Terrain Referenced Navigation Using a GPS Approach

Patrick Oonincx & Daniela Vaman

Introduction

To navigate, a position estimation system is needed. Inertial Navigation System (INS), Satellite based navigation such as the Global Positioning System (GPS) and Terrain Referenced Navigation (TRN) represent three different types of technology to obtain position estimates.

An INS uses measurements of acceleration to compute relative velocity and displacement. Accelerations and rotations are measured along three perpendicular directions. This type of technology was already being used in the early sixties for the guidance of Intercontinental Ballistic Missiles (ICBMs). Figure 1 (left) shows a picture of the INS used in the Titan II missile. Today's state of the art in acceleration sensors allows the realization of a solid state INS with a weight below one gram. Figure 1 (right) shows an example of a solid-state accelerometer integrated on a miniature circuit board.

Already, more than 35 years ago, the first TRN system, using a radar altimeter and a terrain elevation database for position estimation, was tested on a Hound-Dog missile. The first GPS satellite was launched more than 30 years ago.

All three types of navigation technology are used in military systems. For example, the AGM-86B cruise missile uses a terrain contour-matching guidance system, combined with an INS. A later model, the AGM-86C/D uses an onboard GPS coupled with its INS. With GPS, the vast consumer-electronics market has been a strong driving force behind innovations in

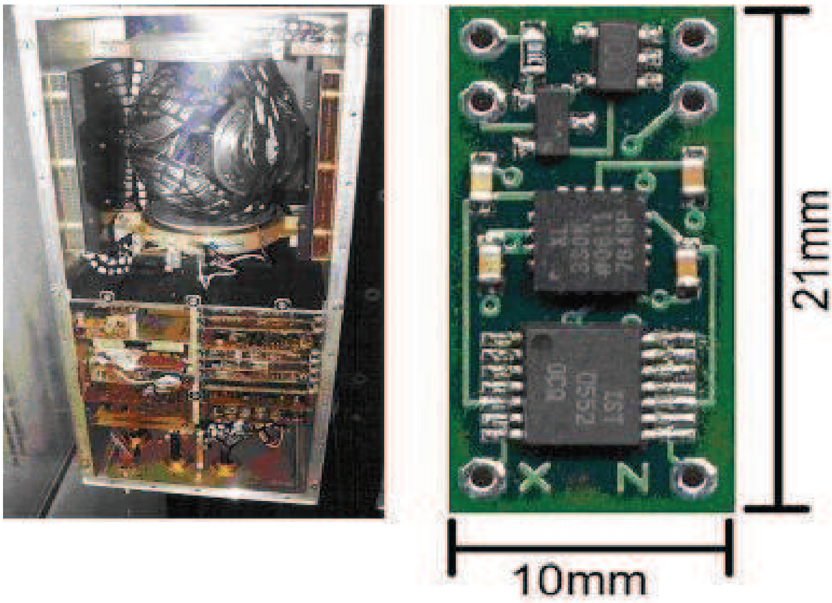

Figure 1: INS used in a Titan II Intercontinental Ballistic Missile (left) and a solid state 3-axis accelerometer with I/O circuitry (right).

the area of GPS-based applications, and has thus provided the business case for the continuous development of receiver technology. However, for Terrain Referenced Navigation (TRN), such a vast consumer market has never existed. With the advent of GPS, some of the originally intended applications of TRN even ceased to exist. However, given today's state-of-the-art jamming technology it is fair to assume that the uninterrupted availability of GPS in future conflicts will be impacted.

Due to the dissimilarity with GPS and INS, an integration of TRN can increase the availability of accurate position estimates in situations where GPS is temporarily unavailable and the reduction of accuracy of the INS with an increase in time limits the operational capability. Although not interesting for consumer-type applications, certain military operations can benefit from such a capability.

The developments in the area of digital signal processing, combined with the continuing increase in computing performance have enabled the realization of coin-size GPS receivers. Developments in the area of acceleration sensors have provided the possibility to build solid-state INSs which weigh less than a single gram. The developments in the area of data storage enable the creation of hand-held devices containing world-wide hi-resolution elevation maps.

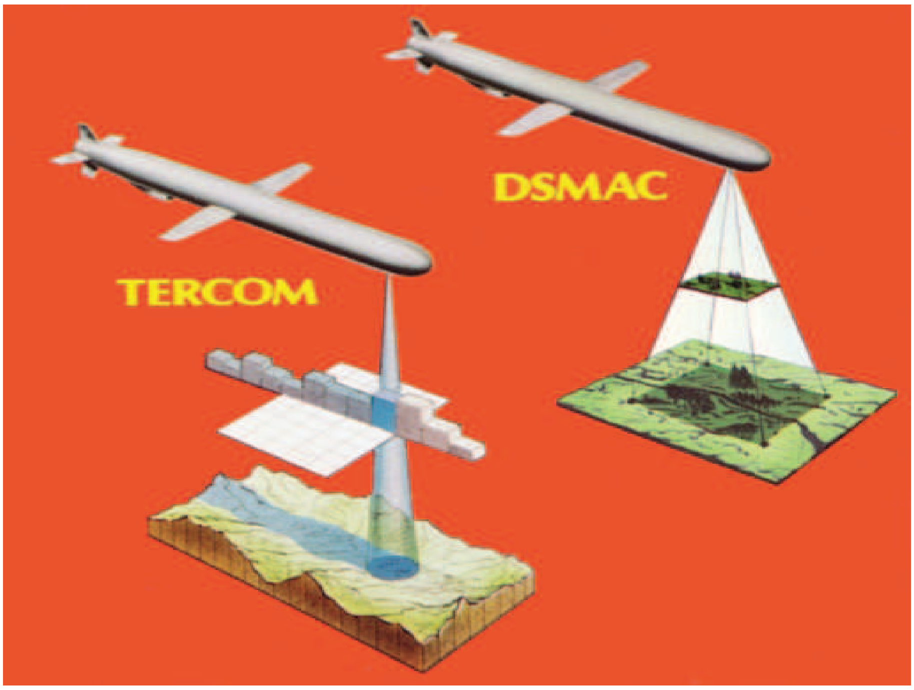

Figure 2: Two types of terrain related navigation for missile guidance: TERCOM generates position fixes by sampling terrain heights and correlating the elevation profile with ground truth information; DSMAC correlates stored images with snapshots of the terrain beneath the missile to generate a position fix.

