Open access • Proceedings Article • DOI:10.1109/PLANS.2010.5507243

# DINGPOS, a GNSS-based multi-sensor demonstrator for indoor navigation: Preliminary results — Source link

G. Vecchione, Damian Palenzuela, Manuel Toledo, Jose J. Rosales de Caceres ...+8 more authors

**Institutions:** Alenia Aeronautica, Autonomous University of Barcelona, European Space Research and Technology Centre

**Published on:** 04 May 2010 - IEEE/ION Position, Location and Navigation Symposium

**Topics:** Indoor positioning system, Air navigation, GNSS applications, Satellite navigation and GNSS augmentation

# DINGPOS, a GNSS-based Multi-sensor Demonstrator for Indoor Navigation: Preliminary Results

Giovanni A. Vecchione, Damián Palenzuela, Manuel Toledo, José J. Rosales de Cáceres
GMV Aerospace and Defence, S.A.
c/Isaac Newton, 11
Tres Cantos, Spain

Damien Kubrak, Yves Capelle
Thales Alenia Space
26 Avenue Jean-François Champollion
31100 Toulouse, France

Gonzalo Seco, Jose López-Salcedo
Universitat Autònoma de Barcelona (UAB)
Engineering School, Campus Universitari s/n
08193 Bellaterra (Barcelona), Spain

Peter Tiley
Advanced Digital Institute
The Waterfront, Salts Mill Road
Saltaire, Bradford, UK

Angelo Consoli
Saphyrion
Via San Maurizio
6934 Bioggio, Switzerland

David Jiménez-Baños, Gustavo López-Risueño
ESA/ESTEC
Keplerlaan 1, 2200 AG
Noordwijk, The Netherlands

*Abstract* — **The goal of this paper is to present the architecture and first performance results of the DINGPOS platform. The DINGPOS platform (Demonstrator for INdoor GNSS POSitioning) is a project funded by the ESA that covers the design, development and integration of an experimental indoor positioning system based on the fusion of three different technologies: High Sensitivity GNSS (GPS and the future Galileo), MEMS-based Pedestrian Navigation System and WIFI. This paper introduces the different architectural trade-offs of the final platform and presents the first results of its performance and capabilities with real data.**

***Keywords: Indoor navigation, HS-GNSS, sensor hybirdization***

## I. Introduction

Accurate and robust indoor localization is one of the most demanding challenges that the navigation research community faces today. Under the umbrella of this research line, numerous indoor localization solutions based on different technologies and several degrees of sophistication have been proposed. Nowadays, the trend is to make the most of the different signals available, plus the measurements coming from autonomous devices that do not rely on external signals. Examples of useful signals are GNSS, WIFI or UWB, among others. Examples of autonomous devices are, for example, Pedestrian Navigation Systems (PNS), step counters pedometers or odometers. This approach has the advantage that the navigations systems do not depend on a single technology or on the availability of signals: the navigation system uses at each epoch all the available information to estimate the position of the user.

Under this scheme, Global Navigation Satellite Systems (GNSS) present a very good opportunity for general positioning due to availability of signals and the global coverage that they offer. However, GNSS has been designed for open sky environments where perfect line-of-sight can be established between visible satellites and the user terminal. In presence of unfavorable environments, like urban canyons or indoors, GNSS signals are affected by severe attenuation, multipath effects and near-far interference. These working conditions are much more demanding than the ones assumed in the original design of the system, thus making not possible for traditional GNSS receivers to operate normally. In order to circumvent this limitation, advanced signal processing techniques are required to be implemented at the user terminal, leading to the concept of high-sensitivity GNSS receivers (HS-GNSS). A robust positioning platform for harsh environments can then be implemented not only by using HS-GNSS, but also by combining this technology with data coming from external sensors (e.g. MEMS) and communication infrastructures (e.g. WIFI, cellular networks). The use of GNSS is justified because it provides the extra feature of having a seamless navigation platform; this is, the capability of the navigation system to provide position in both indoor and outdoor environments.

In this context, the DINGPOS project proposes a navigation platform based on the fusion of three different technologies: GNSS (GPS and the future Galileo), Pedestrian Navigation

978-1-4244-5037-4/10/$26.00 ©2010 IEEE 

System based on MEMS inertial sensors and WIFI. First introductory papers with the preliminary architecture approach and results based on simulations might be found in references [1] and [4].

Regarding the GNSS module, the system can use different modulations for the GNSS; in particular it can work with GPS L1, Galileo E1, GPS L5 and Galileo E5a. The platform allows working simultaneously with L1/E1 frequency and either E5 or L5. This has made possible to evaluate the performance for different signal formats and different combinations of sensors, while taking advantage of the innovative features of the Galileo signal such as better multipath resistance or higher data rate. The acquisition of GNSS signals with C/N0 values down to 5dBHz has been achieved using aids provided by an outdoor receiver to the indoor receiver. This very challenging sensitivity is reached using outdoor tracking loops that provide both carrier and code phases to the indoor receiver.

