

A low-cost real-time GPS/INS integrated system with hardware and software field re-configurability.

**Author:**

LI, Y; MUMFORD, P; RIZOS, C

**Publication details:**

Coordinates

v. IV

Chapter No. 3

pp. 12-17

**Publication Date:**

2008

**License:**

<https://creativecommons.org/licenses/by-nc-nd/3.0/au/>

Link to license to see what you are allowed to do with this resource.

Downloaded from <http://hdl.handle.net/1959.4/44375> in <https://unsworks.unsw.edu.au> on 2024-03-29

# A low-cost real-time GPS/INS integrated system with hardware and software field re-configurability

**Yong Li, Peter Mumford, and Chris Rizos**

University of New South Wales, Australia

Tel: 61 2 9385 4173, Fax: 61 2 9313 7493, Email: [yong.li@unsw.edu.au](mailto:yong.li@unsw.edu.au)

## ABSTRACT

A real-time GPS/INS integrated system based on a field programmable gate array (FPGA) platform has recently been developed at the Satellite Navigation and Positioning Laboratory, School of Surveying & Spatial Information Systems, University of New South Wales. Compared with an application-specific integrated circuit (ASIC) approach, the FPGA approach can shorten the research and development cycle. Its reprogrammable hardware provides a system design methodology of lower risk and maximum flexibility, being able to integrate a wide range of GPS and INS sensor packages.

Tests demonstrate the feasibility of this type of system on an FPGA platform. Functionality of the system has been tested including: the stability and accuracy of the time synchronisation mechanism, the performance of the hardware and software architecture, the workability and accuracy of the algorithm, and finally, the capability of the integrated system to bridge GPS outages.

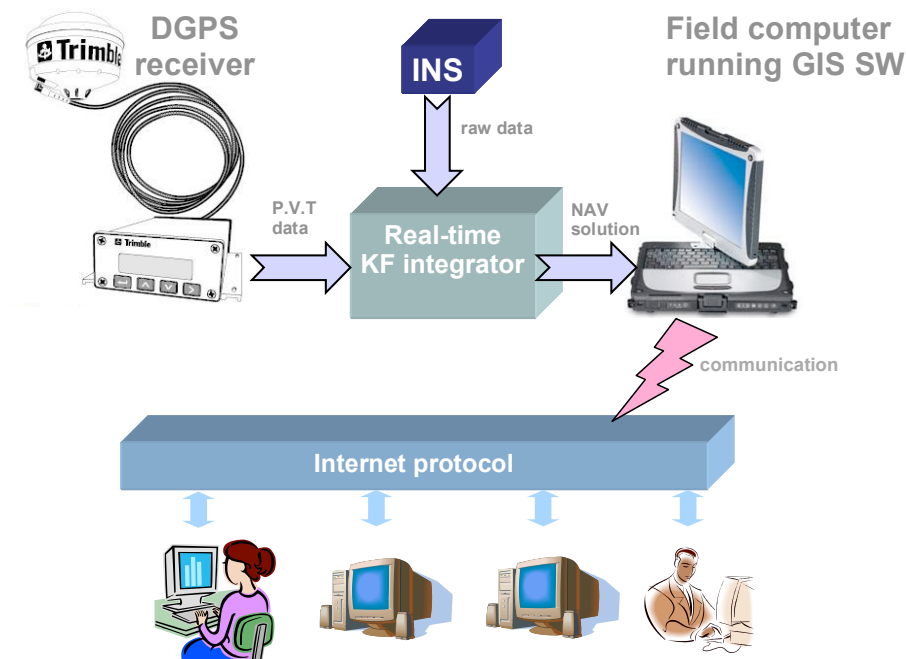
## 1. INTRODUCTION

Although Global Navigation Satellite Systems (GNSS) technology is developing rapidly, the major disadvantage of GNSS will still exist even when the European Galileo system is fully operational, that is, signal blockage due to obstructions and the low power of the signals. The combination of GNSS with a self-contained inertial navigation system (INS) provides an ideal solution, which can not only address the weakness of GNSS and but also bound the INS error that grows with the time when operating on its own. The integrated system can provide a continuous position, velocity, and attitude solution at a high output rate even during a GNSS outage, albeit for a limited period. These advantages drive GNSS/INS integration in military and civil applications (Buck *etal* 2006, Kennedy *etal* 2006).