With the availability of the technical components needed to realize a system that integrates GPS, TRN and INS, the challenge now lies on the development of the functionality which utilizes the available measurements to estimate the location. Application specific performance requirements regarding availability, accuracy, integrity, update-rate and susceptibility to intentional interference, in combination with specific possibilities for trade-offs, will cause the optimum combination to be determined by the intended application(s).

Terrain Referenced navigation vs. GPS

To be able to define both hardware and processing requirements for a given application, a better insight into the possibilities and limitations is needed. The goal of the research conducted within the NLDA's Navigation Technology department is to contribute to the creation of a design framework which can be used to match technology and processing concepts to a particular set of system performance requirements. Here the focus lies on exploring the potential of digital signal processing concepts for signal acquisition and tracking in the area of TRN.

TRN techniques provide the user's position by using a stored digital map of terrain elevation data together with real time measurements of a platform's height above the ground (altitude/depth). The basic rationale of the system is to find the best match between the features of the environment and the stored database, in the same way that a person would compare a landmark with a map, see Figure 2. The methods for obtaining position from the measurements can be divided into two categories: sequential (each measurement is processed separately) and batch processing (a series of terrain height measurements, known as transect, are processed together). The TRN system we present in this paper belongs to the second category.

In order to have a clear match between the measurements and the terrain map, it is necessary to have a good indication of the velocity and the orientation of the platform. When dealing with measured errors in the speed and course parameters we found similarities with the problems associated with the tracking of arrival times and Doppler offsets in a GPS receiver.

To estimate the user's position, velocity and time in a GPS receiver a series of different processes must be accomplished, namely capture the signals transmitted by the satellites, indentify the satellites in view, measure the signal's transit time and Doppler shift and decode the navigation message to determine the position, velocity and clock parameters. The transit time and Doppler shift are changing parameters in time, because of the movement of the satellites. However, once determined on a coarse scale it is possible to keep track of these parameters in an elegant way, because these parameters will only change slightly during each measurement. In a TRN system course and speed of a platform, e.g. a submarine, are also expected not to change heavily during a dense set of measurements.

Furthermore, with GPS the presence of a Doppler shift can cause the estimation of the right transit time of a satellite to deteriorate, while in the case of TRN an error in the estimate of the velocity deteriorates the correlation of the measurements and the terrain database. Similar to the correlation with multiple codes (satellites) in case of GPS, in TRN an uncertainty in the track of a platform will require the correlation with samples from a range of directions. These issues also apply to the tracking phase.

In the next section we will revisit GPS signal processing techniques, that will be used in our proposed TRN tracking algorithm.

GPS revisited

This section focuses on the digital processing stages that take place in a GPS receiver. After a short description of the GPS signal, following [1,2], we will recall the concepts necessary to understand the similitude between GPS signal processing and TRN, as briefly discussed before.

Each GPS satellite transmits messages continuously on two radio frequencies in the UHF band, referred to as link L1 and L2. Each GPS signal comprises three components:

- carrier (refers to the RF sinusoidal signal with frequency f_{L1} =1575.42 MHz and f_{L2}=1227.60 MHz),
- ranging code (refers to a family of pseudo random noise(PRN) sequences: each satellite transmits a unique (civil) coarse-acquisition C/A code on L1 and a precise (military) P code on both L1 and L2),
- navigation data (refers to the coded message consisting of data on the satellite's status, ephemeris and clock bias parameters).

For simplicity we will restrict ourselves here to signal models using only the C/A code on the L1 frequency. This comes down to

$$s(t) = \sqrt{2P}\, D(t)\, x(t)\, \cos(2\pi f_1 t + \theta_1),$$

with P the average power, $D(t)$ the navigation message and $x(t)$ the C/A code. The code $x(t)$ is of length 1023 bits and repeats itself every millisecond. The message $D(t)$ is of length 12.5 minutes and is transmitted at 50 bits per second. Both x and D are built up by a series of block function (chips) of amplitude 1.

In a GPS receiver, the incoming signal must be multiplied with a locally generated C/A code (corresponding to the specific satellite) and mixed with a locally generated carrier wave. Two parameters are needed: the (Doppler) frequency and the code phase of the incoming signal. The motion of the transmitter satellite relative to the GPS receiver causes the Doppler