The GNSS module of the platform has been developed with an innovative architecture using GPUs (Graphics Processing Units) and CUDA language (Compute Unified Device Architecture) to take advantage of parallel computation. CUDA is a general purpose parallel computing architecture that leverages the parallel compute engine in NVIDIA graphics processing units (GPUs) to solve complex computational problems. Using CUDA, the latest NVIDIA GPUs effectively become open architectures like CPUs; unlike CPUs however, GPUs have parallel "many-core" architecture, each core capable of running thousands of threads simultaneously. This architecture is perfectly suited for GNSS signal processing, since several satellites are to be acquired simultaneously, and extensive use of FFT processors is required for an exhaustive search both in time and frequency. In these circumstances, GPUs can offer large performance benefits, and the use of the CUDA engine allows parallelizing these operations speeding up all the system.

The Pedestrian Navigation System (PNS) developed for the DINGPOS system is an autonomous navigation system that provides estimations of the speed and heading of the user. It is based on Xsens MEMS inertial technology and is connected to the platform through a USB port.

The WIFI module is able to estimate absolute positions based on the power of the received signal and a fingerprinting map.

Thus, the DINGPOS platform is capable of providing continuous navigation in indoor environments, by synchronization and hybridization of the inputs from different sensors into a Complementary Extended Kalman Filter. These data hybridization techniques make possible to combine the advantages of a high sensitivity GNSS receiver (capable of mitigating interferences and detecting cross-correlation, near-far and multipath effects) with WIFI and the MEMS-based PNS. Other methods, such as indoor Map Matching algorithms are also considered. The architecture for the hybridization module is based on tight coupling (pseudoranges and Doppler measurements are used in the filter) and close loop approach (final PVT solution is fed back to the INS module for error calibration). DINGPOS platform allows the user to select the sensors with which to navigate, so that it is possible to run in different navigation modes.

The paper is structured as follows: After the introduction, the second section presents the different architectural trade-offs of the final platform, covering the general architecture, the different components and the data flow management and fusion approach are presented. After this section, the different modules are presented independently. This is, the GNSS, the PNS and the WIFI modules.

The third section covers the GNSS whole chain, going from the antenna to the SW receiver, passing from the GNSS Front-End developed ad-hoc for the DINGPOS platform. Special attention is paid to the and the computational aspects linked to the High Sensitivity Galileo algorithms used in the platform.

The fourth section is devoted to the PNS. Besides the description and basic principles of the both modules, some results for the PNS on standalone mode are presented.

Finally, the main conclusions are provide the interpretation of the test results, basically the assessment of capability of the platform to navigate in indoor environments and to support further research in indoor navigation.

## II. Architecture trade-off of the DINGPOS platform

### A. *Platform architecture*

Figure 1 shows the functional diagram of the DINGPOS platform architecture. The following subsystems can be identified: a Galileo/GPS receiver, Sensor units, a Control and Data Logging Unit (CDLU) and a Real Time Monitoring and Control (RTMC) unit.

The Galileo/GPS receiver includes all the SW and HW from the RF front end to the PVT computation output. It is the basis of the platform development on top of which the entire set of algorithms, including the global Kalman filter, will be tested. Additionally to these subsystems, Figure 1 presents a preliminary selection of the external wireless communication systems that will be used to provide assistance data or stand-alone location technique, which can be GSM/GPRS or WIFI.

The two GNSS RF Front Ends, the MEMS, WIFI and UWB sensors are connected to the platform through a number of USB interfaces. Moreover, the platform is connected to the Location server via an Ethernet interface and an IP router.

The platform where all the hardware in connected and all the software runs, has the following processing characteristics:

- 1 Mother board Intel® Core™ i7 Extreme Quad 45nm
- 1 Intel® Core™ i7 920 2.66 GHz FSB 1066Mhz, 45nm 8MB L2
- 3 x 1GB DDR3 1333MHz SDRAM

