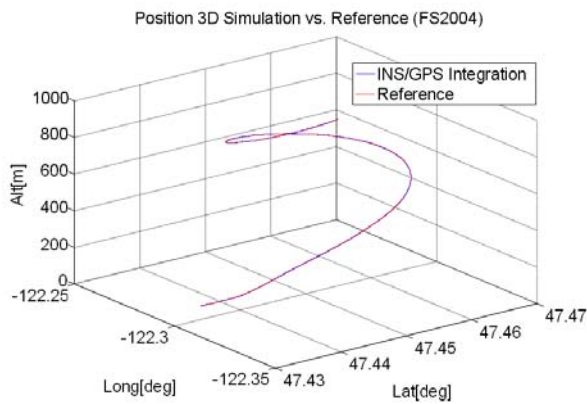Fig. 7. Simulink implementation of data acquisition software

Fig. 8. 3D trajectory simulation vs. reference

Position: The Kalman filter provides the updated corrections of the position errors every 10ms. The maximum error located on an axis is about 5 meters. This period corresponds to the high dynamic period during the flight. It can be seen from the Fig. 9 that the position errors remain small and stable during the simulation.

P matrix: i.e. the state covariance matrix gives the information about Kalman filter estimation performance. Its convergence means that the estimates are close to the reality. The P matrix of this experiment shows that all the parameter estimates converge quickly as shown in Fig. 10.

Velocity: Compared with the reference solutions, the velocity errors depicted in Fig. 11 never exceeds 1% of the velocity reference profile, indicating that the estimated velocity values are accurate.
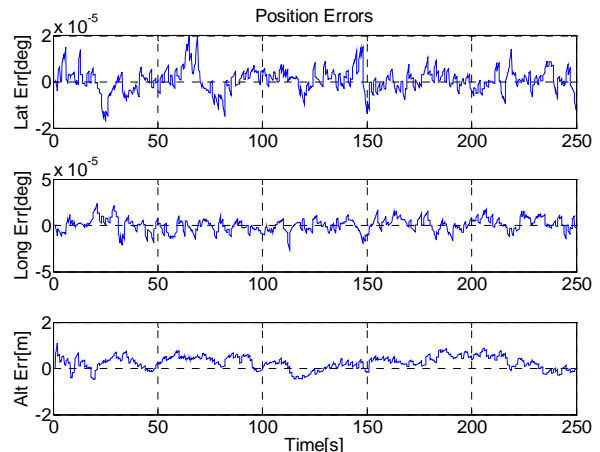