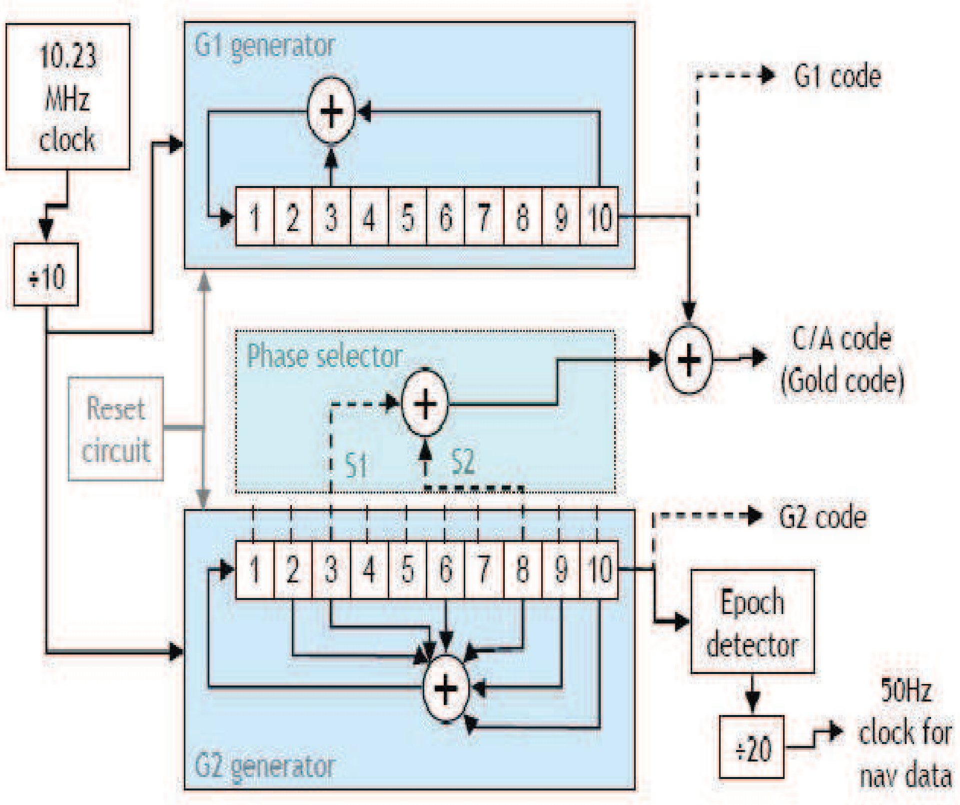

Figure 3: C/A code generator combining two maximum-length sequences into a Gold sequence.

shift: a difference between the frequency of the received signal and the frequency at the source. Due to the transit path, the point in the data block where the C/A code begins is also unknown to the receiver.

Frequency and code offsets are unknown and must be estimated. This is done in a two-step process. First, the receiver attempts to acquire the signal, by finding coarse values for the parameters. The acquisition module performs a two dimensional search on all possible shifts in code phase and frequency. A fast algorithm for this search is known as parallel search and is based on a paper by Coenen and Van Nee [3]. Once the parameters are estimated, they will be refined and kept track of as they change continuously over time. The tracking module is implemented with feedback control loops, containing two parts. Code tracking continuously adjusts the replica code to keep it aligned with the code in the incoming signal. Carrier frequency/phase tracking generates a sinusoidal signal to match the frequency and phase of the incoming signal. The used methods are based on the GPS signal properties.

Our attention has been mainly oriented towards the C/A codes $x(t)$ and the way their properties are keyed in the GPS digital signal processing stages. This code belongs to a unique family of sequences referred to as the Gold codes [4]. A Gold code is the sum of two so-called maximum-length sequences, see Figure 3, of the same length and is always balanced, i.e. the numbers of zeros and ones differs by only one. The GPS C/A code contains 512 ones and 511 zeros. The C/A code is characterized by important correlation properties, that play a key role in the acquisition stage in a GPS receiver. There the correlation function $R^{k,l}(\tau)$ is computed, given by

$$R^{k,l}(\tau) = \frac{1}{T_c} \int_0^{T_c} x_k(t)\, x_l(t-\tau)\, dt,$$

with k, l denoting satellite identification numbers and T_c the code length on which we integrate. In case of the Gold code we have the following properties

$$\begin{aligned} R^{k,k}(0) &= 1, \\ |R^{k,l}(t)| &\leq 65/1023 \approx 0.064,\ t \neq 0, \end{aligned}$$

for any k, l including $k = l$. Obviously, in case of a Gold code, the correlation function is a strong tool, since it almost vanishes for every combination of satellites and time transits except for the appropriate one.

For acquiring a signal, the receiver generates a replica of the known C/A code and attempts to align it with the incoming code by sliding the replica in time and computing the correlation. The correlation function exhibits a sharp peak when the code replica is aligned with the code received from the satellite, providing the code phase estimate.

To keep track of transit times Delay Lock Loops (DLL) are used, based on an early-late discriminator, see [2]. The DLL is based on a two-correlator structure. Each correlator is set with a small time offset relative to the promptly received signal code timing phase, producing early and late signals, as seen in Figure 4.

The three correlation outputs (early, late and prompt) are compared to see which one provides the highest correlation. Figure 5 shows two examples of this approach. At the left the incoming code was shifted in time, resulting in a maximum correlation for the late replica, while at

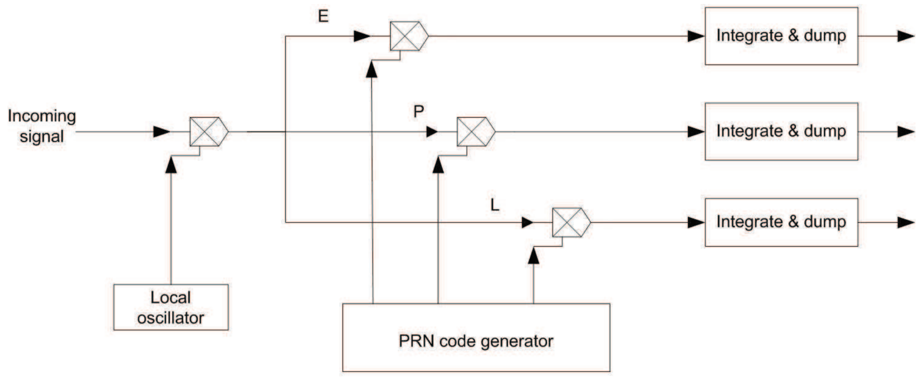

Figure 4: A basic GPS DLL diagram: the incoming signal is multiplied with a generated carrier wave and a generated code replica. In three branches, the Early, Late and Prompt correlation results are Integrated and Dumped.

the right, the incoming code remained unchanged in time, resulting in a maxium correlation for the prompt replica.

The early-late tracking loop performs well for GPS signals, due to the type of code that is used. For general signals (codes) there is no guarantee that this tracking loop performs equivalently well. However, in the case of terrain elevation data we expect this type of feedback loop to be of interest to keep track of course and speed of a platform, for at least certain regions of a terrain. For tracking the Doppler frequency offset a similar feedback loop approach is used, called the Phase Lock Loop (PLL). This tracking loop is not discussed here, since it will not be used in our proposed TRN algorithm.