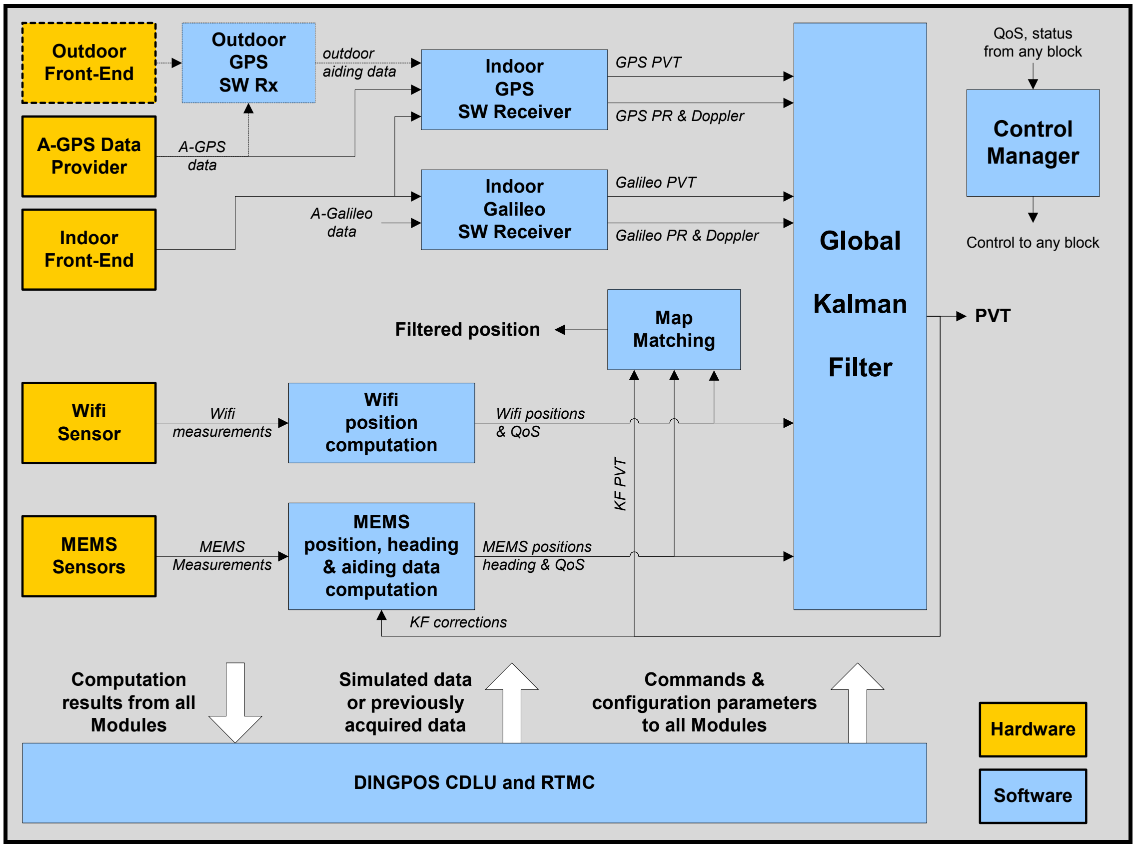


**Figure 1: DINGPOS architecture overview**

- 1 NVIDIA Quadro® FX 380 PCIE 256MB
- 2 NVIDIA TESLA C1060 Compute Board

Note that the platform is equipped with two GPU. The use of GPUs for data processing is motivated by their capability of massively parallel processing capability. This feature is especially useful to tackle the computational complexity of the GNSS signal processing.

From a software point of view, the different algorithms are controlled via a Real Time Monitoring and Control Unit. Each sensor is associated to a dedicated algorithms block that provides the corresponding standalone PVT solution. How this information is used, is responsibility of the Global Kalman filter. Finally, a specific map matching algorithm enhances the performance of the different PVT computation modes.

### *B. Hybridization approach and RTMC*

The hybridization module is responsible of blending all the signals available to estimate an optimal state. Under this scheme, the selected algorithmic approach was the use of an extended Kalman Filter (EKF). In this context, the strategy for data hybridization is the tight-coupling, where the GNSS raw measurements are blended with both the PNS and WIFI measurements. The motivation behind this selection is to make the most of the GNSS. Note that the DINGPOS platform is designed to estimate position in indoor environments, where the GNSS signals is received very attenuated and suffering from several errors such as severe multipath. With the tight-coupling approach, the system wins in robustness and also allows the platform to use the GNSS in case there are less than 4 satellites in view.

Moreover, the implementation considers a feedback from the EKF module to the PNS module. This feedback provides to the PNS module with error estimates to correct the raw measurements, thus preventing the PNS error to grow rapidly.

The Real Time Monitoring and Control manager (RTMC) is in charge of managing all the data and information flows, centralizing all the communications between the different modules of the DINGPOS system and issuing commands to the system (e.g, start and stop). It collects and forwards the data from one module to another and continuously monitors the state of the system. In the case of failure, it also decides what to do and how.

This flexible and modular architecture to manage the system configuration and the data flow and communications between modules allows the system to be upgraded with more sensors or instruments in the future.

This architecture has been implemented and initial results with simulated data (

Figure 2) were shown in [3].

Next sections will cover some preliminary results of the platform with real data.

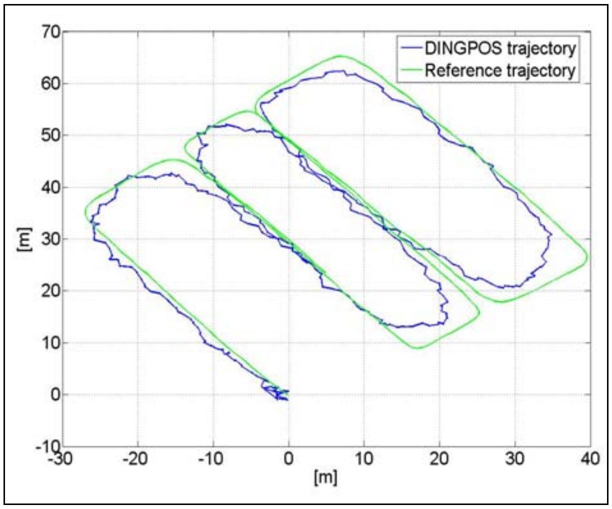


**Figure 2: Full integrated DINGPOS hybrid trajectory.**

## III. GNSS MODULE

### *A. GNSS Front-End*

The hardware of the DINGPOS system consists of an FPGA development board, a mezzanine board carrying various base-band and interface functions and an easily exchangeable GNSS RF front-end board.

As FPGA board, a commercial development board has been selected in order to reduce development risk, time and costs. The FPGA is a 200k gates ALTERA Cyclone III device. On the FPGA all digital acquisition, control and data formatting functions have been placed, while space is still available for possible enhancements or upgrades.