A multisensor integration platform based on a field programmable gate array (FPGA) has been developed at the Satellite Navigation and Positioning Laboratory (SNAPlab), School of Surveying & Spatial Information Systems, University of New South Wales. The biggest advantage of the FPGA-based system is that all the hardware and software components of the system are field re-programmable without any hardware changes, with even the processor of the system itself being “soft”. A “hardcopy” FPGA can be made after the system has been sufficiently tested.

An FPGA is an integrated circuit capable of implementing digital circuits by means of a configuration process. The designer can use it to implement the specified logic (Hidalgo 2003, Meyer-Baese 2001). Two major hardware description languages (HDLs) are popularly used for the FPGA design today, namely Verilog and VHDL, both of which are IEEE standards. The authors have used the Altera HDL “AHDL” for most of the design.

Figure 1 illustrates the system architecture of a typical setup of a real-time GPS/INS integrated system. As shown in Figure 1, the GPS and INS data are fed into the FPGA system where the real-time Kalman filter estimates the INS errors that are then used to correct the INS solution. The corrected solution is sent to a field computer on which a geographic information system (GIS) runs. The INS solution can then be plotted onto a map or visualised on the GIS platform. The solution and data collected in the field is sent to a command or monitoring centre via a wireless communication link, i.e. wireless internet, or the mobile phone network. High-level commands or decisions can be made by the centre on the basis of the real-time information that is received. In comparison with post-processing systems, real-time systems can respond to urgent events promptly, with minimum delay. This is vital in, for example, emergency service applications.



**Figure 1.** Illustration of a real-time GPS/INS integrated system

## 2. SYSTEM DESIGN AND IMPLEMENTATION

### 2.1 Hardware

The real-time system is built around the Nios II soft-core on a Stratix EP1s10 device. The GPS pulse-per-second (PPS) signal is required for the time synchronisation process and is connected to the prototype device via a BNC socket. The device is currently configured with four UARTs, two of them for INS and GPS input, and another two for integration result output. The device has an LCD screen for menu and status information display and four buttons for option selection and operation control.

Custom designed logic has been developed for the FPGA to provide count stamping on the incoming serial data streams. The processor logic residing in the FPGA chip hosts the application software that interfaces with the user and controls the custom logic and Compact Flash card operations (Altera 2005). The hardware design file and the firmware are downloaded to the flash memory (AMD AM29LV065D). When power is applied to the board, a configuration controller device attempts to configure the FPGA with hardware configuration data stored in flash memory. Figure 2 shows the hardware box of the system.



**Figure 2.** Hardware of the real-time FPGA-based GPS/INS system

The INS used is Boeing's C-MIGITS II, a so-called tactical grade inertial measurement unit (IMU) that provides raw inertial data (Boeing 1997). An OmniStar-HP8200 GPS receiver is used to provide the PPS signal and the GPS navigation solution. The main specifications of the real-time system are listed in Table 1.

**Table 1.** Characteristics of the real-time system

<b>FPGA</b>	Altera's Stratix EP1S10
<b>Processor</b>	Nios II
<b>Oscillator</b>	50MHz
<b>SRAM</b>	512KB x 2
<b>SDRAM</b>	16MB
<b>Flash</b>	8MB
<b>Embedded OS</b>	eCos for Nios II, ver5.1.
<b>Interface</b>	one LCD; four Buttons; one CF card slot; four UARTs; one BNC.
<b>GPS</b>	OmniStar-HP8200
<b>INS</b>	C-MIGITS II