The TRN approach

In the TRN system that we deal with, a sequence of samples is collected by the platform, formed from several depth measurements along a line. The database of terrain maps is searched for the ground truth profiles that best match with the recorded profile and this indicates the present position. The estimation process consists of correlating the measured elevation profile with database extracted elevation profiles.

Depth values from positions between the known elevation points are often required. These values are extracted by interpolating the map. From several interpolation techniques available, the bicubic method has been chosen for implementation in the algorithm. The bicubic interpolation [5], fits a bicubic surface through existing data points. The new interpolated

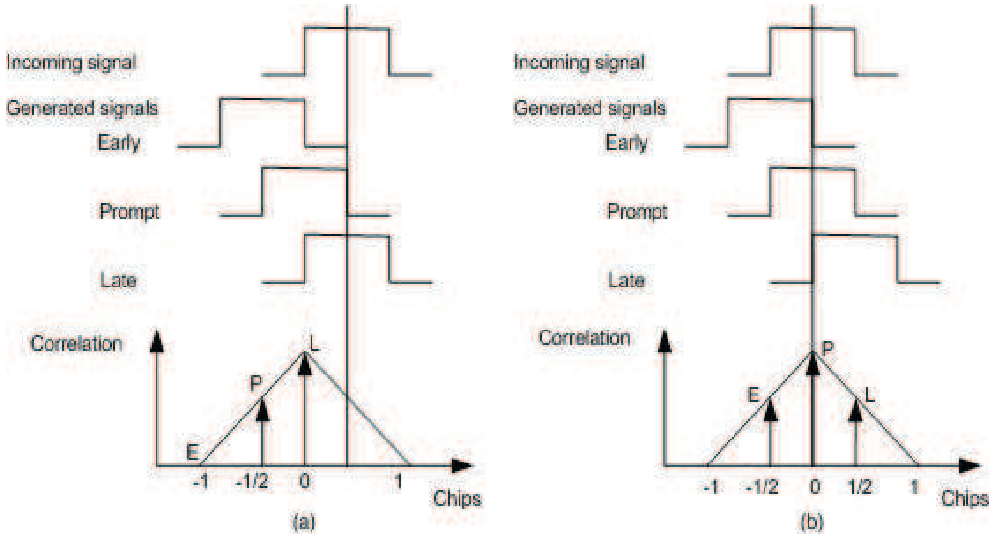

Figure 5: Code tracking: three local codes are generated and correlated with the incoming signal. (a)The late replica has the highest correlation; code sequence must be delayed. (b)The prompt code has the highest correlation; code sequence is properly tracked.

value is computed by weighting sixteen known surrounding elevation points in the database. The method is known for producing smooth surfaces. Extracted depth values will deviate from the true depth, but interpolation errors can be minimized if the velocity and course of the vehicle are accurately determined. An error in the course will determine shifts in the direction of the extracted tracks compared to the real one. An error in the speed will cause a modification in the length of the extracted tracks compared to the real one.

In order to expect accurate results from the TRN system, velocity and course of the platform should be acquired and monitored. As mentioned before, this issue is comparable to the topic of tracking code onset times and Doppler offsets in a GPS receiver. In accordance with the GPS receiver approach, a TRN correlation algorithm in two stages was implemented. During the 'acquisition module' a two dimensional search of all possible shifts in speed and course is performed, calculating coarse estimates for the two parameters. Next, the 'tracking module' begins by adding new depth measurements and deleting old ones. The position is progressively updated. The values of the speed and the course are refined and kept track of as they change during the platform's movement.

Two tracking loops will be required. A block diagram of the implemented TRN system has been depicted in Figure 6 to illustrate the global

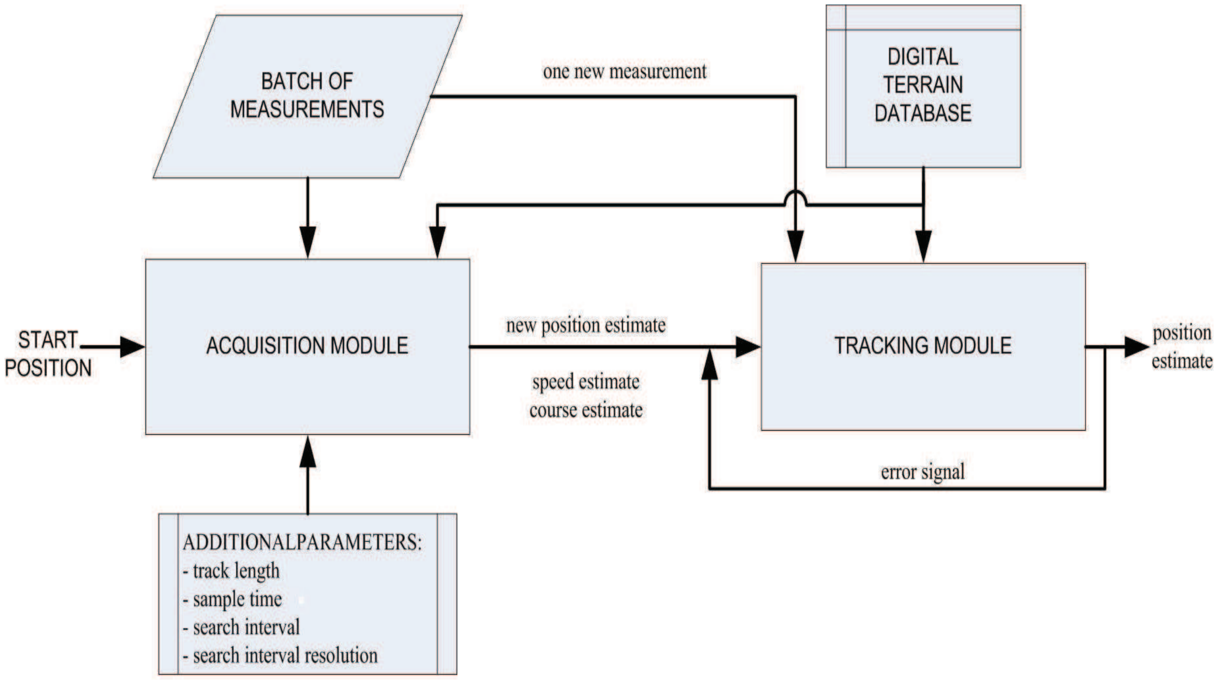

Figure 6: Block diagram of the proposed TRN algorithm.

operating structure. Here, we give a brief description of the functionality of each block in the diagram.