In order to implement all functions not present on the FPGA development board, a mezzanine board that plugs directly onto it has been designed. This board carries clock generation, power supply and a USB interface. As clock generator a 10MHz OCXO with 25ppb temperature stability has been used. Power conditioning and a clock buffer were used to prevent possible frequency shifts due to supply or load variations. In addition to the on-board oscillator, the system can also accept an external clock signal. A USB 2.0 interface which provides communications with the host workstation has been also mounted on this mezzanine board, as well as power supply regulation and distribution. The system also generates a 100Hz reference signal.

The RF front-end has been designed as a small separate module. This allows for an easy way to swap RF modules for different GNSS signal frequencies and bandwidths. Taking advantage of SAPHYRION GNSS RF chip-set consisting of SY1007 (RF downconverter) and SY1017 (ADC and interface ASIC) , a well performing RF front-end module with a size of just 40mm x 25mm has been designed. With the proper configuration, this RF module can receive all GPS, Galileo and GLONASS signals up to a bandwidth of 25MHz, sampling them with a resolution of 3 +3 bits in quadrature.

The complete DINGPOS system has been housed in an extruded aluminum box (225mm x 130mm) which provides for mechanical robustness, RF shielding and protection.

### *B. HS-GNSS algorithms*

The difficulties in using GNSS for indoor positioning come from the fact that GNSS have been designed and dimensioned for outdoor operation. When moving to indoor environments, GNSS receivers are forced to operate in much more demanding conditions than the ones assumed in the nominal design of the system. Mainly, severe attenuation, multipath components and near-far effects degrade the received signal. In these circumstances, reliable code acquisition with a traditional GNSS receiver is nothing but a utopia. The only possible alternative is to resort to advanced signal processing techniques and rely on the implementation of high-sensitivity GNSS receivers (HS-GNSS) such as the one designed within the framework of the DINGPOS project.

Since the main source of indoor signal degradation is the severe attenuation caused by the ceiling, walls and other kind of indoor obstacles, one of the main goals of the DINGPOS' HS-GNSS receiver is to extend the overall correlation interval. However, because of the presence of phase/frequency uncertainties in the received signal, coherent integration during code acquisition must be restricted to just a limited time window. Since this short coherent integration is not enough for reliable detection, the receiver is forced to extend the correlation interval by using several noncoherent pre-detection integrations. By doing so, the overall correlation interval can be extended far beyond the bit interval, thus increasing the pre-detection signal-to-noise ratio without requiring the knowledge of the navigation message.

The efficient implementation of this correlation is a key milestone of the HS-GNSS receiver. The code acquisition module has been carefully designed so as to make an extensive use of FFT operations, thus favoring the subsequent implementation in GPUs. The double-FFT method [5] has been selected, which can be understood as the optimal implementation of the time-frequency matched filter to the received signal. Two are the main advantages of this scheme. First, it makes extensive use of FFT processors for the joint and efficient implementation of both the input correlation and the fine frequency search. Second, it performs the maximum possible coherent integration in the absence of bit assistance. That is, one whole bit period of 20ms. The result is acquisition architecture with superior performance compared to traditional schemes.

The HS acquisition module is the core element of the HS-GNSS receiver, for which the overall flow diagram is shown in

Figure 3. The starting point is given by the input of data samples coming from the GNSS front-end. The first task to be performed is a test to detect (and remove) any possible

narrowband interference that may be present in the incoming data. Although narrowband interferences may not be a problem at all for an outdoor GNSS receiver (due to the inner spreading gain of most GNSS spread-spectrum signals), for an indoor receiver, interference power levels may be on the order of 100dB higher than the GNSS signal power. Thus, some precautions must be taken to avoid the signal degradation at the correlation output. Once this check has been performed, acquisition is undergone via the double-FFT method previously mentioned. At this step, assistance information from the Thales-Alenia location server is incorporated. The basic set of assistance parameters includes: the list of visible satellites, their corresponding Doppler error, ephemeris for enabling the user's position determination and the position of the server itself.

Based on these assistance data, the receiver carries out the correlation with the input samples and a fine frequency search around the assisted Doppler value. The frequency search covers a range of +/- 500 Hz with 25 Hz resolution. The output correlation samples are stored in a four dimensional matrix(hypercube) with dimensions corresponding to the number of samples per code, the number of fine frequency bins and the number of possible bit transitions hypotheses. This matrix is then noncoherently integrated with the aim of extending the overall correlation interval without being affected by unknown bit transitions. Once noncoherent integration is finished, noise floor normalization is required to ensure that all frequency bins share the same level of noise power spectral density. Otherwise, different noise levels would lead to an increased probability of false alarm when evaluating the signal detection threshold.