Fig. 9. Position errors

Similar to the position and velocity results, the attitude errors shown in Fig. 12 are small and stable. From the above results it can be concluded that the simulation test validates the proposed navigation algorithms.
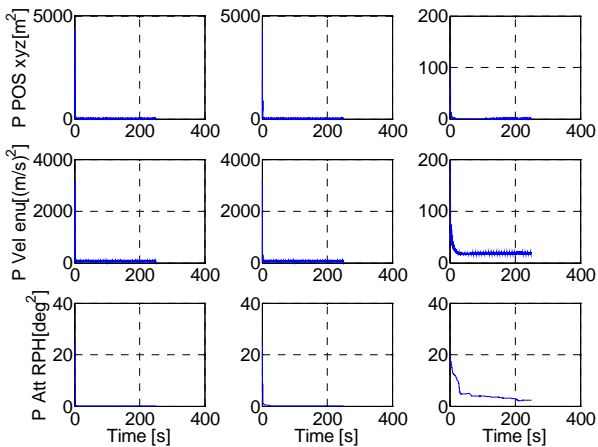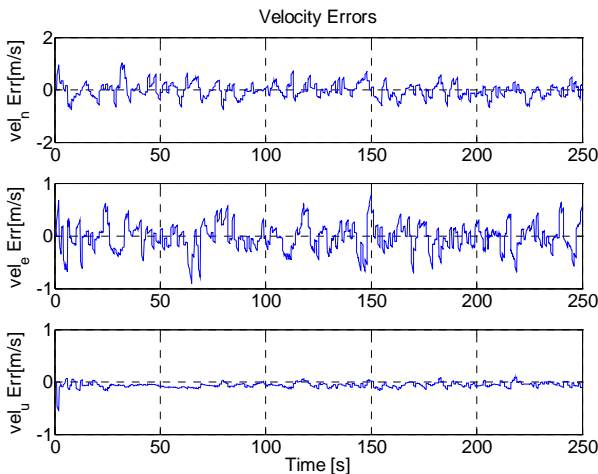
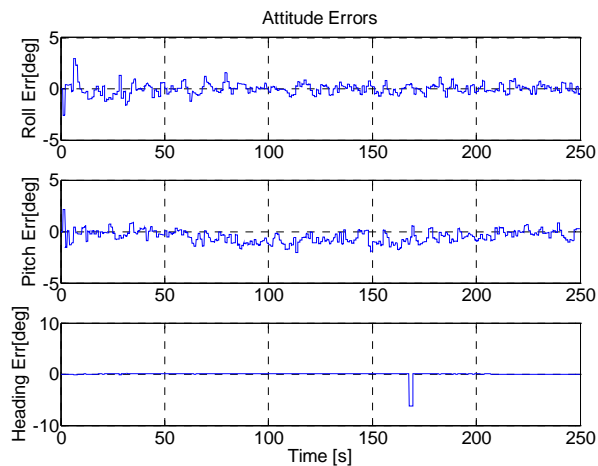Fig. 10. Convergence of Kalman estimation

Fig. 11. Velocity errors

Fig. 12. Attitude errors

3.2 Real data test

The real data experiment was performed to test the proposed navigation platform using a very low cost MEMS inertial sensor (MEMSense™'s AccelRate3D).

Only analog inertial measurements are available from AccelRate3D MEMS sensor, which were collected and converted to digital data in PC104 data acquisition card (Diamond MM 16 AT). The compiled navigation software was downloaded to the platform from host computer through the PC104 Ethernet card. The control commands and navigation software parameters were setup by an independent keyboard. Moreover the control commands and the real-time navigation solution may also be displayed by the monitor with the aid of PC104 graphic card. The navigation solutions (NovAtel SPAN ™ Best PVA) derived from a tactic grade IMU (Honeywell HG1700 IMU™) were employed as the reference. The inertial devices are shown in Fig. 13 and Fig. 14. IMU specifications are provided in Table 1 and Table 2 respectively.
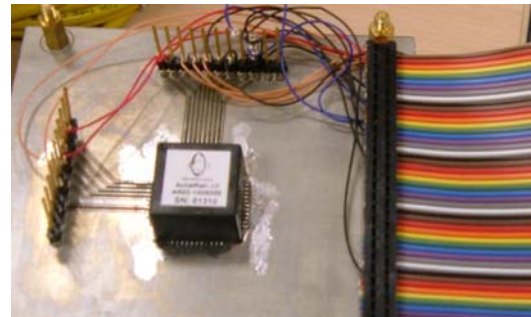

Fig. 13. AccelRate3D MEMS inertial sensor

Fig. 14. Reference SPAN IMU

Table 1 MEMS sensor specs (MEMSense AccelRate3D)

AccelRate3D	Dynamic Range	Noise
Accelerometer	±2 (g)	35(μg/√Hz)
Angular Rate Sensor	±300 (º/s)	0.1(º/h/√Hz)

Table 2 Reference IMU (NovAtel SPAN)

SPAN IMU	Dynamic Range	Noise(Random Walk)
Accelerometer	±50(g)	34(μg/√Hz)
Angular Rate Sensor	± 1000 (º/s)	0.125(º/√hr)