Digital terrain database
Shuttle Radar Topography Mission (SRTM) digital elevation models have been used as maps. These types of files contain only binary data corresponding to the height, with no georeferenced or other related information. The resolution of the map is 3 arc seconds, corresponding to 90m between elevation points. The SRTM elevation models cover the continental surface of Earth. It was preferred to have a real-world database, as we lacked access to accurate sea bed charts. However, a desert area has been used.

Batch of measurements
The depth measurements which are made by the platform come straight out of the reference database. They are extracted using the 'correct' values for speed, course and position. If the coordinates do not concur with the grid points, interpolation techniques are used. This profile is used in the error-free simulations. However, to get closer to a real approach, depth measurements errors must be added to this profile.

Acquisition block
The focus of the research was to investigate whether the platform's position

can be tracked in time similar to a GPS approach, by estimating the velocity and the orientation of the vehicle. Based on these considerations, we assume that the start position has been accurately determined. Meanwhile the speed and the course are due to be estimated. The acquisition module is built up as a search process of these two parameters. A long batch of measurements gives a terrain elevation profile which is compared to several extracted profiles. Serial search acquisition is performed: for each value of the speed, different values for the course are taken and a profile using these values is extracted and correlated with the measurements. When all possibilities have been verified, a correlation matrix is built and plotted, and the best estimates are picked up and delivered to the tracking module, together with the position estimate.

Tracking block

The number of measurements in the tracking phase is fixed. When the next depth measurement is done by the platform, the profile propagates one measurement ahead, discarding the oldest one. The tracking loop has been built in the same manner for both speed and course estimation. The GPS early-late tracking concept has been used as a basic principle. A diagram of the basic TRN tracking loop has been depicted in Figure 7.

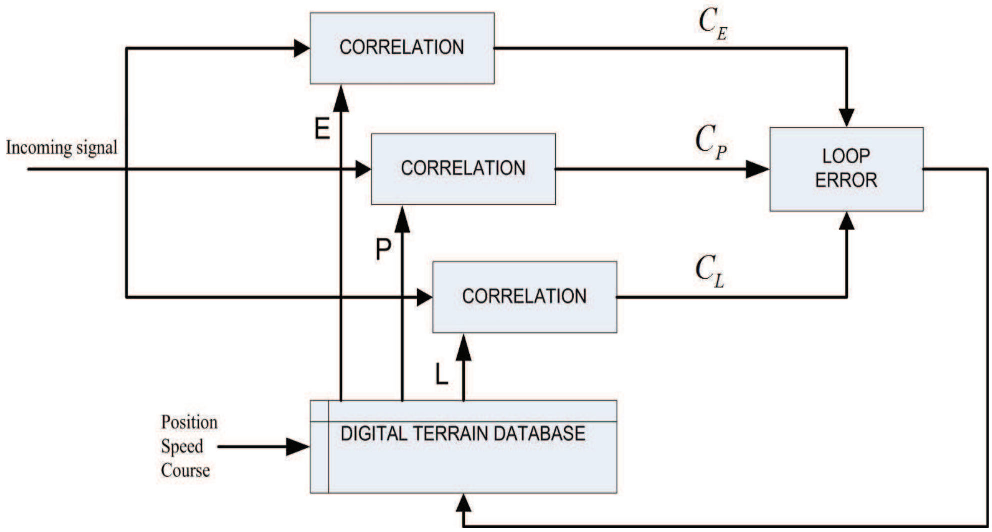

Figure 7: A basic TRN tracking loop diagram.

The incoming signal consists of the batch of measurements. We generate two local replicas, by extracting profiles from the database using two small offsets relative to the estimated speed and course. A prompt replica of the profile will be extracted, using the exact estimated values of the parameters. The correlation between these tracks and the measurement profile is computed. The output is a numerical value indicating a matching property of the specific signal replica with the incoming signal. An error signal is provided from the combination of the correlation outputs.

Simulations and Preliminary Results

The 'correlation' method used in our approach does not compute correlation in a strict statistical sense, but uses a matching function that shows how well a measured bottom profile matches the map. For such matching functions we consider a series of m measurements: d_i will denote an incoming measured depth and $h_{i,k}$ the height in the map in the kth location, or simply written h_i, with $i = 1, \ldots, m$. The different tested methods are introduced below:

- The Pearson product-moment correlation coefficient. This coefficient measures the linear dependence between two variables, see[6]. Based on samples of paired data, the coefficient is calculated as:

$$C_{d,h} = \frac{\sum_{i=1}^{m}(d_i - \overline{d})(h_i - \overline{h})}{\sqrt{\sum_{i=1}^{m}(d_i - \overline{d})^2} \cdot \sqrt{\sum_{i=1}^{m}(h_i - \overline{h})^2}},$$

  with $\overline{d}$ and $\overline{h}$ the mean of the data d and h respectively.

- The average distance. In this case the absolute distance between the measured depth and the map depth in the candidate points is computed and then averaged over m:

$$C_{d,h} = \frac{1}{m} \sum_{i=1}^{m} |d_i - h_i|.$$

- The squared difference. The differences between the depth measurement and the correlation depths are calculated and then these values are squared and summed and divided by the number of measurements. The standard deviation of the differences is added to reduce the effect of the systematic errors. For non-correlating tracks, the deviation will be quite large. The formula for this correlation measure is given by

$$C_{d,h} = \frac{\sum_{i=1}^{m}(d_i - h_i)^2}{m} \cdot \sigma(d - h),$$

with $\sigma(d-h)$ the standard deviation of the difference data $d-h$.