Signal detection is required to determine whether the satellite of interest is present or not. In case of failure, acquisition is restarted for a new satellite. In case of success, a final test is undergone to ensure that signal detection was really caused by the presence of the desired satellite and not because of near-far effects (i.e. the presence of another satellite with stronger signal power). The near-far validation is carried out by analyzing the received signal statistics and comparing them to the case where near-far is absent or present. Finally, assuming that no near-far is present, some refinements are performed onto the output acquisition data. They mainly consist on resampling the input signal with the estimated fine Doppler and bit-level code phase error. Then, five correlation points are recalculated around the maximum correlation peak. These points serve as the basis for interpolating a more accurate code phase value, and thus providing precise pseudoranges to the position determination module. Such a determination is implemented by means of the Peterson's method [6]. This is a two-step procedure where first, the position of the acquired satellites is calculated and second, the user's position is determined by linearizing the pseudoranges variation. At this second step, information regarding the satellites velocities is also included to overcome the pseudorange ambiguity.

The described operation of the HS-GNSS receiver can be applied to either GPS or Galileo signals. Indeed, the DINGPOS HS-GNSS receiver is fully compatible with GPS L1/L5 and Galileo E1/E5a signals. Compared to the traditional GPS-L1 signal, the features of the new signals are mainly the incorporation of pilot symbols in the quadrature component and a tiered spreading structure with a primary and a secondary code. Both features have been easily incorporated into our HS-GNSS receiver. For the case of pilot symbols, they allow the receiver to focus on the quadrature component only and extend the correlation interval in a coherent manner so as to take advantage of the absence of bit transitions

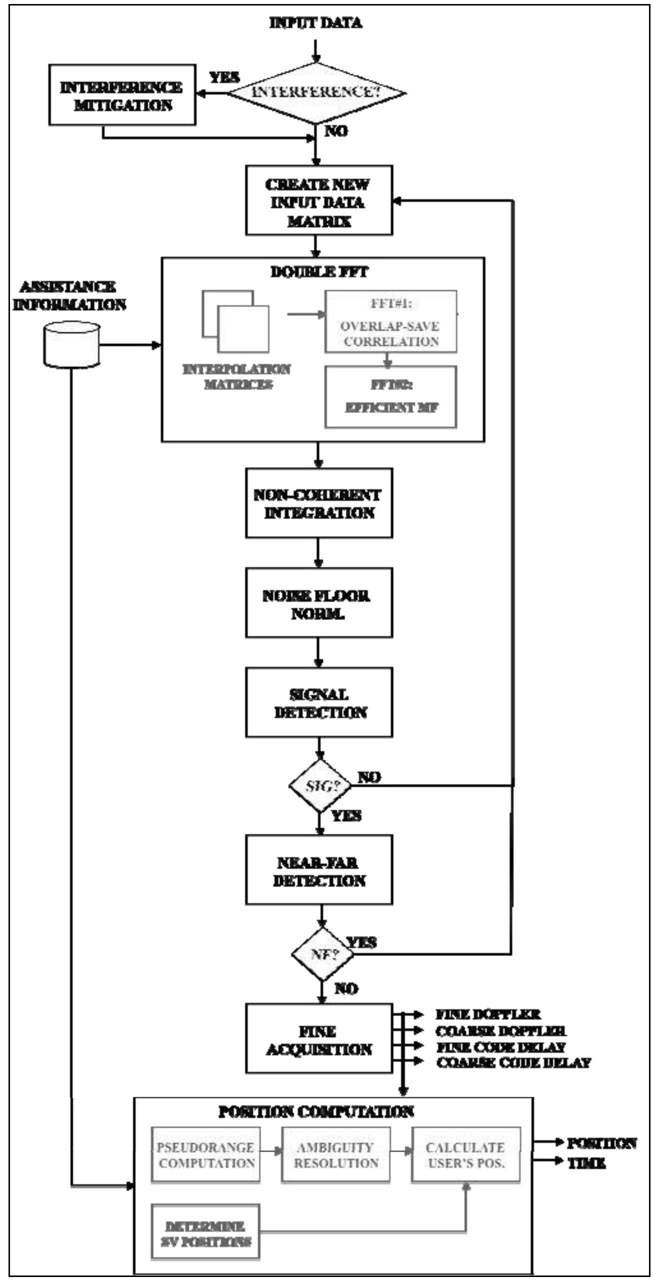


**Figure 3 Flow diagram of the DINGPOS HS-GNSS receiver.**

There exist many contributions in the current literature for the combination of both the in-phase and quadrature components (i.e. data and pilots). In our receiver, however, only the quadrature signal is processed. This saves some extra hardware complexity, and for long enough integration intervals (as it occurs in HS-GNSS receivers), the contribution of combining both data and pilot signals can be considered to be marginal. For the case of the tiered spreading structure, it is interesting to realize that the primary code of Galileo or GPS-L5 signals plays the same role as the traditional GPS-L1 C/A code does. Then, searching for the secondary code of Galileo or GPS-L5 signals is indeed equivalent to search for the bit transition in GPS-L1 signals. Only minor modifications were introduced in this search to allow sign changes from primary code to primary code. Please note that several parts of the previously explained algorithms and identified in Figure 3 are protected under ESA patents.