## 2.2 Time-synchronisation UART

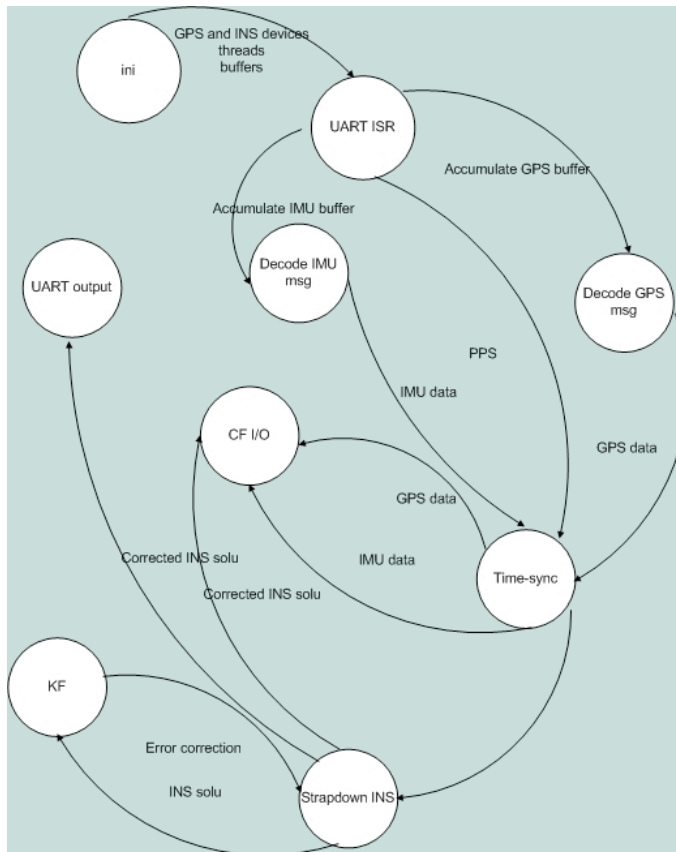
A specified UART has been designed for the time synchronisation of the GPS and INS data. This "time-sync UART" logic is attached to the processor as a memory-mapped peripheral with one interrupt line (Mumford *etal* 2006).



The UART must detect transmission, receive the data in serial format, strip off the start- and stop-bits, and store the data word in a parallel format, as well as access a free-running counter that is latched at the start bit of a serial transmission. This count is appended to the incoming byte and placed in a first-in-first-out (FIFO) buffer. The PPS signal along with GPS time data are used in an interpolation algorithm to calculate the time-of-arrival of serial data from the INS. As a result, the INS data is time-tagged with GPS time, and therefore the INS data is available for comparison with the GPS data in the GPS time frame.

### 2.3 Software

The embedded software is developed using a special version of the Embedded Configurable Operating System (eCos) – ‘the eCos for Nios II’, which provides support of the FAT32 file I/O for the Compact Flash (CF) card, multi-task programming, LCD display, and interrupts from UARTs and buttons (Massa 2002, Nios Community Forum 2005).

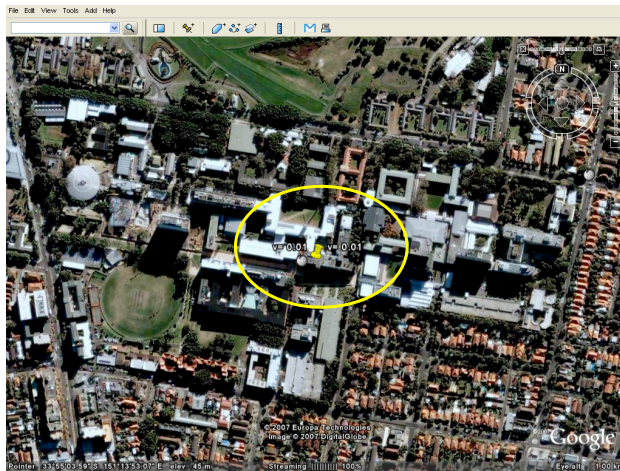


**Figure 3.** Operation diagram of the embedded software

The operations of the program are depicted in Figure 3. The software consists of four threads, which are the user interface (UI), time synchronisation (TS), strapdown INS (SDINS), and Kalman filtering (KF). The system first allocates the memory for the circular buffers, and