Correlation between a measured profile and several extracted tracks has been calculated using, in turn, the three formulas. For simulations the following set up has been used:

1. The measured profile is error-less and is extracted from the database;
2. The extracted profiles are selected by varying, in turn, the error in the velocity and orientation;
3. A set of different scenarios were used.

Figure 8 shows an example of the difference in the three different types of correlators for different errors in the speed (the functions have been rescaled or reversed, for the sake of comparison). For the errors in course a similar figure can be depicted. The average distance function is imple-

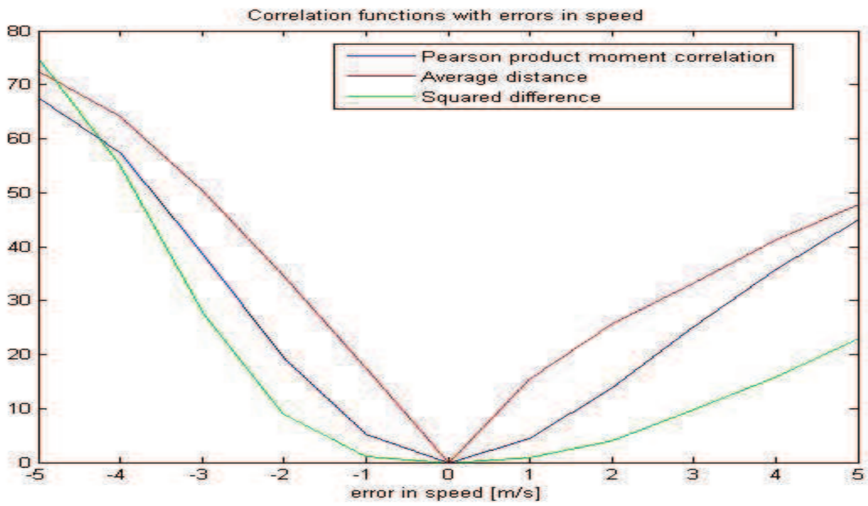

Figure 8: Correlation functions dependent on speed parameters.

mented in our TRN algorithm. It has been chosen because it is the most discriminating in the area of interest (convex, 'V'-shaped). Choosing this correlating function will result in more notable differences for small errors of the parameters.

A key driver on the performance of the overall system is the terrain and its characteristics: the uniqueness of the ground profiles within the

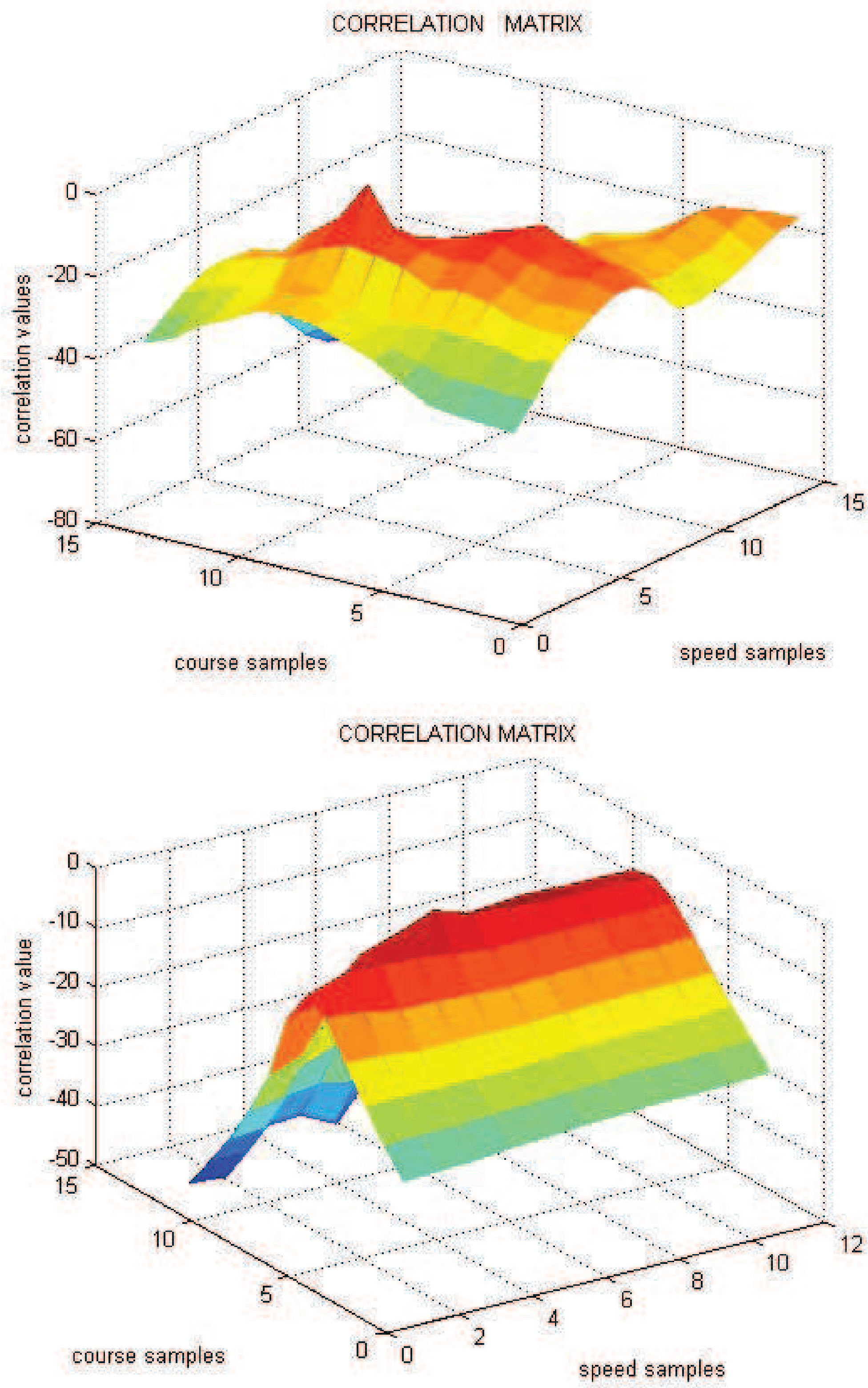

Figure 9: Displayed correlation matrices showing a clear visible peak (up) and a less visible peak (below)