First of all, a stand alone test was made with this very low-cost MEMS sensor. The purpose of this test was to test its performance without any aiding from GPS. The trajectory was made inside THE ETS building. Since the positions are know precisely in the testing corridor, the reference can be achieved accurately as depicted in Fig. 15.
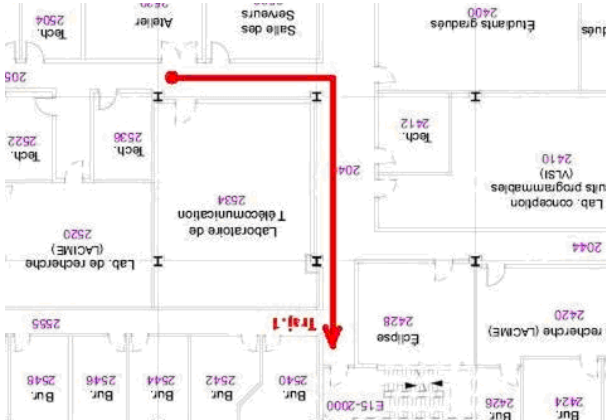

Fig. 15. Corridor trajectory

The data were processed by the navigation platform, and the results are shown in Fig. 16. The detailed results are shown in Table 3.

Due to the MEMS sensor's high noise characteristics, the test of 20 seconds is fairly long for MEMS standalone application. Hence there are significant time-dependent drifts in the navigation solutions as shown in Fig. 16.
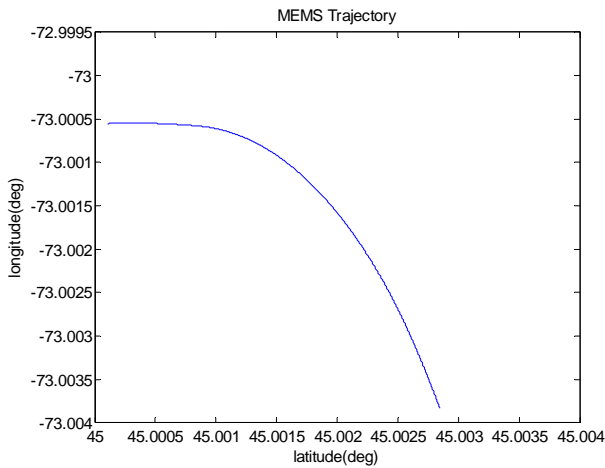

Fig. 16. MEMS standalone solution

Table 3 Trajectory vs. MEMS results

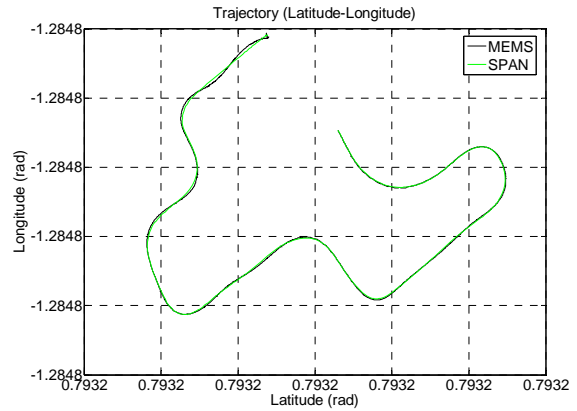|	Reference	MEMS
Total Distance	15.97m	8.30m
Latitude Dist	8.23m	5.32m
Longitude Dist.	13.67m	6.36m
Test Duration	20.36s	

Fig. 17. Trajectory MEMS vs. SPAN

Second, the integrated navigation test was performed on the platform. The trajectory was made outside the building with the good acquisition of GPS signals. The GPS and MEMS inertial data were acquired and processed by the PC104 platform and the results were logged. The reference solutions were calculated in the laptop computer. The trajectory, i.e. the integrated MEMS/GPS solution vs. the reference, is depicted in Fig. 17. The duration of the test was 107s. It can be seen that the MEMS/GPS integration solutions start to diverge from the reference at the end of the test due to the MEMS IMU error growing much faster than that of the high quality tactic grade SPAN IMU sensor.