### *C. GNSS performance*

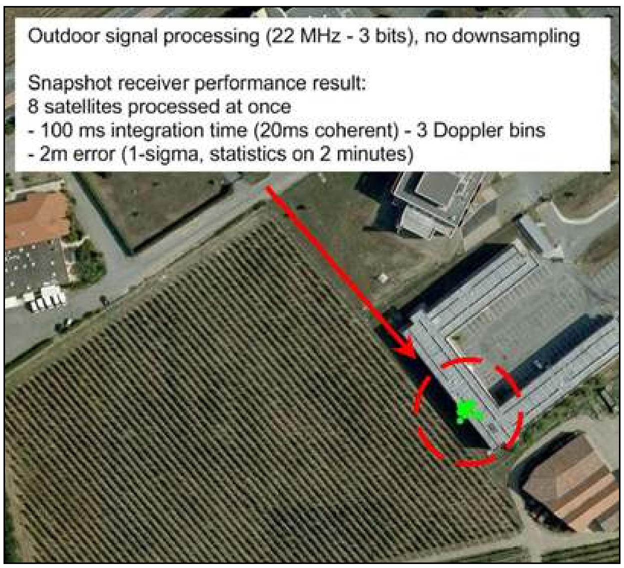


**Figure 4 : GPU-based GPS L1 C/A Performance Assessment during development.**

The implementation of the receiver is being evaluated with respect to key functions translated from a pure CPU code to a GPU one. Both processing are indeed very different as they involve different way of sequencing the GNSS algorithm.

For the time being, the receivers have not been tested against several scenarios, starting from the urban case to the indoor one. However, some tests have been conducted with a rooftop antenna to validate the GPU implementation of the GNSS algorithms. Results in the position domain are illustrated below in

Figure 4. In this test, the L1/E1 dataflow with no down sampling is processed sequentially on the basis of a 100ms integration time (20ms coherent integration) to test the overall CPU load. 8 satellites are processed at each snapshot epoch, and 3 Doppler bins are tested once a first position is computed. Statistics on 2 minutes of data collection show an accuracy of about 2 meters (1-sigma) from the position of the reference antenna.

During this test, it has been observed a mean CPU charge of 24% and peak CPU charge of 51%, which takes into account all the computation stages from the raw data parsing to the position computation.

Some indoor tests have been performed inside Thales building. In the picture some test with indoor acquisition is shown. The test has been done into a two floor building. At first floor the standard deviation computed is in the order of 7 meters with a mean C/N0 of 28 dbHz.

**Figure 5 : GPU-based GPS L1 C/A Indoor Performance.**

### *D. Computational issues*

One important feature the platform has to deal with is first the management of the dataflow coming from the front-end. The maximum datarate is 132 Mbps, corresponding to the

L1/E1 3-bit @ 22 MHz. Parsing and downconverting 1 second of such data requires approximately 1 + 5 seconds in the case the processing is managed by the CPU in a traditional way. However, taking advantage of the GPU card, and the fact data parsing and down converting are well adapted to parallel processing it has been possible to reduce the processing time to 195 ms, with still room for improvement.

A part of the correlation engine, the time required to process 8 satellites over 500 ms, including 3 Doppler bin hypotheses is approximately 480 ms using GPU, including memory transfer from CPU to GPU and opposite, leading to a time to fix of 760ms.

The GPS/Galileo algorithm developed for DINGPOS is computationally extremely demanding. The requirements on memory and processing power make the task of running the algorithm in near real-time very challenging. In exploring ways of increasing the efficiency of possible implementations it became clear that the algorithm was highly suited to a parallel processing architecture. The opportunities for creating a highly parallelized implementation on a platform consisting of a single PC are limited to using either one or more multi-core CPUs or a Graphics Processing Unit (GPU).

Recent advances in GPUs have seen their processing performance increasing dramatically when compared to the increases observed for CPUs over the same time. A snapshot of processing capabilities taken around June 2008 shows that a fast CPU had a peak performance of around 100 GFLOP/s whilst a good GPU had a peak performance of just under 1000 GFLOP/s[2]. The potential computing power is only useful if it is made available to the scientific programmer, and an example of such a toolkit is NVIDIA's CUDA (Compute Unified Device Architecture). CUDA provides an interface through the C programming language, with extensions that expose the GPU's architecture to the programmer. A GPU works best when it is provided with thousands of concurrent tasks to work on, with minimal interaction between the tasks. This means that the implementation should work on the Single Instruction Multiple Data (SIMD) model. A potential bottleneck in an implementation that utilizes the GPU is the transfer of data between the CPU and GPU, so this should be minimized, preferably keeping intermediate results on the GPU and only transferring the final result back to the CPU. CUDA has been successfully used in fields such as medical imaging, computational fluid dynamics and oil exploration techniques. There are several examples of traditional multi-million dollar computing clusters being replaced with a single PC, which will contain one or more appropriate GPUs. An example of this is the Antwerp University ASTRA department which has replaced a traditional cluster containing more than 300 CPUs with a single PC containing 4 GPUs to perform their tomography calculations.

The GPS/Galileo double FFT algorithm is a good fit for parallelization on to GPU hardware. Large FFT operations are aided by having a CUDA implementation of FFT provided by NVIDIA, while the remaining aspects of the algorithm can be broken down into suitable blocks that have few dependencies between array elements. The largest data store, the hypercube, is needed only on the GPU, so is created and worked on solely on the GPU with no copying to and from the CPU.