commands the GPS and INS devices to output the requested messages. The ISR of UART event notifies the UI or TS thread to drain the data from the UART FIFO buffer and performs the decoding procedure to convert the binary stream to the data messages. The TS procedure aligns the IMU data with the GPS time frame so that comparison of the IMU and GPS data is possible. The strapdown navigation solution is calculated from the time-synced IMU data. Using the GPS data, the Kalman filter further estimates the errors in the inertial solution (due to the inertial sensors). The error estimates are used to correct the inertial solution and improve the result. The corrected INS solution is sent to the external world via a UART port. Meanwhile the system stores the time-synced IMU data and GPS data together with the corrected inertial solution onto the CF card for replay or post-processing.

The corrected navigation solution is sent out in a pre-defined format to an external device, e.g. a handheld computer, where a program runs to receive the solution and send it to GoogleEarth via the internet. The ".kml" file describes the server address on the internet to which the client sends the data. By linking to the address, the position of the host platform can be monitored on any PC in the command/monitor center using the GoogleEarth viewer. Figure 4 depicts a screen copy of a test at the top of the building in which the SNAPlab is located. Using the real-time GPS/INS solution, GoogleEarth automatically zooms in to the area around the SNAPlab building.



**Figure 4.** Real-time GPS/INS solution displayed on the GoogleEarth viewer

## 2.4 Algorithm

The strapdown inertial computation has been performed in the navigation coordinate system (n-frame). The psi-angle model is used in the 15-state GPS/INS integration Kalman filter as the INS error model (Bar-Itzhack and Berman 1988).

Three operation modes have been implemented in the embedded algorithm: (1) the coarse alignment; (2) the fine alignment; and (3) the strapdown INS and integration Kalman filtering. During the coarse alignment, the platform remains static while the tilt angles are computed from the accelerometer data. In addition, the sensor noise levels can be estimated during the coarse alignment (Kennedy 2006). The heading angle can be roughly estimated from gyro-

compassing the gyroscope data. However the C-MIGITS II has a gyro bias of 30deg/hr (Boeing 1997), a magnitude almost twice the earth rotation rate and hence the heading result derived from the C-MIGITS II is not very meaningful. A heading correction can be obtained from the GPS velocity when the platform is moving. During the fine alignment, the Kalman filter estimates the tilt errors and the sensor biases. Due to the weak observability of the heading angle, the fine alignment cannot prevent the heading from gradually drifting.

### 3. TESTS

Long-term testing of the real-time performance of the system has been conducted in the laboratory. The focus of testing included stability of the multi-threaded firmware, real-time decoding of the GPS and INS data messages, real-time time synchronisation, multiple-stage circular buffering, float-pointing calculation, stability of the Kalman filter, button interrupts and response, CF file I/O, data output through additional UARTs, and interfacing with GoogleEarth.

With the 50MHz system design, the timing resolution of the counter is 5.12 $\mu$ s. To compare the time derived by the FPGA device with the time of the INS output, the FPGA-based system has demonstrated time-sync accuracy of better than 0.3ms. The system has potentially higher accuracy because it can reveal the C-MIGITS II's 10 $\mu$ s/sec clock drift, as analysed in Li *et al* (2006).

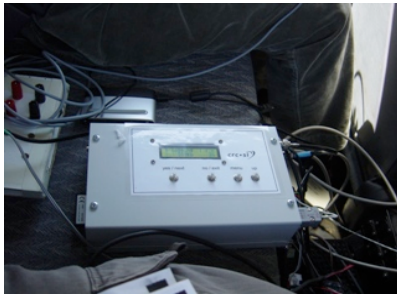
#### 3.1 Static test

An important requirement of the system is to estimate the attitude angles and the inertial sensor biases. Static data is used to evaluate this capability because the tilt angles are invariant during the test. Without correction from the Kalman filter estimates, the strapdown attitude solution gradually drifts with time. In the test, a 160-second alignment period is pre-defined. The zero-velocity is used to update the attitude and sensor errors in the alignment. Once the alignment process has been completed, the system automatically changes to the navigation mode where the GPS position data is used to update the estimate. The tilt solution converges gradually during the alignment phase, and remains stable within a small range of  $\pm 0.05$ deg during the navigation phase.