Compared with the reference solutions, the maximum position solution errors from the platform shown in Fig. 18 are about 5×10^{-7} radian (1-2 meters in Cartesian coordinates).
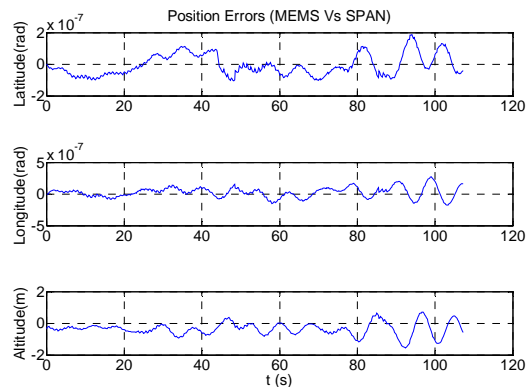

Fig. 18. Position solution errors

The velocity solution errors are depicted in Fig. 19. The errors remained small during the test, while the vertical velocity started to diverge at the end. One of the reasons causing this divergence is the time dependent biases in MEMS accelerometer raw measurements. It can be seen that the raw specific force measurements of the MEMS IMU are much noisier than those of the tactic grade IMU as shown in Fig. 20.
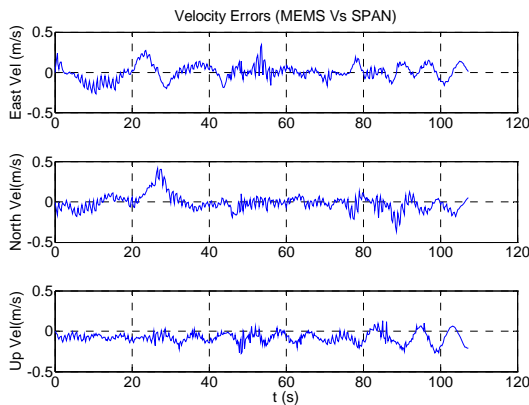
Fig. 19. Velocity solution errors

Compared with the reference solutions, the attitude errors are 5.7º, 2.9º and 11 º respectively in roll, pitch and heading angle, as shown in Fig. 21. These errors are caused by the high noise contaminating the MEMS angular rate measurements, and more importantly there is no direct attitude observation available from GPS. The heading error grows over the time, which is the essential reason causing the slight divergences in velocity and position solutions.
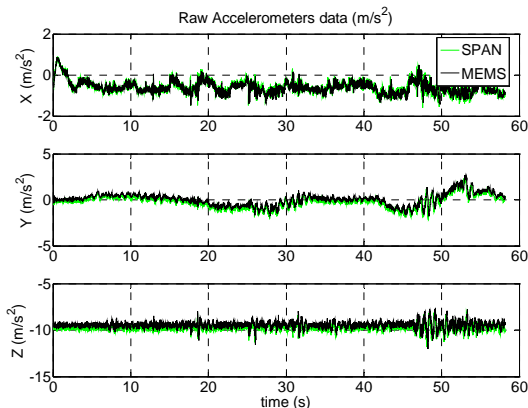

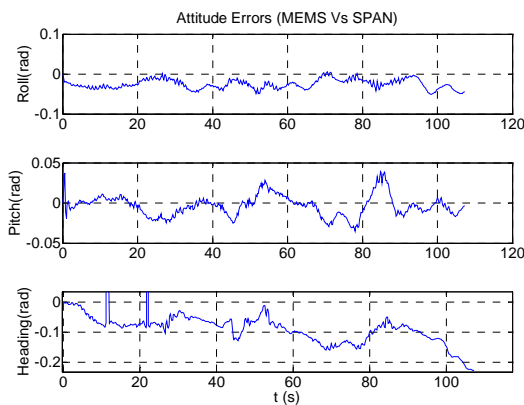Fig. 20. Raw specific force measurements MEMS vs. SPAN