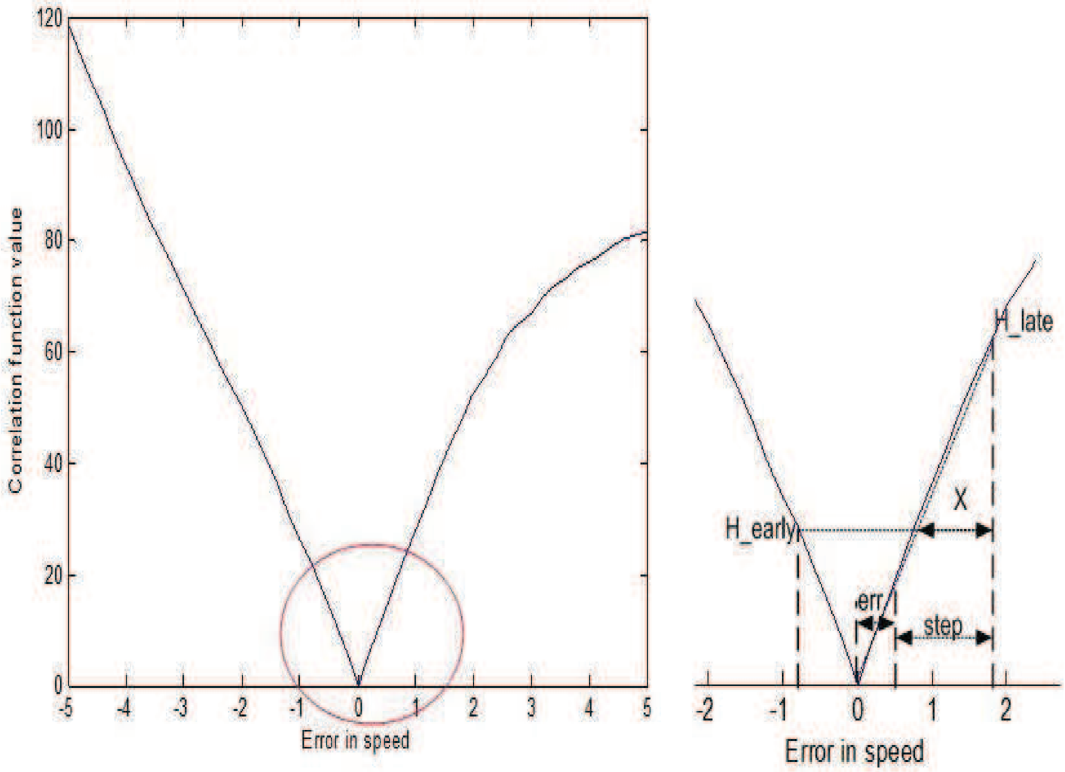

Figure 10: Average distance correlation function and error correction.

searched map area. The cross correlation between tracks may have a dynamic range, due to the unpredictability of terrains. The correlation plot will show a significant peak (auto-correlation peak) when the tracks are matched. However, additional peaks are likely to appear, leading to position ambiguity. Figure 9 shows two different situations: (Top) the auto-correlation peak can clearly be seen; (Bottom) the auto-correlation peak is less obvious and an additional peak can be noticed.

The existence of a prominent auto-correlation peak has been the premise of adapting the GPS 'early-late' tracking concept to TRN. Both tracking loops have been built up following the same rationale. Due to this remark, we restrict ourselves to the loop that follows the changes in speed. The early and late versions are separated by a distance we called step. Assuming the symmetry of the function and linearizing it, a correspondence between the error and the 'X' segment is assumed, as shown in Figure 10.

The error signal is calculated using:

$$X = \frac{\text{step} \cdot \left(H_{\text{late}} - H_{\text{early}}\right)}{H_{\text{late}}},$$

with X, step, H_{late} and H_{early} as in Figure 10.

The performed tests have given good results, as the correction is determining a convergence for the estimated speed. However, the terrain shape has a big influence on the number of iterations needed in order to obtain a good estimation, as can be seen from Figure 11 (for speed, with similar results existing for course).

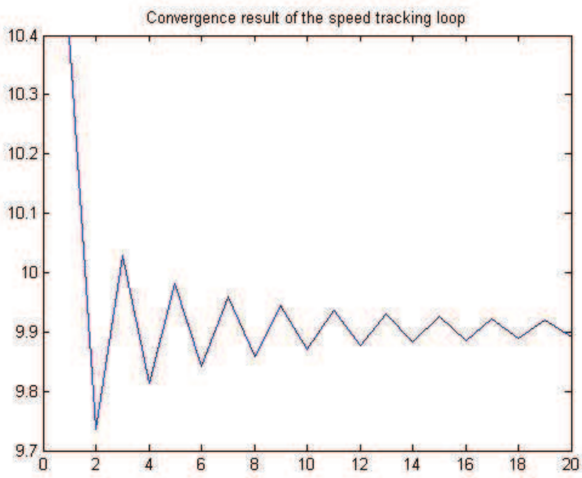

Figure 11: Convergence results of the tracking loop for speed.

Conclusions and Future Work

In this article a GPS inspired approach has been developed for TRN systems. We investigated whether the acquisition and tracking GPS concept has the potential to be applied in the TRN field with good results. The performance of the proposed system is analyzed using MATLAB simulations. The results obtained so far have not confirmed this assumption. Good estimates of the velocity and course of the vehicle are obtained. However, we are aware that we can not conclude that the testing is representative for terrains in general.

At most we can say it is representative for our own database. The terrain characteristics are a key driver in the performance of the overall system. Measurement and database resolution limitations can make it difficult to distinguish features nearby, while environments often contain repeating features.

The approach presented in this article has mainly been based on [7].