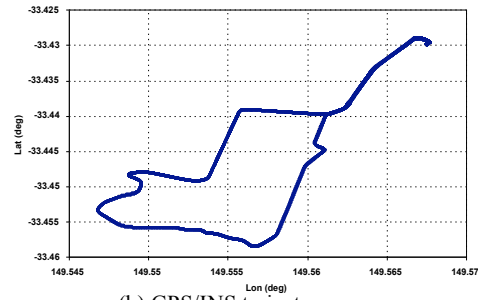
#### 3.2 Van test

Kinematic data was collected along roads in a variety of environments, including a race track with significant attitude manoeuvres, a highway, forested mountain areas with GPS signal blockage, and also through tunnels.

The device setup *in a test* is shown in Figure 5a. The ground trajectory depicted in Figure 5b shows the result from the test carried out around the Mount Panorama racing circuit, Bathurst, in the state of New South Wales. The velocity and attitude solution is depicted in Figure 6.

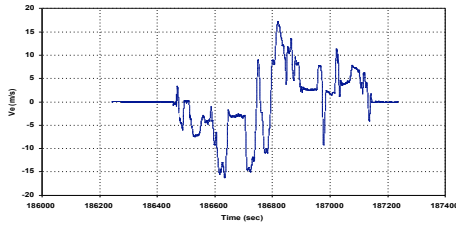


(a) FPGA-based GPS/INS system

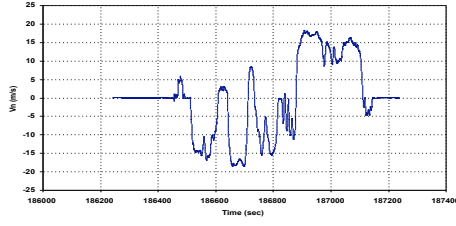


(b) GPS/INS trajectory

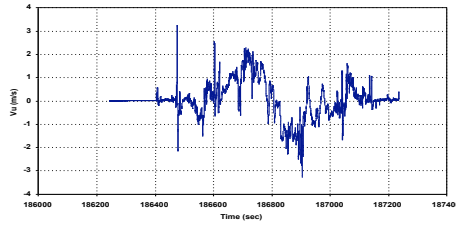
**Figure 5.** The device in the test and trajectory from the GPS/INS solution



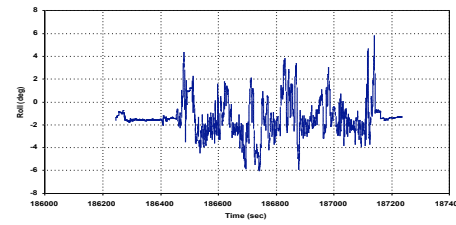
(a)  $V_e - t$



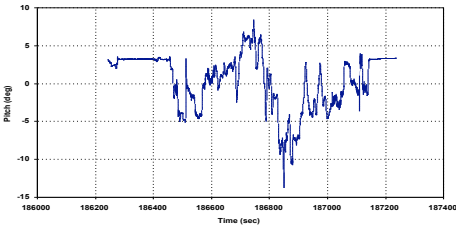
(b)  $V_n - t$



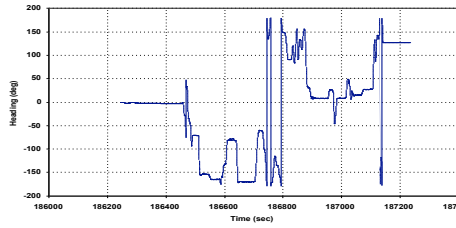
(c)  $V_u - t$



(d) Roll - t



(e) Pitch - t



(f) Heading - t

**Figure 6.** Velocity and attitude solution

In comparison with the INS-only solution, the integrated velocity solution (Figures 6a – 6c) is stable and correctly reflects the movements of the vehicle – backward and forward when the vehicle is driven from the parking site, stops to wait for the traffic light, and speeding up. The integrated attitude solution is also stable, and properly reflects the angular movement of the vehicle – especially on pitch (6e) and heading (6f). The tilt angles (roll in 6d and pitch in 6e) converge when the vehicle is static, however the heading (6f) remains on the initial value with