The GPS/Galileo algorithm was ported from the prototype Matlab code directly to C and CUDA. The performance improvement observed is in the order of 150 times faster execution (e.g. a run time of 117 seconds in Matlab is reduced to a run time of less than a second on the GPU), which allows for a near real-time acquisition of the desired results. The GPU used for development and for the target system is an NVIDIA TESLA C1060, which has 240 processing cores and 4GB of memory.

In Figure 6 it is show the boost obtained using CUDA in comparison with CPU, it is evident the benefit of using GPU for parallel task like the case of an FFT Gnss receiver.

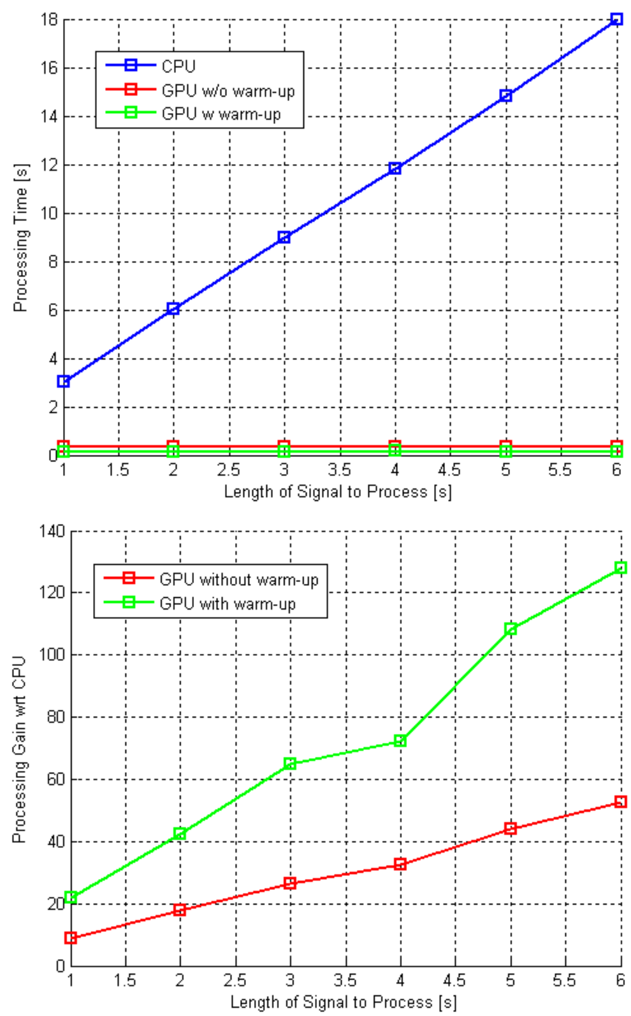


**Figure 6 Application to assisted GPS L1 C/A – standard non-coherent acquisition with 1satellite and 3 Doppler bins**

## IV. PNS MODULE

### A. Description

In order to reduce the impact of the sensors bias on the overall system performance and thus to increase the reliability of the navigation system, a pedestrian dead-reckoning mechanization has been preferred to the traditional INS mechanization. The PNS mechanization, as illustrated in figure below, requires the sensors unit to be closely attached to the pedestrian. According to medical researches (see [1] and [2]), one can establish a relationship between the velocity or step length of a walking pedestrian and some parameters that characterise the acceleration experienced by this pedestrian

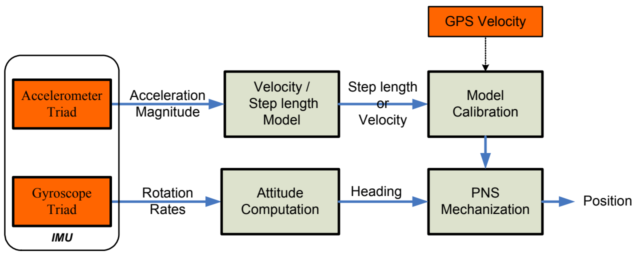


**Figure 7: Pedestrian Navigation System Mechanization.**

Figure 8 shows the theoretical performance that can be achieved with the pedestrian mechanization, assuming a drifting heading source (blue plot) and a 5°-biased heading (red plot), and a constantly walking pedestrian.

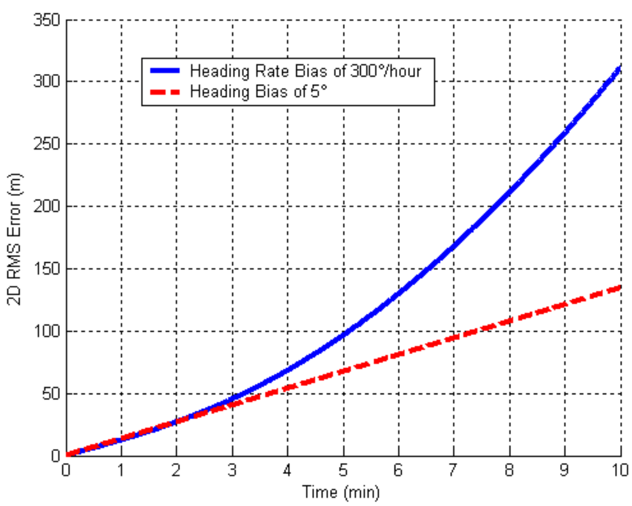


**Figure 8: 2D Upper Bound Position Error given a Constant Velocity Bias of 2m/s and several heading errors (pedestrian mechanization).**

### B. Standalone performance

A preliminary test was carried out to test the integration of the PNS module in the DINGPOS platform – this is, the performance of the platform when only the PNS is available.