Fig. 21 Attitude solution errors

Similarly the raw angular rate measurements from the MEMS IMU and the reference IMU are depicted in Fig. 22. Although measuring the same angular dynamics, the raw angular rate measurements of the MEMS IMU are much noisier than those of the reference IMU. Hence it can be concluded the MEMS navigation solutions are greatly improved by the integration of the IMU with GPS.
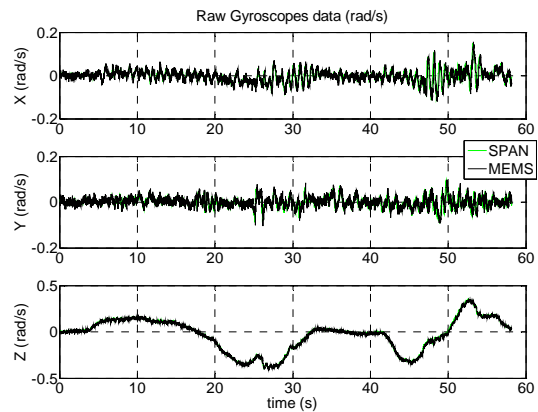

Fig. 22. Raw angular rate measurements MEMS vs. SPAN

4 Conclusions

This paper presents the design & experiment for a low cost inertial/GPS integrated navigation platform based on PC104 computers. The navigation system software has been specifically designed in Simulink for low cost inertial sensor applications. The simulation experiment validates the proposed hardware and software designs.

The real-time/data test has demonstrated that the PC104 navigation platform can deliver the integrated navigation solutions comparable to the reference solutions, which are calculated in the conventional desktop computer system, whereas with less power consumptions, less system volume/complexity and much lower over-all costs. Furthermore the platform hardware is compatible with various inertial sensors of different grades.

Although there are various advantages in the Simulink based software design proposed by this study, many PC104 hardware devices are currently not supported by the Simulink xPC Target. Moreover the C-programming based digital mode navigation software comprising attitude coning and velocity sculling compensation algorithms is more appropriate for a high performance navigation system. Hence fully functional C-programming based platform software is currently under development. By applying a newly acquired USB-based digital MEMS nIMU (MEMSense™) instead of the analog AccelRate 3D, a complete digital hardware/software PC104 navigation platform design

with a C programming high performance navigation software will be implemented.

References

Hayward, R., Gebre-Egziabher, D., Schwall, M., and Powel, J.D. (1997) *Inertially Aided GPS Based Attitude Heading Reference System (AHRS) for General Aviation Aircraft*, Proceeding of Institute of Navigation – ION-GPS Conference,Page 1415-1424.

Savage, P.G. (2000) *Strapdown Analytics Part I*, Strapdown Association, Maple Plain, Minnesota.

Savage, P.G. (2000b) *Strapdown Analytics Part II*, Strapdown Association, Maple Plain, Minnesota.

Li, D. and Landry, R. (2007) *MEMS IMU Based INS/GNSS Integration: Design Strategies and System Performance Evaluation*, the Navigation Conference & Exhibition – NAV07, London, UK

Li, D., Landry, R., and Lavoie, P. (2007) *Validation and Performance Evaluation of Two Different Inertial Navigation System Design Approaches*, International Global Navigation Satellite Systems Society Symposium 2007, Sydney, Australia

MEMSense AccelRate3D. www.memsense.com.

Giroux R., Landry R.Jr., Leach B. and Gourdeau R. (2003), *Validation and Performance Evaluation of a Simulink Inertial Navigation System Simulator*, Journal of Canadian Aeronautics and Space, Vol. 49, No. 4, p 149-161.

Giroux, R., Gourdeau R. and Landry R.Jr. (2005) *Extended Kalman filter real-time implementation for low-cost INS/GPS Integration in a Fast-prototyping Environment*, 16th Canadian Navigation Symposium, CASI , Toronto, Canada, 26 - 27 April.

Titterton, D. and Weston, J. (2004) *Strapdown Inertial Navigation Technology: Second Edition*. AIAA.

Flight   Simulator   2004 http://www.microsoft.com/games/flightsimulatorx/