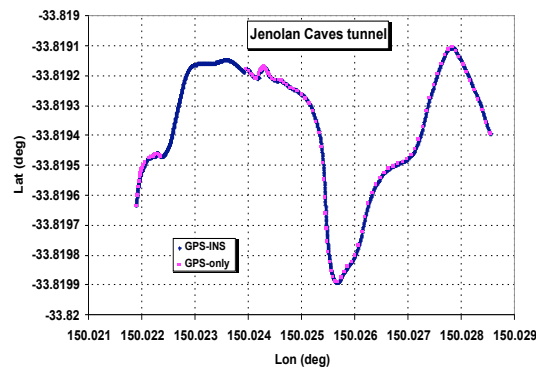
slight drift during the static period. The heading becomes observable when the vehicle moves and the GPS-derived velocity vector is then used to initialise the heading. From Figure 6f it is easy to see that the heading quickly changes from the initial value to the correct value when the vehicle starts to move. In the last section of the heading curve in Figure 6f, the heading has a jump of about 126deg in comparison with the initial value. In comparison with the compass data collected in the test, the angle at the end reflects the correct heading direction. Because the vehicle starts and stops at the same site and heads in the same direction the correct heading angle at the end demonstrates that the integration Kalman filtering works properly, at least in a qualitative sense.

### 3.3 Performance in tunnels

One test was performed in the Jenolan Caves area near Bathurst. The road goes through a tunnel near the Jenolan Caves township. The tunnel has a length of 197.7m in east, and 32.8m in north. It took 45sec to drive through the tunnel, including the time waiting due to traffic. Figure 7a shows the entrance to the tunnel, and Figure 7b depicts the trajectory from GPS and GPS/INS solutions. During the 45-second GPS outage in the tunnel, the INS solution correctly outlines the shape of the tunnel. This result demonstrates that the integration system is working satisfactorily – once an accurate navigation solution and the inertial sensor biases have been estimated before the vehicle enters the tunnel.



(a) Entrance of the tunnel



(b) GPS/INS solution in the tunnel

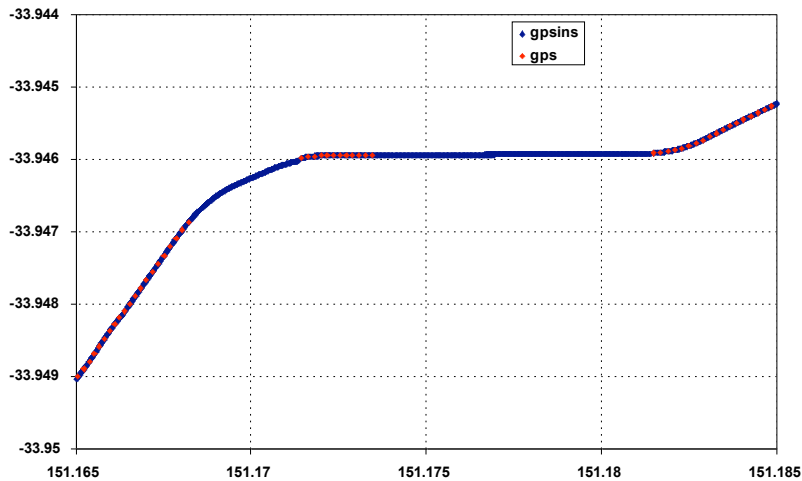
**Figure 7.** Comparison of solutions in the tunnel

The second test was performed in the Sydney Airport tunnel, as depicted in Figure 8. There are two successive GPS outages. The first 17-second GPS outage occurred under a bridge just 12 seconds before the car entered the tunnel, and then a 44-second GPS outage in the tunnel itself.