Figure 9 shows the result of the test with the PNS integrated in the final platform. The test was carried out in GMV premises and lasted for three minutes. Figure 9 shows the trajectory estimated by the PNS and also the position every 10 seconds. It can be seen that the PNS is able to estimate the position with good quality for about one minute. After one minute, the solution degrades due to a bad heading estimation, but the pattern of the trajectory can still be identified.

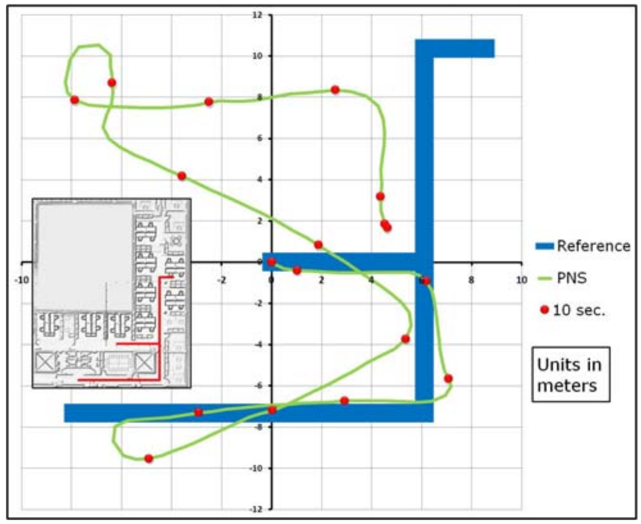


**Figure 9: Standalone PNS test**

One of the main goals of the Kalman filter is use all the available information (WIFI and GNSS) to correct this drift by feeding corrections to the PNS module.

## V. CONCLUSION AND FINAL REMARKS

This paper addresses the development and first results of the DINGPOS platform: a hybrid multisensor platform for indoor navigation: The approach to achieve ubiquous position for the DINGPOS platform is a tight coupling navigation in closed loop which will process as input measurements HS-GNSS pseudorange and Doppler observables, WIFI and heading and speed observations from a PNS model processing IMU MEMS raw data.

The DINGPOS platform integrates the following components:

- A front-end with integrated precise clock necessary to perform indoor acquisition
- High sensitivity GNSS SW-radio navigator
- An IMU-based Pedestian Navigation System
- RSSI WIFI positioning technology

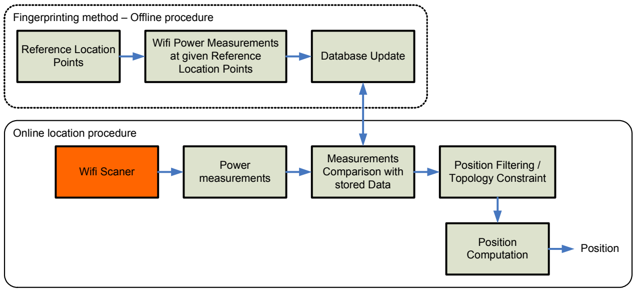


**Figure 10 : Wifi RSS-based Location Technique Principle.**

In this context, each sensor plays a particular role in the platform and has its advantages and drawbacks. The idea behind the election of these sensors is that each sensor compensates the drawback of other, is such a way that the combination of all them enhances the overall performance of the platform. Thus, the GNSS provides global coverage, but it is not designed to navigate in indoors environment. On the other hand, the PNS is autonomous, but the positioning error grows unbounded in a short period of time. And finally, WIFI offers good and robust positioning indoors, but its coverage is limited to a particular place and is very sensitive to the environment.

The platform is currently in its last stage of the integration phase and some preliminary tests with the PNS and the GNSS have been conducted in real time and with real data. Some preliminary results based in simulations were presented in [3].

Future work is mainly concerned with completing the platform integration and the experimentation in different environments.

## ACKNOWLEDGMENT

The authors would like to extend their thanks to the European Space Agency which is funding the project.

## REFERENCES

[1] Ladetto. On foot navigation: continuous step calibration using both complementary recursive prediction and adaptive Kalman filtering. Proceedings of the ION GNSS 2000.

[2] V. Gabaglio. Ph.D. Thesis Report, EPFL

[3] J. López-Salcedo et al., "DINGPOS: A Hybrid indoor navigation platform for GPS and GALILEO", Proceedings of GNSS ION 2008

[4] J. López-Salcedo et al., "Demonstration of ubiquitous positioning of WIFI, INS and assisted HS-GNSS hybridization", NAVITEC 2008, 4th ESA Workshop on Satellite Navigation User Equipment Technologies

[5] D. Jiménez-Baños, N. Blanco-Delgado, G. López-Risueño, G. Seco-Granados, A. García-Rodríguez, "Innovative techniques for GPS indoor positioning using a snapshot receiver", *Proc. 19th ION tech. meeting*, 2006.

[6] B. Petterson, R. Hartnett, G. Ottman, "GPS receiver structures for the urban canyon", Proceedings of ION GPS 1995of ION GPS 1995.

[7] NVIDIA CUDA Programming Guide Version 2.3 July 2009