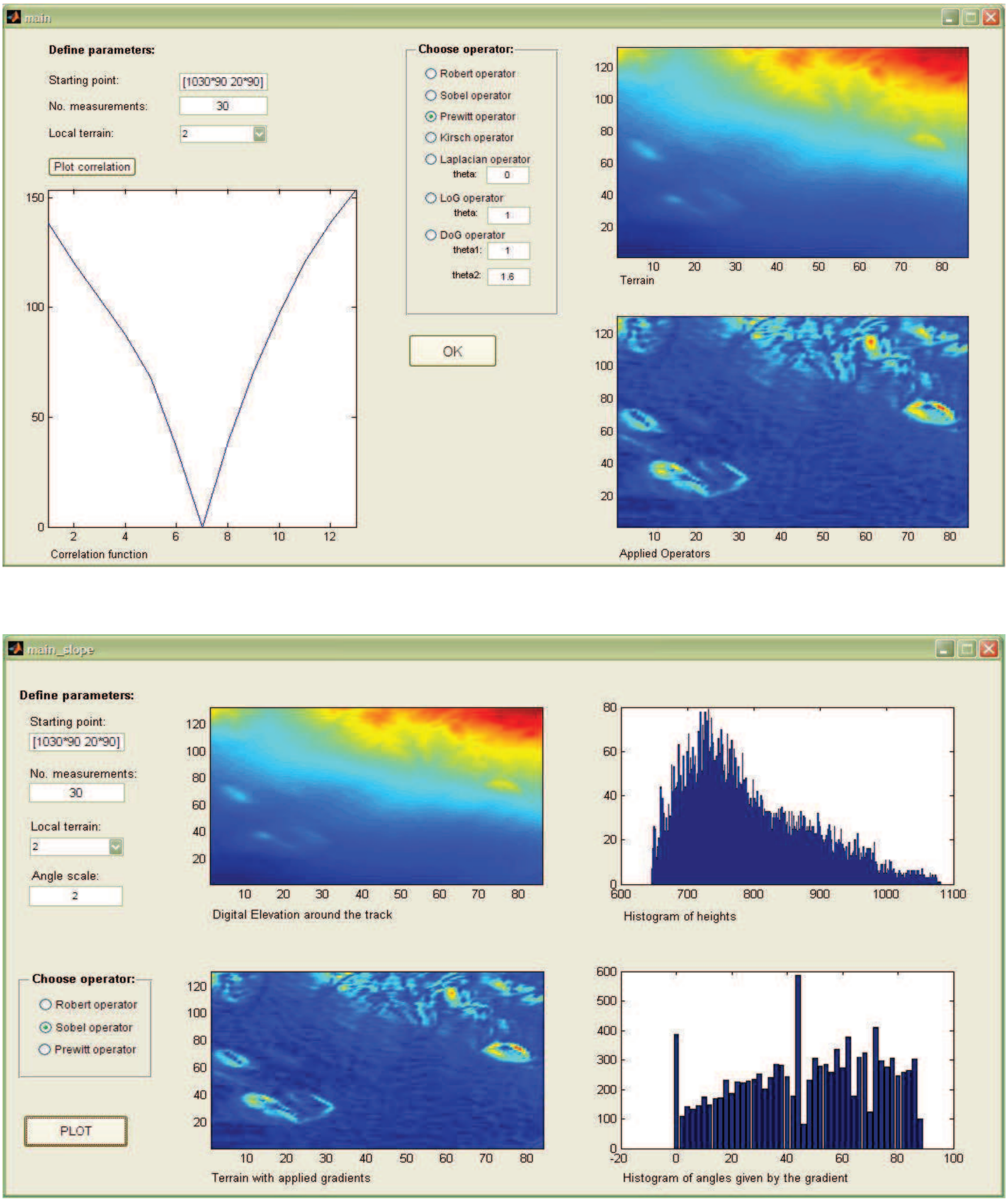

Figure 12: Screen shots of the newly derived simulation tool for studying terrain parameters and the relation to correlation results in more detail. In the upper picture: Transformation of the terrain using the Prewitt operator with corresponding correlation result. In the lower picture: Transformation of terrain using the Sobel operator with an analysis of heights and gradient angles.

Currently, we study the proposed algorithm in a more rigorous way by taking terrain parameters more into account. A new simulation tool has been built, see Figure 12 in order to analyze the relation between terrain conditions (depth, roughness) and the algorithm's performance. Primarily, a classification of the terrains that form the database of terrains we use is elaborated on. In a latter stage, the parameters found from the terrain description will be used to quantify the accuracy and availability that can be achieved by the proposed TRN system. Results from this work will be presented in an upcoming paper [8].

References

[1] D. Akos, K. Borre, N. Bertelsen, P.Rinder and S. Jensen, *A single-frequency software-defined GNSS receiver*, Birkhaüser, 2007.

[2] P. Misra and P. Enge, *Global Positioning System - Signals, Measurements, and Performance*, 2nd edition. Lincoln, (Massachusetts): Ganga-Jamuna Press, 2006.

[3] D.J.R. Van Nee and A.J.R.M. Coenen, "New Fast Code-Acquisition Technique Using FFT," *Electronic letters*, vol. 27, pp. 158–160, 1991.

[4] R. Gold, "Optimal binary sequences for spread spectrum multiplex", *IEEE transactions on Information Theory*, 619-621, 1967.

[5] Application note: on bicubic interpolation topic. http://www.mathworks.com/access/helpdesk/help/toolbox/vipblks/ug/f12622.html

[6] J.L. Rodgers and W. A. Nicewander, "Thirteen ways to look at the correlation coefficient," *The American Statistician*, vol. 42(1), pp. 59–66, 1988.

[7] D.Vaman and P.J. Oonincx, "Exploring a GPS Inspired Acquisition and Tracking Concept for Terrain Referenced Navigation," *Proc. Technical Meeting ION*, San Diego, 2010.

[8] D. Vaman and P.J. Oonincx, "Performance Analysis of a GPS Inspired Terrain Referenced Navigation Algorithm," *Proc. European Navigation Conference GNSS 2010*, Braunschweig, 2010 (to appear).