**Figure 8.** Sydney Airport bridge and tunnel

Figure 9 illustrates the integrated solution (in blue) and the GPS-only solution (in red). It can be seen that the two GPS outages are bridged smoothly.



**Figure 9.** GPS/INS solution bridges the outages under the bridge and in the tunnel

#### 4. CONCLUDING REMARKS

An FPGA-based real-time GPS/INS integrated system has been developed. A time-sync UART is designed to connect with the Nios II processor system to enable communication between the Nios II and the GPS and INS devices, as well as time-synchronise the GPS and INS data streams.

The embedded software has been developed using eCos – an open source embedded operating system. The software is programmed to implement multiple tasks; decoding the GPS and INS data streams, time synchronisation, strapdown inertial computation, and the integration Kalman filtering. With eCos support, the software implements the FAT32 filing system for CF card I/O, operation status display on the LCD, and button controls. The real-time solution is sent out via two additional UARTs and can be displayed on a GoogleEarth viewer. Long-term tests have demonstrated the functionality and operational robustness of the embedded software.

The GPS/INS integrated algorithm has been developed and tested in the laboratory and in the

field. The results have demonstrated that the integration Kalman filter estimates the inertial errors correctly, to compensate for the drift in the inertial solution. The results of the tests in several tunnels have shown that the corrected INS solution can bridge the GPS outages with reasonable accuracy.

## ACKNOWLEDGEMENTS

The project is funded under the Australian Cooperative Research Centre for Spatial Information (CRC-SI). The authors would like to acknowledge the support from the CRC-SI.

## REFERENCES

- Altera (2003) *Nios development board – reference manual*, Stratix edition, <http://www.altera.com>.
- Altera (2005) *Nios II software developer's handbook*, <http://www.altera.com>.
- Bar-Itzhack IY and Berman N (1988) Control theoretical approach to inertial navigation systems, *Journal of Guidance, Control and Dynamics*, 11 (3): 237-245.
- Boeing North American Inc (1997) *User's manual of C-MIGITS II*.
- Buck TM, Wilmot J, and Cook MJ (2006) A high G, MEMS based, deeply integrated, INS/GPS, guidance, navigation and control flight management unit, *Proceedings of IEEE/ION PLANS 2006*, San Diego, California, 25-27 April.
- Kennedy S, Hamilton J, Martell H (2006) Architecture and system performance of SPAN – NovAtel's GPS/INS solution, *Proceedings of IEEE/ION PLANS 2006*, San Diego, California, 25-27 April.
- Li Y, Mumford P, Wang J, and Rizos C (2006) Development of a GPS/INS integrated system on the field programmable gate array platform, *Proceedings of ION GNSS 2006*, Forth Worth, Texas, 26-30 September, 2222-2231.
- Hidalgo JI, Fernandez F, Lanchares J, Sanchez JM, Hermida R, Tomassini M, Baraglia R, Perego R, and Garnica O (2003) Multi-FPGA systems synthesis by means of evolutionary computation, *GECCO 2003, LNCS 2724*, E. Cantu-Paz et al (Eds.), Springer-Verlag, Berlin, 2109-2120.
- Massa AJ (2002) *Embedded software development with eCos*, Prentice Hall Professional Technical Reference, New Jersey.
- Meyer-Baese U (2001) *Digital signal processing with field programmable gate arrays*, Springer-Verlag Berlin.
- Mumford P, Li Y, Wang J, Rizos C, and Ding W (2006) A time-synchronisation device for tightly coupled GPS/INS integration, *Proceedings of IGNSS Symposium 2006*, Holiday Inn Surfers Paradise, Australia, 17-21 July.
- Nios Community Forum (2005), eCos for Nios II, <http://www.niosforum.com/>.